Natural-Language Retrieval of Images Based on Descriptive Captions

EUGENE J. GUGLIELMO
Monterey Bay Aquarium Research Institute (MBARI)
and
NEIL C. ROWE
Naval Postgraduate School

We describe a prototype intelligent information retrieval system that uses natural-language understanding to efficiently locate captioned data. Multimedia data generally require captions to explain their features and significance. Such descriptive captions often rely on long nominal compounds (strings of consecutive nouns) which create problems of disambiguating word sense. In our system, captions and user queries are parsed and interpreted to produce a logical form, using a detailed theory of the meaning of nominal compounds. A *fine-grain* match can then compare the logical form of the query to the logical forms for each caption. To improve system efficiency, we first perform a *coarse-grain* match with index files, using nouns and verbs extracted from the query. Our experiments with randomly selected queries and captions from an existing image library show an increase of 30% in precision and 50% in recall over the keyphrase approach currently used. Our processing times have a median of seven seconds as compared to eight minutes for the existing system, and our system is much easier to use.

Categories and Subject Descriptors: H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*indexing methods*; *linguistic processing*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*query formulation*; *search process*; *selection process*; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*predicate logic*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*language parsing and understanding*

General Terms: Algorithms, Experimentation, Human Factors, Performance

Additional Key Words and Phrases: Captions, multimedia database, type hierarchy

1. INTRODUCTION

Recent work in information retrieval has looked at applying natural-language processing and knowledge-based techniques to overcome the

This work was performed while E. J. Guglielmo was employed by the Naval Air Warfare Center Weapons Division, China Lake, CA 93555.

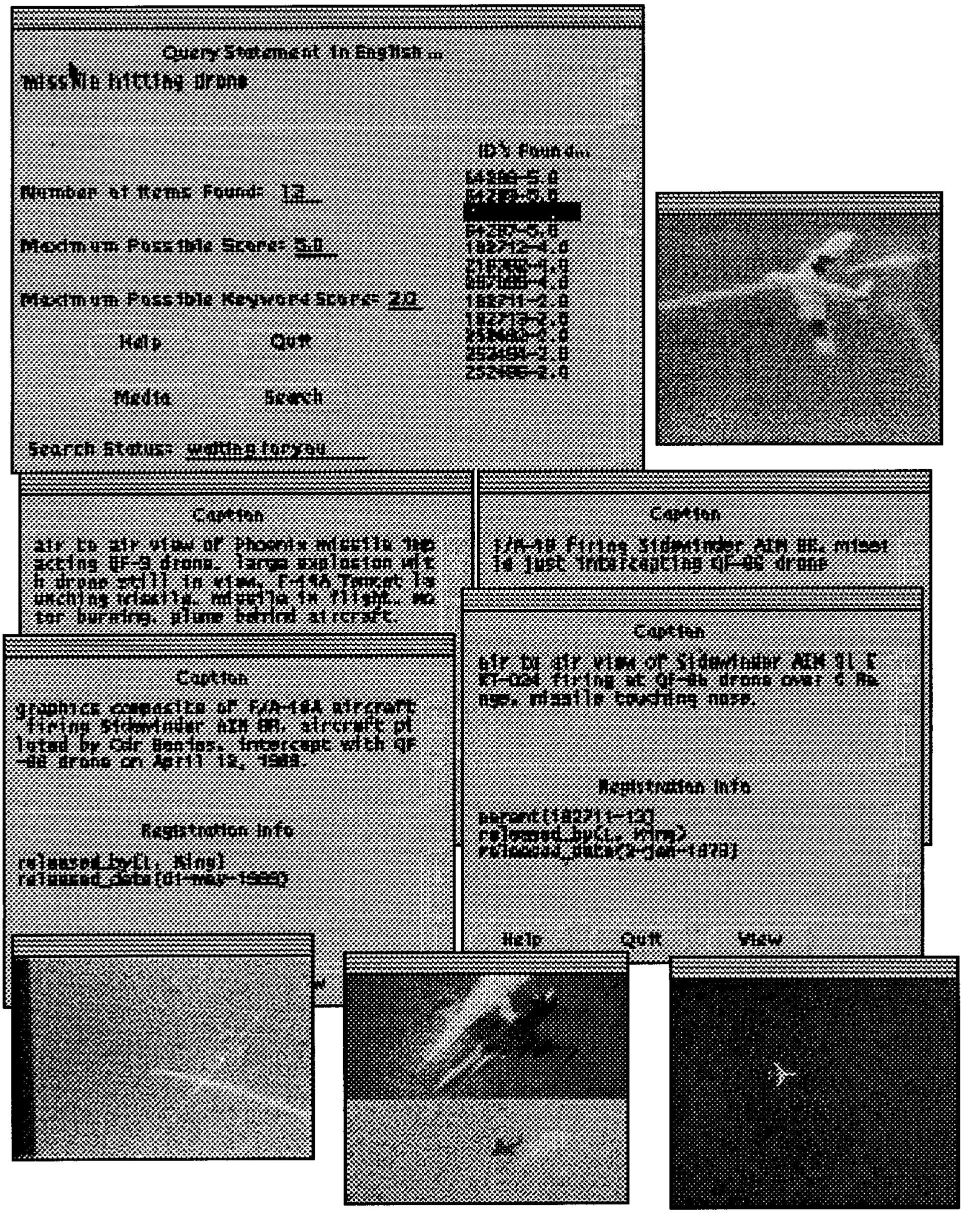
Author's addresses: E. J. Guglielmo, Monterey Bay Aquarium Research Institute, 7700 Sandholdt Road, Moss Landing, CA 95039; email: gene@mbari.org; N. C. Rowe, Naval Postgraduate School, Monterey, CA 93943.

shortcomings presented by keyword-based retrieval. In keyword approaches, the user is often required to remember the valid words (i.e., keywords), how these keywords correlate with the concepts that he or she wishes to find, and how the keywords may be combined to formulate queries. By removing such limitations and allowing unrestricted English phrases for queries as well as for describing the data, the goal is an information retrieval system that will be easier to use and provide more relevant responses to user queries. Our work looks at applying these artificial intelligence techniques for the general problem of retrieving multimedia data identified by caption descriptions. This article describes these techniques applied specifically to images.

We have developed and tested our ideas using the image database from the Naval Air Warfare Center Weapons Division China Lake Photo Lab. This database contains over 100,000 historical photographs, slides, and selected video frames of aircraft and weapon projects from the last 50 years. The images show mostly aircraft and missiles in flight, weapon systems configured on various aircraft, targets and drones being hit, aerial views of the scenery surrounding the Center, test equipment, etc. The images are used for reference, project documentation, and publications. Registration (bookkeeping) information captures customer information about a particular image shoot, including customer name and code; the date, time, and location of the shoot; the photographer and film used; cross references to related images; and an English caption description. Image identifiers are used to uniquely index the images.

The caption provides free-form information about the image, to describe either an event occurring in the image or unique characteristics and features of weapon systems and the Center. The linguistic structures of the captions can be characterized as a sublanguage [Sager 1986] with many nominal compounds but few verbs, adverbs, determiners, and conjunctions. The nominal compounds cause difficulties for a keyword-matching approach because order in the compound is crucial to meaning. For instance, "target tank" refers to an actual tank, but "tank target" means a construction that looks like a tank to radar, so there is no tank in a "tank target." Krovetz and Croft [1992] show that these and similar ambiguities cause significant misunderstandings in information retrieval.

Anick [1991] pointed out the user preference for natural-language querying versus Boolean expression queries when looking at enhancements for the STARS retrieval system. Our prototype system to support natural-language querying is called *MARIE*—Epistemological Information Retrieval Applied to Multimedia. The user interface for MARIE is shown in Figure 1 and consists of three types of windows: a window type for entering an English query and listing the search results ("Query Statement (In English)"), a window type to view the captions and registration information (of which there are four instances in Figure 1), and a window type to display images (of which there are four instances).



Query Statement in English...

missile hitting drone

ID's Found...

64206-5.0
64289-5.0
[illegible]
64207-5.0
182712-4.0
210300-4.0
687100-4.0
182711-2.0
182713-2.0
253493-2.0
253494-2.0
252495-2.0

Number of Items Found: 12

Maximum Possible Score: 5.0

Maximum Possible Keyword Score: 2.0

Search Status: waiting for you

Caption

air to air view of Phoenix missile impacting QF-9 drone. large explosion with drone still in view. F-14A Tomcat launching missile. missile in flight. motor burning. plume behind aircraft.

Caption

F/A-18 firing Sidewinder AIM 9R. missile just intercepting QF-86 drone

Caption

graphics composite of F/A-18A aircraft firing Sidewinder AIM 9R. aircraft piloted by Cdr Benlos. intercept with QF-86 drone on April 12, 1983.

Registration Info

released_by[L. King]
released_date[01-may-1983]

Caption

air to air view of Sidewinder AIM 9L f [illegible] firing at QF-86 drone over C Range. missile touching nose.

Registration Info

parent[182711-13]
released_by[L. King]
released_date[3-Jan-1978]

Fig. 1. MARIE query interface. Images corresponding to result windows are shown counterclockwise starting at the bottom left.

1.1 Previous Work

Two approaches in applying natural-language processing and knowledge-based techniques for information retrieval are text skimming and full parsing. Text skimming tries to extract all the information in designated categories present in some text, as for instance in scanning news wires for reports of terrorist activities. Examples of this popular approach include CYRUS [Kolodner 1983], SCISOR [Rau 1987], and FERRET [Mauldin 1991] which have used such ideas as spreading activation and script modeling [Schank 1977]. However, both text-skimming methods and script modeling are generally quite domain dependent, and the methods that work for one application may not work for another. General-purpose information retrieval requires methods more domain neutral.

The alternative is to try to capture all the important meaning of some text in data structures that can be easily indexed. There is a wide range in the degree of linguistic analysis used by such systems, as surveyed in Smeaton [1992]. The range is from systems that add a type hierarchy to keyword matching such as PLEXUS [Vickery and Brooks 1987], to sentence-summarization methods such as START [Katz 1988], to full parsing as in Sembok and van Rijsbergen [1990]. These systems do require some domain knowledge, but generally the information in a simple dictionary, including syntactic categories, a type hierarchy, and part-whole relationships will suffice. Such information is now available for most of the English language in systems such as WORDNET [Miller et al. 1990]. We chose a full parsing approach for our own work because we believed our sublanguage was easier than unrestricted English (thanks to the rarity of verbs, pronouns, and dialogue issues), and our captions are relatively short, averaging 40 words in length.

Once appropriate information has been extracted from a user query, several methods can be used to efficiently match data. With a type hierarchy, indices become hierarchical, and we will show shortly how they can be simplified using specialization. As an indicator of importance, index items can be weighted by frequency of occurrence. If the meaning is represented as a semantic network rather than keywords, indexing can be done on relationship names or property names to improve match precision, as in Sembok and van Rijsbergen [1990]. This is important for nominal compounds where there can be many unstated relationships among the words. But matching of multiple relationships requires match consistency, first using the major concepts (nouns and verbs), then their relationships as expressed with a case grammar. Thus the penalty one pays for smarter query processing is a harder matching problem, albeit one limited by the size of a usually short user query. Rau [1987] proposed a method where first the key nouns in the query were matched using the type hierarchy with only those nouns with scores exceeding a threshold being subjected to a more intense match. We use this idea in our work as our queries usually mention quite specific nouns.

With image data, content analysis of the data can be combined with natural-language processing of the captions to supplement the original English caption, as in the weather map system of Yokokota et al. [1984] and the face-locating program of Srihari and Rapaport [1989]. But content analysis of audio, pictures, and video can be very slow and is only successful for narrowly restricted content. We prefer instead a general-purpose information retrieval system for captioned data.

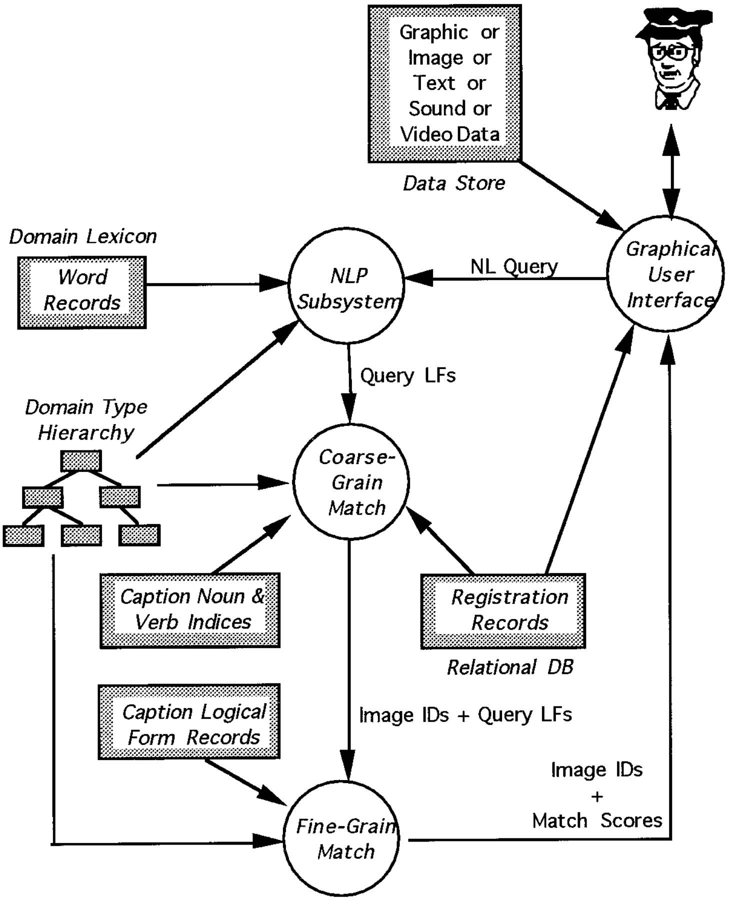


Fig. 2. Query processing flow.

Section 2 describes the approach we have taken. Section 3 describes the current implementation. Section 4 provides the performance results, and Section 5 contains concluding remarks.

2. APPROACH

Figure 2 illustrates the query-processing flow. A graphical user interface accepts an English query from the user and sends it to the natural-language processing (NLP) subsystem. Using a domain lexicon and type hierarchy, the NLP subsystem parses the query into logical form records. Once these records are produced, caption matching proceeds in two phases. The first phase is an enhanced keyword search referred to as the *coarse-grain* match. The match uses the nouns, verbs, and relationships inferred from the prepositions found in the logical form records to search the caption noun and verb indices and the registration data. Registration data are stored in a relational database system whereas the noun and verb indices as well as the logical form records are stored in ordinary files. The most promising image identifiers found by the coarse-grain match are next presented to the *fine-grain* match (second phase) which selects the corresponding caption logical form records for matching against the query logical form records. The results of this match are sent to the user interface. The user is then able to select the image and registration data for any of the image identifiers listed. This process will be further explained shortly.

2.1 Natural-Language Processing

The NLP subsystem translates English captions into a logical form [Allen 1987] describing the images and translates English queries into a similar logical form for matching against the caption logical forms. A logical form is a collection of case grammar records that capture the meaning of an English text. We adapted the case grammar constructs defined in Haas [1991] for the logical form and structured it similarly to the slot-assertion notation described in Charniak and McDermott [1987]. The slot-assertion structure we produce can be viewed as a neutral structure that uses type hierarchy classes for identifying instances (lexical tokens). For example, the logical form record "inst(noun(1234-1-1), Sidewinder)" defines the lexical token "noun(1234-1-1)" as an instance of the class "Sidewinder." We describe this structure as being neutral in that it is not based on predefined knowledge structures such as frames [Minsky 1981] or scripts [Schank 1977]. Both were not employed to avoid the problems encountered in Mauldin [1991]. Specifically, the problem with these structures is that someone has to handcode the applicable ones for each domain in addition to setting up the lexicon. A shortcoming of not using them, however, is that the resulting logical form may not provide a complete description of the events in a caption and may not answer some query questions correctly.

In a logical form, verbs have predefined cases to capture the meaning of the verb's activity such as agent, destination, location, recipient, etc. In addition to verb cases, an idea proposed by Allen which we employed was to allow nouns to have cases. Thus for the sentence "missile on stand mounted on the truck," only one main verb would be encountered (i.e., "mounted") where "on the truck" would be associated with its location case, and "on stand" would be associated with the location case for the noun "missile."

Parts of the natural-language processing subsystem were adapted from the Database Generator Message Understanding System [Montgomery et al. 1989], which was used for understanding aircraft pilot dialogues. This system employed a syntactic parsing strategy guided by rule weights. Unlike dialogue, the captions contained numerous nominal compounds as well as confusing prepositional and participle phrases. In fact, many captions were not sentences, but a sequence of noun phrases. These noun phrases used very few determiners (which exist in dialogue) making it difficult to distinguish some nouns from verbs. To understand the activity in the image, an understanding of the interactions expressed by nominalized verbs was needed. Spatial relationships among objects were characterized by the prepositional phrases: for example, a “missile under wing” versus “missile away from aircraft.” Analysis of these phrases leads to additional relationships being inferred about the objects. In the first phrase, the fact that a missile is under the wing can be taken to imply that the missile is also on the aircraft. Limitations in the lexicon structure limited us in using only one word sense as defined by its military usage. Thus the term “Sidewinder” refers to a missile and not the snake.

Problems we encountered in the syntactic parsing of captions contradicted our assumption that parsing sublanguage phrases would be relatively easy. This shortcoming leads us to speculate that an automated text entry tool for entering captions may simplify some of the natural-language-understanding issues. This tool could check both spelling and grammar applicable for the sublanguage. In addition, if the goal is to perform matching using semantic structures, then deriving a syntactic parse and using it to derive a logical form may not be as productive as one would be led to believe. The approach described by Katz in the START system would be a better starting point. Further information on the natural-language processing for this work can be found in Guglielmo [1992].

2.2 Type Hierarchy

The type hierarchy contains both verb and noun classes that correspond to world and domain-specific knowledge. In creating the logical form records, the type hierarchy allows specialization of class instances through examination of the nominal compounds, establishment of correlations among classes (e.g., *part-of, owned-by, employee-of*) and rules for defining case relations (e.g., *agent, source, theme*). For example, the aircraft class will usually have an owner and registration number, contain various components such as wings, a nose, and tail, and be the agent of most missile firings or bomb droppings. The overall appearance is that of a tree. Inheritance is used to specify defaults when specific rules for a class do not exist.

Two issues in defining the classes involve the handling of proper nouns and verbs. To illustrate, for a query containing “AIM-9M” (a kind of Sidewinder missile), a keyword search should be able to find immediately those images that contain this specific version of the missile because of its mere presence in the query. Likewise, if “AIM-9” is supplied in the query, the search process should find all “AIM-9” images. But a keyword search for “AIM-9” will not find the “AIM-9M” or other Sidewinder versions unless a wildcard character is used. Likewise, if the query contains “Sidewinder,” a keyword search finds only those captions that contain the word “Sidewinder,” and not those that contain some variant of “AIM-9*x*” or some other designator for “Sidewinder” if one was used. By creating classes for the versions and organizing the classes into a type hierarchy, we can then search for versions either as a group or individually. For example, if “AIM-9M” is specified as a subclass of “AIM-9,” a search for “AIM-9” will include a search for “AIM-9M” as well (see Figure 3).

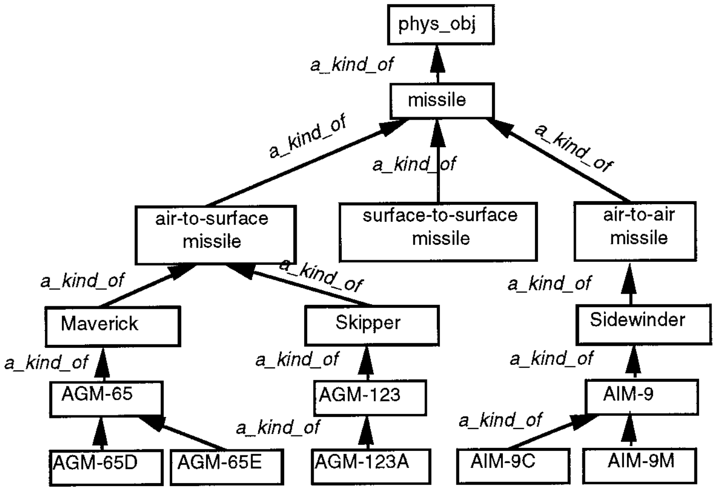


Fig. 3. Type hierarchy structure.

In the preceding paragraph, the noun classes shown were individual words (e.g., "missile"). However, a noun class may itself be a nominal compound (e.g., "U S Navy," "aircraft carrier," "Beavertail cactus"). This treatment reflects the current Photo Lab view of how index terms are currently handled and what images should be retrieved for the terms. In addition, these class terms can be part of a long nominal compound which then has to be interpreted properly. Gay and Croft [1990] described the use of categories, roles, and an interpretation algorithm for handling nominal compounds from the CACM collection where the majority of compounds consisted of less than three nouns. We have had to develop a similar technique for interpreting military nominal compounds that may have more than three nouns. By virtue of a class/subclass relationship, certain phrases can be simplified through the use of specialization. For example, the nominal compound "Sidewinder AIM-9R missile" reduces to the class "AIM-9R" based on the structure in Figure 3. The following nominal compounds require looking at the correlations and cases as well as specialization:

HTW (Helicopter Trap Weapon) multiple drop sequence

USN UH-2A BU# 149033 BQM control helicopter (full side view)

USMC VMA-513 AV-8A BU# 158389 Harrier aircraft (nose 6 and WF on tail).

In the last phrase, the head noun is "AV-8A" with "USMC" playing the role of owner, "VMA-513" the role of squadron, "BU# 158389" the registration number, and a nose part with identifier 6 as well as a tail part with identifier "WF."

The second issue is the handling of verbs. Verb classes were created based on ideas presented in Charniak and McDermott [1987], Schank

[1977; 1975], and, to some extent, Allen [1987]. The more important issue for us is the treatment of nominalized verbs and whether a distinction needs to be made between the verb and the nominalized verb; that is, should "assembly" and "assemble" coexist in the type hierarchy? Anick [1991] stated that stemming will sometimes map unrelated words to the same stem, inadvertently affecting the meaning, which would indicate that both words should coexist in our system to preserve meanings. We explore this more closely by assuming the lexicon has a separate entry for "assembly" and "assemble." As we look ahead to the matching, if the two are treated separately, then the query "soldering assembly area" and caption "the area for assembling solder" will not match completely using the parser. The coarse-grain match will perform lookup on "soldering," "assembly," and "area" as well as "assembling" and "solder." "Soldering" does not match "solder." Likewise, the root form of "assembling" ("assemble") does not match "assembly." Only "area" would match and contribute toward the determination of the match score. On the other hand, if only roots are used, then the two phrases would have "soldering" being treated as "solder" and thus matching "solder" in the second sentence. "Assembly" is treated as "assemble" and is matched to the root of "assembling" in the second sentence. "Area" matches "area." There are more matches possible in the second case, allowing progression to the fine-grain match for further analysis. There is no assurance, though, at this stage that the two phrases are indeed a good match.

We initially created two different types of logical form records for "assembly" and "assemble," the former being noun centered with associated noun cases, the latter being verb centered. However, retaining both logical forms complicated the matching, as we may have to additionally transform verb-centered logical form records created from queries to noun-centered logical form records found in captions or vice-versa. Thus, we decided to move toward one logical form type and transform the other to it. The major question was deciding on which logical form type—noun or verb centered. When parsing the captions, we discovered that when the system was able to distinguish a nominalized verb from an actual verb, processing of the adjectives and nouns preceding the nominalized verb was less difficult than determining the verb cases, favoring a noun-centered approach. However, Dick and Hirst [1991] suggested using more verbs in the logical form. To aid in our decision, we experimented first with a small number of captions by creating a *transformation* slot in the type hierarchy with rules that could be applied to specific words once certain conditions were met. These rules substitute one subset of logical form records for another. In the caption

> Sparrow III (Guardian of the Skies) operational with Seventh Fleet in western Pacific. four missiles on underside of F3H-1 BU# 137010 aircraft (Point Mugu 7010 on tail). air to air view from side during firing on one missile.

the nominalized verb "firing" whose canonical form is the noun "launch" with theme "Sparrow III" (a type of missile) is transformed to the *activity* "launch" with the same theme. In the caption "Shrike firing at SCR 584 Bullpup target," the parser interpreted "firing" as a verb and assumed "Shrike" to be the agent of "firing." However, "Shrike" (a missile) cannot "fire" anything and is actually the theme of "firing." Thus a general transformation rule for the missile class was created that transformed any missile acting as the agent of a launch to a theme of launch. Both these examples transformed nominalized verbs to verbs with their associated cases.

In addition to noun- and verb-centered transformations, we discovered that transformations can be applied to other types of situations as in "F/A-18 with Maverick missile." A transformation rule can take an *attribute* case containing a *part-of* relation and translate it to a *location* case to indicate that the missile is located on the aircraft. Our major motivation for this rule was to move toward more accurate canonical representations, since the *part-of* relation, for example, is not a true case grammar construct but is captured in the more general attribute case.

Transformations, however, did not address all our logical form problems. Sometimes new logical form records needed to be added based on the existence of certain objects or relationships encountered. As a result, we created *inference* slots and rules attached to a specific concept that examine what is known about a concept, and if some information does not exist, adds it. These rules, however, turned out to be very domain dependent and require additional manual effort, suggesting in turn the use of scripts. Our approach, however, can be viewed as a collection of the components contained within scripts. For example, in the earlier caption containing "air to air view," its theme, which is not stated explicitly, is inferred to be the Sparrow 3 missile. However, the location of the Sparrow 3 must be in the air because of the phrase "air to air view." Hence, a location case for the Sparrow 3 can be inferred that places it in the air. This inference then allows matching to queries such as "airborne Sparrow missile" or "Sparrow missile in the air." In the caption "Cdr. Antonio with night vision goggles in cockpit," the verb "wears" can be inferred based on the way nouns are connected via the prepositions. This inference then allows the caption to match the query "pilot wearing goggles."

A somewhat riskier inference appears in the following caption involving the verb "uploaded."

> air to air view of Harm missile launched from F/A-18A BU# 161720 aircraft (nose 102) over NWC ranges. Harm, pod, and MK 91 Silver Bullet uploaded. excellent view of aircraft just before firing of missile.

We see that several pieces of equipment are "uploaded," but the destination is not explicitly stated. However, an implicit assumption can be made that they are on the aircraft, and a rule can be added that makes this inference. A similar approach was described in Finin [1986] in which new facts were added to the database if certain conditions were found to exist in the text being parsed.

The results of our experimentation with various queries and captions was a decision to move gradually to a verb-centered approach and allow matching to be based on verb cases while utilizing noun cases predominately for adjectives and number. We have retained both the transformation and inference slots in the type hierarchy and have used this information in the last stages of the natural-language processing to create the logical form records. We have observed that to improve the matching requires that we generate better meanings which, in turn, means more domain information is sometimes needed. The question as to the amount of domain knowledge necessary needs to take into account the caliber of people formulating the queries. Most technical personnel will focus on specific objects and concepts of interests, concentrating on leaf classes in the type hierarchy. The queries from nontechnical personnel will usually be in the interior classes of the type hierarchy as they are not familiar with the specific technical jargon indicative of the leaf classes. This view is also true when dealing with correlations among classes.

We believe that many inference and transformation rules we have created are applicable across domains. For example, many domains may use aircraft and vehicles, and it will always be true that an aircraft has wings and that land vehicles have wheels in those domains. This would seem to suggest a common type hierarchy of concepts that can be shared across domains, together with specialized type hierarchies that can be connected to specific points in the common hierarchy when accessing a specific domain, such as military aircraft.

2.3 Coarse-Grain Match

Once the NLP subsystem creates the logical form records for the query, the coarse-grain match selects the nouns and verbs from the records and uses them as pointers into the type hierarchy to determine the index files to examine. It then does a modified set union of the relevant index files incrementing a count for each similar image identifier. Meanwhile, the registration records in the relational database are queried using a Structured Query Language (SQL) SELECT command if applicable attributes are implied in the query. Image identifiers from the first set union are then combined with the resulting image identifiers from SQL commands using the modified set union operation. Those identifiers matching the query on more than k noun and verb classes are candidates for fine-grain matching.

Index Files. Before a query can be processed, the index files must first be created. Index files are generated by determining the noun and verb classes in the caption logical form and adding index records into the corresponding noun and verb index files. These records contain the image identifiers and the roles (i.e., *cases*) a class' lexical tokens play in the caption. The reason to include the *case* is discussed later. At query time, the noun and verb classes from the query logical form determine which index files need to be examined during the match. Then through specialization, the corresponding index files for the subclasses of each class are also determined and examined.

To reduce the size of the index files and eliminate redundancy, the most specialized class in a nominal compound of related subclasses is used to store the index record. From our earlier discussion on the type hierarchy, recall that in the nominal compound "Sidewinder AIM-9M missile," the most specialized class is "AIM-9M." Hence, its index file will contain the image identifier. For the phrase "Sidewinder missile," the image identifier is stored in the "Sidewinder" index file.

Match Criteria. We explain the rationale for the match and the class threshold (a lower bound on the number of classes in the query that a matched caption should have) through an example. We assume the query "Sidewinder mounted on a stand" is being matched against Caption 262865, "3/4 front view of Sidewinder AIM-9R missile on stand." The classes for the query and caption are {"Sidewinder," "assemble,"[1] "stand"} and {"front view," "AIM-9R," "stand"}, respectively. If a class threshold were set to the total number of noun and verb classes appearing in the query, three for this example, the coarse-grain match would fail to consider this caption even though semantically they can be considered the same. Hence, one hypothesis we made was to exclude the verb classes in the calculation of the class threshold.

Consider a different search where the query asks to find "types of Sidewinder missiles." Our intent by this query is to find multiple distinct Sidewinder or AIM-9*x* missiles, not captions containing the term "type." Queries involving "kind of," "version of," "instance of," etc. present the same problem. Similarly, words that indicate the multimedia medium, such as "view of," "picture of," "photograph of," etc., also add little value when matching. We have observed that such phrases appear in only a few captions, as do nouns such as "ACIMD" and "bomb skid" which are better discriminators for determining relevancy. On the other hand, other terms such as "Sidewinder," "F/A-18," range, and aircraft appear in a large number of our captions and only slightly reduce the possible set of relevant images to a query. In lieu of using term-weighting schemes, some of which are described in Salton and McGill [1983], we have introduced an exception list of nouns to be excluded from the match. We then set a lower bound on the class threshold to be the number of noun classes found in the query minus those classes from the exception list.

Using the Semantic Data. Once the class threshold has been computed and the index files identified, the files are combined to produce a list of identifiers with associated counts. The counts indicate the number of nouns and verbs in a caption that match the query. We refer to this process as the "union-count" of sets. However, this approach is not as simple as it appears.

[1] Note that the canonical representation for "mounted" is "assemble" as a result of natural-language processing and logical form creation.

Consider the index files corresponding to the query class "Sidewinder" as shown in Example 2.3.1. Image identifier 29773 appears in three index files—"AIM-9B," "AIM-9C," and "AIM-9D." Should the count for the 29773 identifier be incremented thrice or once? In other words, should Caption 29773 carry more weight than a caption that has only one occurrence of a Sidewinder missile? We could argue that, for this caption to be preferred over one with only one Sidewinder term, the query should explicitly state that a plural number of missiles is desired. However, the implication of a plural noun (e.g., "Sidewinders") could be interpreted as a search for either multiple missiles with different versions (as in Caption 29773) or multiple missiles of the same version (e.g., "AIM-9Cs"). The former situation can be handled by the current index file structure. The latter requires that an additional field be included in the index record to indicate whether the class term used in the caption is singular or plural.

We decided to base the matching on the current file structure for the time being. Our approach to solving this problem was to first employ a set union to remove duplications among the index files for a particular query class (and its subclasses), then combine the sets of each query class while incrementing the count for each identifier appropriately. The set M in Example 2.3.1 illustrates the result.

Example 2.3.1 (*Coarse-Grain Matching Sets*)

Query: 'Sidewinder mounted on a stand':

Keyword files to search: {'Sidewinder', 'Stand'}, K = 2

Subclasses of 'Sidewinder' including itself: }('Sidewinder,' 'AIM-9,' 'AIM-9B,' 'AIM-9C,' 'AIM-9D,' 'AIM-9L,' 'AIM-9M,' 'AIM-9R,' and 'Modified AIM-9C'}

Subclasses of 'Stand' including itself: {'Stand,' 'Pallet,' 'Pedestal,' 'Rack,' 'bomb skid'}

Index file records for subclasses for 'Sidewinder' *Including itself*:

ifile('Sidewinder') = {3204, 5824, 258795}
ifile('AIM-9') = {251701, 251703, 251704, 251706, 251707}
ifile('AIM-9B') = {29773}
ifile('AIM-9C') = {29773}
ifile('AIM-9D') = {29773}
ifile('AIM-9L') = {161082, 182711, 182712,, 182713, 242099, 256393, 256394, 256395}

ifile('AIM-9M') = {216382, 252492, 252494, 252496, 255577, 255578, 255580}
ifile('AIM-9R') = {247181, 247186, 256393, 256394, 256395, 257055, 262865, 262866, 262867,262868, 262869, 262870, 262871, 262872, 262873, 264968}
ifile('modified AIM-9C')={221353, 221354}

INDEX FILE RECORDS FOR SUBCLASSES OF 'STAND' *INCLUDING ITSELF*:

IFILE('STAND') = {29773, 228544, 228545, 228546, 234064, 262865, 262866, 262867}
IFILE('PALLET') = {181761}
IFILE('PEDESTAL') = { }
IFILE('RACK') = {34070, 239097}
IFILE('BOMB SKID') = {10851}

MATCH, M, RESULTS:

{⟨29773,2⟩, ⟨262865,2⟩, ⟨262866,2⟩, ⟨262867,2⟩, ⟨3204, 1⟩, ⟨5824,1⟩, ⟨10851,1⟩, ⟨34070,1⟩, ⟨161082,1⟩, ⟨181761,1⟩, ⟨182711,1⟩, ⟨182712,1⟩, ⟨182713,1⟩, ⟨216382,1⟩, ⟨221353,1⟩, ⟨221354,1⟩, ⟨228544,1⟩, ⟨228545,1⟩, ⟨228546,1⟩, ⟨234064,1⟩, ⟨239097,1⟩, ⟨242099,1⟩, ⟨247181,1⟩, ⟨247186,1⟩, ⟨251701,1⟩, ⟨251703,1⟩, ⟨251704,1⟩, ⟨251706,1⟩, ⟨251707,1⟩, ⟨252492,1⟩, ⟨252494,1⟩, ⟨252496,1⟩, ⟨255577,1⟩, ⟨255578,1⟩, ⟨255580,1⟩, ⟨256393,1⟩, ⟨256394,1⟩, ⟨256395,1⟩, ⟨257055,1⟩, ⟨258795,1⟩, ⟨262868,1⟩, ⟨262869,1⟩, ⟨262870,1⟩, ⟨262871,1⟩, ⟨262872,1⟩, ⟨262873,1⟩, ⟨264968,1⟩}

Using the Relational Data. The matching operation thus far is a straightforward class lookup using a collection of index files. A semantic analysis on the query is also performed at this time to determine if some of the query information might reference the registration data. For example, if a query mentions a location, a mapping function will examine a predefined location field in the registration data using a SELECT statement. The list of image identifiers returned is a set, as the identifier itself is the primary key. Similar analysis is done for dates and can be done for other attributes.

The results of the select can be incorporated into the match by first performing a set union of the SQL-generated image identifiers with the set unions from the preceding section to remove duplication. This set union occurs for only those attributes that can appear in a caption description as well as a fixed-field registration field. For example, a photograph shot at Armitage airfield can have its location stated in the caption description as well as the location field in the registration data. Regardless of where it appears, it is only counted once. The combine process involving the count increment is unchanged.

2.4 Fine-Grain Match

The fine-grain match entails matching the query logical form against the caption logical form using domain knowledge stored in the type hierarchy. The problem is to determine if the caption contains a subset of logical form records matching those in the query with the added caveat that a noun/verb in the query can match a noun/verb in the caption via a specialization (e.g., "Sidewinder" ⇔ "AIM-9R"). The total number of logical form records that the query and caption have in common is referred to as the fine-grain match score, *f*. Issues in determining whether a partial or exact match occur between a query and caption logical forms are beyond the scope of this article. We do wish to point out an interesting situation we encountered when a query contains more information than the caption (see Example 2.4.1). There are actually fewer logical form records in the caption to match against the query. An ideal exact match would have all the query logical form records matching the caption records, but for this scenario, this situation could not occur. However, we believe an end-user who issued this query and got this caption as a result would consider this to be an exact match; hence we make this assumption.

Example 2.4.1 (*More Detailed Query Than Caption*)

QUERY: "MISSILE MOUNTED ON F/A-18 AIRCRAFT"

```
inst(noun(query-1-1), missile).
inst(noun(query-1-2), 'F/A-18').
activity(pastpart(query-1-1), assemble).
theme(pastpart(query-1-1), obj(noun(query-1-1))).
location(pastpart(query-1-1), on(noun(query-1-2))).
```

CAPTION: "SIDEWINDER AIM-9M ON F/A-18A"

```
inst(noun(123456-1-1), 'AIM-9M').
inst(noun(123456-1-2), 'F/A-18A').
location(pastpart(123456-1-1), on(noun(123456-1-2))).
```

The fine-grain match algorithm is divided into four steps. The first step, *instance matching*, involves finding corresponding nouns and verbs in the query and caption. The nouns and verbs are matched based on specializations—that is, a query instance will match a caption instance if the caption instance class is either the same as, or a subclass of, the query instance class. Hence, the query instances form a cover over the caption instances, and the matching proceeds downward, from the query instance nouns and verbs to the caption nouns and verbs.

Direct case relation matching is the second step and involves checking each pairing from the first step to determine if both elements in the pairing have matching slot values. For some *cases*, direct case relation matching involves checking that the case relation names are the same (e.g., theme case) and that destination instances represent two instances being matched. In the query and caption logical forms, this corresponds to matching both arguments in a query record against those in a caption record. For other *cases*, there is a single-term quantity or quality (e.g., "2" or "red") that is to be compared. These values do not represent relationships but are values describing a noun or verb instance itself.

The previous step examined both arguments in a query logical form record to see if a correspondence could be found in the caption. However, situations exist that fall outside this type of checking and that require following some inference chain or path in the logical form. The slot values for each query instance must now be examined to see what values were not

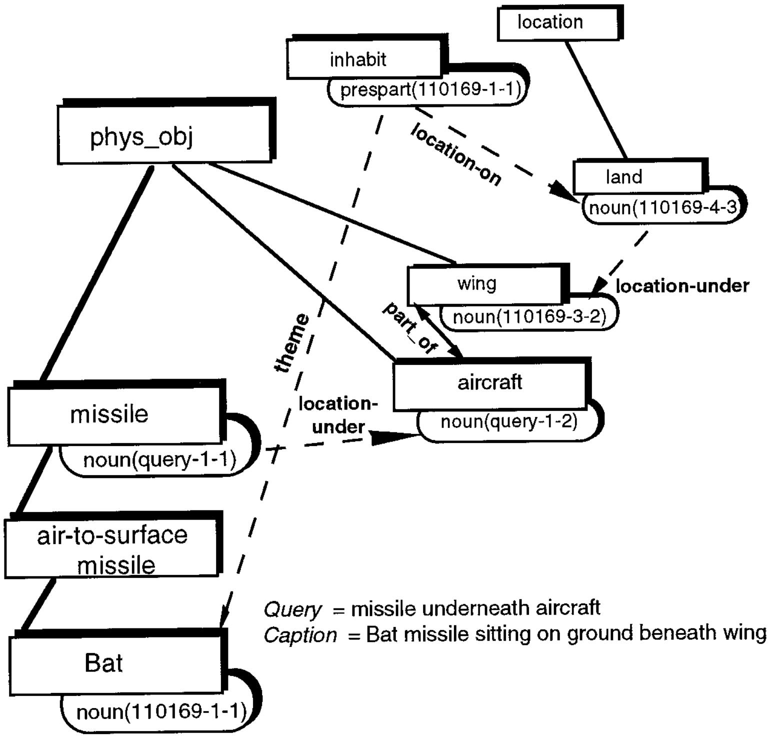


Fig. 4. Matching using correlation links.

matched, and then determine if an inference path can be followed (*indirect case relation matching 1*). The query and caption of Example 2.4.1 illustrate the problem. From a logical form perspective, we are matching query verb-centered cases against caption noun-centered cases.

The last step, *indirect case relation matching 2*, handles the reverse of the previous step, that is, matching query noun-centered cases with caption verb-centered cases. For example, the query reads “missile on F/A-18” with caption “Sidewinder AIM-9M mounted on F/A-18A aircraft.” Another situation involves checking for correlations, such as part-of and owned-by. For the caption “NAF airfield. Overview of Cold Line and Hot Line,” the query “Hot Line at NAF” will match the caption based on existing correlation information that Hot Line is a part-of NAF. For the caption “Bat missile sitting on ground beneath wing” with query “missile underneath aircraft” (shown in Figure 4), two location logical forms are followed before discovering the part-of correlation. This type of transversal can become lengthy and requires distance bounds be implemented to constrain search and match times. We have not incorporated these distance bounds at this time.

Most logical form records containing relationships can be considered to consist of two parts, the *case relation* (e.g., location) and a *case relation modifier* (e.g., on), whereas others consist of just a case relation or correlation (e.g., quantity or part-whole). In matching a query relationship against a caption one, the existence of like case relations is first checked. If they exist and are found to be related to query and caption instances being matched, then a relationship weight can be computed for adding to the match score *f*. The relationship weight can be viewed as consisting of two fractional parts whose sum is 1.0; the case relation modifier provides one half, and the destination noun or verb instances provide the other half. This allows fractional scores to be assigned to relationships, resulting in fractional match scores. For a case relation modifier, checks for exact matching (e.g., "on" ⇔ "on") or similarity matching (e.g., "on" ⇔ "at") can be made. Similarity matching requires analysis on how the nouns and verbs in a relationship may interact, and such a discussion is beyond the scope of this article.

For presenting results to the user, there are two ways to compute the final score. One method counts the number of records that the query and caption have in common. An alternative approach is to present the difference between the number of records that matched and the number of records in the query where a difference of zero indicates an exact match, and anything greater indicates the number of records that failed to match.

2.5 Semantic Match Breakdown

We have seen that keyword matching based on class/subclass relationships allows a user to not have to remember every specific keyword. Any user can use specific terms when they are known or general terms when they are either unknown or if the user wishes to explore a wider range of concepts. Enhancing the system to recognize different potential patterns during the fine-grain matching is a matter of adding the appropriate rules. These either can be incorporated under classes in the type hierarchy as we have done presently or into scripts or other high-level knowledge structures as we have seen in the past. However, there are situations where a match is just not possible no matter how many rules are used. These situations are encountered when the user query contains either more or a differing perspective of detail than what is specified in the caption. For example, one person's interpretation of the content of an image may be entirely different than another's, resulting in two dissimilar caption descriptions. We will see an example of this shortly.

3. EXPERIMENT

The Photo Lab's current retrieval system uses keyphrase records to index the caption data. The keyphrase records are manually created by the Photo Lab retrieval specialist for each caption. Formulating a keyphrase query involves filling in the appropriate keyword and descriptors. SQL queries can be applied against the registration data and joined with a search of the keyphrase records. Searching the registration data can employ an index if one exists for the requested attributes. Because the caption attribute involves free-form text, searching it currently requires creating a regular expression with a sequential search. Multiple images may be grouped

Table I. Caption and Key Phrases for Images 262865-73

Caption	Sidewinder AIM-9R Missile on a Stand and Views Mounted on an F/A-18C BU# 1632 Aircraft, Nose 110, LHL 262867-68 were Released by L. King on 7-24-90.
Keyphrase 1	Aircraft F/A-18C partial Sidewinder AIM-9R AF
Keyphrase 2	Aircraft F/A-18C Sidewinder AIM-9R AF
Keyphrase 3	Missile Sidewinder AIM-9R F/A-18C AF
Keyphrase 4	Missile Sidewinder AIM-9R Stand AF
Keyphrase 5	Program Sidewinder AIM-9R

together based on a customer's photo shoot request. Table I shows the caption and keyphrase records for the set of images 262865 through 262873. Because of the complexity of the keyphrase structure, a customer seeking an image must provide the retrieval specialist with specific type(s) of aircraft, missiles, events, etc., from which the specialist formulates the keyphrases.

To test our ideas, we created an experimental database by randomly selecting 120 publicly released captions (and associated images) from the Photo Lab image database. They encompassed a diversity of caption structures and styles. As stated earlier, we were attempting in-depth understanding of all words encountered in the captions. The objective was to better understand the magnitude of the natural-language processing necessary for high levels of accuracy in information retrieval. To accomplish this objective, the first problem we encountered was that the captions were written by many people over the years. When the original retrieval system was created, images from a photo shoot were often grouped together, and a summary for the shoot was written to conserve space. No policy or guidelines were provided for writing the summaries, and no one provided any editing function with respect to grammatical constructs and spelling. Attempting to parse and understand these summaries would have been extremely difficult, as the following example shows:

AV-8B NIGHT ATTACK BU# 162966 AIRCRAFT, NOSE 87, NIGHT HAWK LOGO, TAIL 162966 MARINES, MAD 'MARINE AIR DETACHMENT'. SUNSET AND SPECIAL LIGHTING. SIDE AND CLOSEUP OF NOSE AND COCKPIT, SIDE VIEW, EXCELLENT.

Further examination revealed that the original hand-written captions still existed and that their structure was easier to parse and understand as they pertained to only one image at a time. There were still a small number of idiosyncrasies that had to be overcome because of the numerous writing styles of the photographers. But by and large, most of these captions could be used with very few cleanups. The resulting set consisted of 217 individual captions (for 217 images) containing 413 distinct sentences and noun phrases. The lexicon used by the natural-language processing program contained 1942 words, and the type hierarchy, implemented separately, consisted of 831 noun and verb classes. For the 217 captions, the natural-language processing subsystem and database creation process produced an average of 19.185 logical form records per caption and 468 noun index files with an average of 3.412 index records per file.

We believe we have identified a majority of the basic classes (e.g., aircraft, missile) needed to support the military domain. New classes that would need to be created to handle new captions are mostly proper nouns for specific weapons systems, locations, etc. (e.g., "F-22," "Panamint Valley"). We have begun work on replacing our lexicon with a standard machine-readable dictionary such as that provided by WORDNET while also including the military terminology.

Once the semantic database was populated from the captions, we created 19 debugging queries based on the types of queries we expected users to ask as well as ones we believed would present problems for the natural-language processing subsystem and match routines. Example 3.1 shows a debugging query with two captions satisfying the query. Also shown are the logical form records for the query and second caption. The Maximum Possible Keyword Score is based on the number of nouns appearing in the sentence not counting the nouns on the list of excluded nouns mentioned in Section 2.3. The Maximum Possible Concept Score is based on the number of logical form records in the query. As the results show, a match of only three records between the query and the captions was computed. As it turns out, three is also the maximum score that could be produced for this query without performing a transformation to verb-centered cases. "Assemble" activities with accompanying themes could be inferred for both captions and transformations made accordingly.

Example 3.1 *Sample Test Query*

QUERY: "SIDEWINDER MOUNTED ON A STAND"

```
theme(pastpart(query-1-1),obj(noun(query-1-1))).
location(pastpart(query-1-1),on(noun(query-1-2))).
activity(pastpart(query-1-1),assemble).
inst(noun(query-1-1),'Sidewinder').
inst(noun(query-1-2),stand).
```

MAXIMUM POSSIBLE KEYWORD SCORE = 2.0, MAXIMUM POSSIBLE CONCEPT SCORE = 5.0

[3.0] SIDEWINDER 1A AND SIDEWINDER 1C MISSILES. COMPARISON OF SIDEWINDER 1A AIM 9B, SIDEWINDER 1C IRAH AIM 9D, AND SIDEWINDER 1C SAR AIM 9C.3/4 RIGHT SIDE VIEW ON STAND.

[3.0] 3/4 FRONT VIEW OF SIDEWINDER AIM 9R MISSILE ON STAND.

```
location('noun(262865-1-4)',on('noun(262865-1-5)')).
inst('noun(262865-1-4)','AIM-9R').
quantity('noun(262865-1-1)','3/4').
theme('noun(262865-1-1)',obj('noun(262865-1-4)')).
```

inst(‘noun(262865-1-1)’,‘front view’).
inst(‘noun(262865-1-5)’,stand).

Another debugging query is shown in Example 3.2. Note that “right side view” in the first caption does not refer to any object explicitly. The view is inferred to pertain to the previous subject. By anaphoric reference resolution, “all aircraft” is associated with all previous aircraft encountered in the sentence as the subject. Because the EA-6B is a kind of A-6, there is a match. In the second caption, because the pod is located on the A-6A aircraft and a side view of the pod is stated, the side view can be inferred to pertain to the aircraft as well. These queries and others were used to help define the transformation and inference rules.

Example 3.2 (*Sample Test Query Using Inferencing*)

Query: “Side View of an A-6”
Maximum Possible Keyword Score = 2.0, Maximum Possible Concept Score = 3.0

[3.0] air to air view of four plane formation. Clockwise from left to right: VX-5’s ea-6b, NWC’S A-7E, NWC’S A-6E, and NWC’S F/A-18A. all aircraft Loaded with AGM-88 harm. overhead right side view.

[3.0] Expendable Seeker Simulator (ESS) Installed in Pod on A-6A Aircraft. Full Side View of Pod with Personnel.

Once our testing was completed, we presented a Photo Lab retrieval specialist with the original 120 caption summaries, not the individual 217 captions used to create the database. We then directed the retrieval specialist to construct example English queries representative of the types of queries posed to them by the Center’s end-users. The retrieval specialist devised 46 test queries. Of these 46 queries, 16 required that additional specifics be solicited from any end-user to effectively utilize the existing keyphrase system, as the queries themselves were too general. In addition, two additional queries could not be handled by MARIE, as they were both registration type information that could be handled more appropriately in a separate table. As they currently exist, this information is improperly embedded in the keyphrase description field (the specific case is the pattern {R} at the end of a keyphrase which indicates the image was released). We believe that even though the retrieval specialist saw the original 120 captions, the majority of these were caption summaries which were as poorly structured as the ones shown earlier. In addition, the fact that the retrieval specialist created 16 queries that could not be handled by the keyphrase system indicated to us that the retrieval specialist wanted to see if we could overcome some of the limitations posed by the current keyphrase system. The following examples illustrate some of the situations encountered.

In Example 3.3, no keyphrase can be constructed, as XXXX is the missile type and must be explicitly supplied by the end-user. However, MARIE can process the statement as is and produces the caption shown, with a score of 5.5.

Example 3.3 (*Photo Lab Query Having No Corresponding Keyphrase*)

QUERY: "TRAINING MISSILES ON A SKYHAWK"

MAXIMUM POSSIBLE KEYWORD SCORE = 3.0, MAXIMUM POSSIBLE CONCEPT SCORE = 6.0

KEYPHRASE: AIRCRAFT A-4* XXXX

MARIE PRODUCED THE CAPTION:

[5.5] AIR TO AIR VIEW OF A-4M BU# 160264 SKYHAWK II AIRCRAFT (2ND MAW/ MARINES) WITH TWO LASER MAVERICK AGM-65C USAF TRAINING MISSILES. AIRCRAFT FLYING OVER HIGH SIERRAS. SIDE VIEW.

Certain queries required the examination of registration information. In Example 3.4, there is no keyphrase that can be used based on the existing keyphrase constructs available. To find the images, a SELECT statement must first be formulated to retrieve the *Location* and *Date-Origination* information based on the values supplied (i.e., Snort and 2-23-81); then each record retrieved must be examined. MARIE searches just as before, but in addition also formulates a SELECT query to examine the two preceding fields based on the existence of the prepositions "at" and "on" and the arguments to each. Note that for MARIE to process this query, the query needed to be restated.

Example 3.4 (*Photo Lab Query Using Registration Data*)

QUERY: "TP 1314 SYNCHRO FIRING AT SNORT ON 2-23-81"

MAXIMUM POSSIBLE KEYWORD SCORE = 4.0, MAXIMUM POSSIBLE CONCEPT SCORE = 9.0

KEYPHRASE: SELECT * FROM VISUAL WHERE LOCATION LIKE 'SNORT' AND DATE-ORIG = 2-21-31;

MARIE REQUIRED RESTATING OF QUERY TO: "TP 1314. SYNCHRO FIRING AT SNORT ON FEB 23, 1981."

[9.0] TP 1314. A-7B/E DVT-7 (250 KEAS) ESCAPE SYSTEM TEST (RUN 2). synchro firing at 1090'N × 38'W. dummy just leaving sled.

For certain queries, we discovered that the results produced by MARIE were no better than a keyword search based on types. In Example 3.5, a keyphrase cannot be used because the user must explicitly specify the type of aircraft. MARIE finds those captions containing both a type of aircraft and Armitage airfield. However, it fails to make the "flying over" connection. In the first caption, the relative position of the aircraft is below that of the airfield based on the camera's location. In the second caption, the phrase "over Armitage" is stored in the location case of the NA-3B aircraft; there is nothing for "flying" to match.

Example 3.5 (*Photo Lab Query Resulting in MARIE Producing the Same Results as the Keyphrase System*)

QUERY: "AIRCRAFT FLYING OVER ARMITAGE AIRFIELD"

MAXIMUM POSSIBLE KEYWORD SCORE = 2.0, MAXIMUM POSSIBLE CONCEPT SCORE = 5.0

KEYPHRASE: AIRCRAFT XXXX ARMITAGE

MARIE PRODUCED THE CAPTIONS:

[2.0] AIR TO AIR VIEW OF QF4H-1 BU# 149452 DRONE. DRONE PAINTED SILVER WITH RED ON NOSE, RED UNDER WINGS, RED ON TAIL, AND ON RED FIN. FULL SIDE VIEW. COMPLETE VIEW OF ARMITAGE FIELD ABOVE AIRCRAFT.

[2.0] AIR TO GROUND VIEW OF PROGRAM NA-3B, AAR-47 ELECTRO OPTICAL DEVICE TESTION ON NA-3B BU# 142630 AIRCRAFT OVER ARMITAGE, NA-3B IN CENTER OF VIEW. ALL THREE HANGARS LOOKING WEST. EXCELLENT VIEW OF FIELD AND MOUNTAINS.

In Example 3.6, two separate keyphrase records can be used, but a lot of domain knowledge on the part of the retrieval specialist was required to formulate the appropriate keyphrases. For this query, the retrieval specialist knew that BTV was associated with the Skyray program and that only TA-7C aircraft were used. Notice in the second caption returned that MARIE fails to make the connection that station 7 is a position under the aircraft, resulting in a keyword score only.

Example 3.6 (*Photo Lab Query Requiring Domain Knowledge for Keyphrase*)

QUERY: "BTV on an A-7 aircraft"

MAXIMUM POSSIBLE KEYWORD SCORE = 2.0 MAXIMUM POSSIBLE CONCEPT SCORE = 3.0

KEYPHRASES: MISSILE WALLEYE 1 BTV TA-7C SKYRAY FIVER OPTICS POD
POD SKYRAY FIVER OPTICS TA-7C WWALLEYE BTV

MARIE PRODUCED THE CAPTIONS:

[3.0] PROGRAM SKYRAY. AIR TO AIR VIEW OF TA-7C BU# 156738 AIRCRAFT (NOSE 700). SKYRAY FIBER OPTICS INSTRUMENTATION POD AND WALLEYE BTV (BALLISTICTIC TEST VEHICLE) ARE UNDER WING. CLOSEUP RIGHT SIDE VIEW OF AIRCRAFT.

[2.0] PROGRAM SKYRAY. AIR TO AIR VIEW OF TA-7C BU# 156738 AIRCRAFT (NOSE 700). FULL SIDE VIEW OF AIRCRAFT. HS CAMERA AT STATION 6. WALLEYE BTV (BALLISTIC TEST VEHICLE) AT STATION 7. SKYRAY INSTRUMENTATION POD AT STATION 8. MK 82 BOMB (INERT) ON LEFT WING.

To further test the system, we solicited new queries from four Center end-users: a secretary, a programmer familiar with test range operations, an engineer familiar with missile design and development, and a program manager. Each member was given the same stack of 100 images picked randomly by us out of the 217. This was done to ensure some overlap in the queries to see how different individuals would ask for the same image and provide for a little diversity. They were asked to select 20 images at random and formulate an English query for each image reflecting what they would enter to find that image. They were allowed to look at the original 120 captions for those cases where they did not know what a particular type of object was, for example, an A-7 versus an F/A-18. The resulting 80 queries did not require any modifications to the lexicon or type hierarchy. Example 3.7 shows a caption, the corresponding keyphrases used to index the image, and three different user queries. A retrieval specialist was asked to identify from which 80 user queries keyphrases could be generated, as Center users could not access the keyphrase system directly and must use the retrieval specialist. The bias in this experiment in favor of the keyphrase system was a degree of familiarity with our test database by the retrieval specialist which could result in a higher score for the keyphrase system.

Some of the queries provided did not look like queries, but rather better caption descriptions than the ones already existing (there was no image selected by all four users). The queries provided by the secretary were the closest to what the retrieval specialist used; they were mostly short noun phrases. The queries supplied by the programmer were short sentences, that is, noun, verb, object/location. Queries supplied by the engineer and program manager were the most descriptive using more adjectives-t han either of the previous two users. Although this is by no means representative of all four types of people, it did show that verbs are used more often than long descriptive noun phrases when querying the database.

Example 3.7 (*Multiple User Queries for the Same Caption*)

CAPTION:

PICTORIAL. KERN RIVER VIEWS. LOOKING DOWN RIVER. SMALL GIRL ON ROCK.

KEYPHRASES: GEOLOGICAL ROCK EROSION KERN RIVER
PICTORIAL KERN RIVER
WATER RIVER KERN

QUERIES: RIVER VIEW (SECRETARY)
A GIRL SITTING BY A RIVER (PROGRAMMER)
A DEEP BLUE RIVER FLOWING THROUGH WOODED MOUNTAINS WITH A GIRL SITTING ON A ROCK (PROGRAM MANAGER)

In the last query, notice that there is no mention of "wooded mountains" in the caption description although they actually appear in the image. Example 3.8 provides further illustrations. This lack of information in the captions results in less possible matches, and we believe there is little that can be done with the existing captions to resolve this. Part of the differences may be lessened in the future if the depictability inferences in Rowe [1994] are included or if caption descriptions can be automatically generated via image processing. However, manual intervention is still needed to adequately describe the objects, for example, USAF F-4 aircraft versus USN F-4 aircraft.

Example 3.8 (*User Queries Specifying Information Not Contained in the Captions*)

CAPTION:

USAF AGM-65C LASER MAVERICKS HITTING TANK TARGETS AT EGLIN AFB. VIEW OF TARGET EXPLODING.

QUERIES: AN EXPLOSION IN A FIELD WHICH TURNS TO BLACK SMOKE
A WARHEAD FIREBALL UPON HITTING A GROUND TARGET

CAPTION:

USS FORRESTAL CVA-59 IN GULF OF TONKIN. SMOKE POURING FROM SHIP. USS REPERTUS DD-851 AND HELICOPTER STANDING BY.

QUERIES: FIRE ON AIRCRAFT CARRIER
A HELICOPTER FLYING TOWARD A BURNING AIRCRAFT CARRIER

CAPTION:

HELIOSTAT. DIRECT SOLAR INTENSITY STUDIES. PREPARING TARGET FOR HOSTING.

QUERY: A PLATFORM HANGING FROM A TOWER.

With respect to the overall implementation, MARIE was designed and developed using the client-server model to keep the size of each process small, to enable parallel processing, and to enhance modularity and reusability. The system took approximately 12 man-months to implement, largely because we were able to modify an existing natural-language processing program. In addition to the approximately 11.5K lines of PROLOG code reused from this program, we added approximately 7K lines of PROLOG code to implement our logical form and type hierarchy methodology. The coarse- and fine-grain matching routines required roughly an additional 2K lines of PROLOG code.

4. RESULTS

To evaluate the retrieval effectiveness and search efficiency based on information retrieval measures, the evaluation criteria presented in Salton and McGill [1983] were used—namely, *recall, precision, time, effort, presentation*, and *coverage*. The presentation, that is, the user interface, was shown in Figure 1. The collection coverage of relevant information is limited to those images taken at the Center in the last 50 years and any that may have been provided by other Government facilities to the Center. Recall and precision measures are computed for the 46 Photo Lab queries using both micro- and macroaveraging techniques as defined in Tague-Sutcliffe [1992] (those formulas are restated here).

$$m_{p1} = \sum r_i / \sum n_i \qquad m_{p2} = [\sum (r_i/n_i)]/q$$

$$m_{r1} = \sum r_i / \sum t_i \qquad m_{r2} = [\sum (r_i/t_i)]/q$$

where
— m_{p1} and m_{r1} are the microaverages for precision and recall, respectively
— m_{p2} and m_{r2} are the macroaverages for precision and recall, respectively
— r_i = number of relevant and retrieved documents for the ith query
— n_i = number of retrieved documents for the ith query
— t_i = number of relevant documents for the ith query and
— q = number of queries.

The formula for macroaveraging as provided in Tague-Sutcliffe [1992] does not address the issue when $n_i = 0$, i.e., when there are no retrieved documents for the ith query. However, for cases where an individual $n_i = 0$, r_i is also equal to 0. Hence, we will set the ratio to 0. For the keyphrase system, results are not displayed using a rank position or coordination level [van Rijsbergen 1979]; they are simply listed.

To get our data, the queries described previously were run through MARIE. Keyphrases were also manually created by a Photo Lab retrieval specialist from the test queries. Relevancy data were obtained from manual comparison by the retrieval specialist of the queries against the test set of 217 captioned images. Images were considered one at a time, and queries were compared against the chosen image.

MARIE displays results based on a match score with those image identifiers that have the highest number of logical form records matching the query being displayed first. Let h_q be the number of logical form records in the query. We retrieve those captions having values of f where $f \in [h_q, h_q - 1)$ if $h_q > k$ (i.e., f is in the closed-open interval from h_q to $h_q - 1$) or $f \in [h_q, h_q]$ if $h_q = k$ based on the definitions of f and k defined earlier. Hence, we display only those matched records whose match score equals or is one less than the total number of query logical form records except when this score is equal to the maximum possible keyword score. Our rationale is that MARIE should offer more than just enhanced keyword performance. The distance of 1 in the interval was chosen arbitrarily. Based on this definition, the captions from Example 3.1 would not be retrieved even though semantically they match. The problem stems from the distance of 1 chosen arbitrarily in the interval. Had the distance been 2 instead of 1, then the caption would have appeared in the results list.

Some caption matches in the interval $[h_q, k)$ for $h_q > k$ have also been considered relevant by the Photo Lab, but fall below our retrieval threshold. Recognizing the fact that the natural-language processing subsystem may not provide totally accurate logical forms for both captions and queries as well as the problem of lacking information in the captions, we hypothesize that in some cases the Photo Lab may accept the highest scoring caption(s) as being the most relevant, provided it exceeds or meets the class match score. Instead of treating the entire interval $[h_q, k)$ for $h_q > k$ as determining what is retrieved, we restrict the interval as follows. Let $F = f$ for each query. We then retrieve those captions based on values of f where $f \in [F, F - 1)$ if $F > k$ or $f \in [F, F]$ if $F = k$; that is, the highest match score computed (and that which is one less) determines what is retrieved. For the case where $F = h_q$, we revert back to the original definition. Plotting the score for this definition against the previous one reveals some of our problems in generating complete logical form records.

Recall and precision scores are shown in Table II and Figure 5 for the keyphrase system as well as the two MARIE evaluations: $f \in [h_q, h_q - 1)$ or $[h_q, h_q]$ in the second row and $f \in [F, F - 1)$ or $[F, F]$ in the third row. The figure shows that the averages for Photo Lab's 46 queries increase as the retrieval threshold is set to reflect the highest match score instead of

Table II. Precision and Recall Scores

Queries	m_{p1}	m_{r1}	m_{p2}	m_{r2}
Current Keyphrase Approach	0.660	0.493	0.556	0.571
MARIE Based on h_q Matches	0.724	0.717	0.614	0.649
MARIE Based on F Matches	0.810	0.818	0.761	0.774

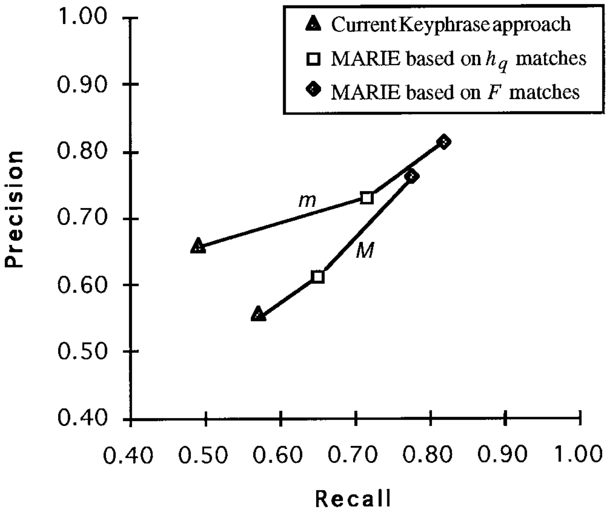


Fig. 5. Micro- (m) and macroaverages (M).

the total number of query logical form records. The results show that we obtained an increase of 30% in average precision and 50% in recall over the keyphrase approach. Although we may not have fully understood the queries and captions as captured in the logical form records, both recall and precision are improved by using natural-language processing techniques. Regardless of whether we are using a small or large collection of captions in our test database, the accuracy of the logical form records is the more paramount issue for increasing overall performance.

The results from the second set of tests where four users randomly selected 20 images and generated 20 queries to retrieve them are shown in Table III and Figure 6. The table and figure show that for the 20 queries, both the keyphrase system and MARIE are unable to retrieve all 20 corresponding images. This occurs because either additional or different information is specified in the query than that actually written by the original photographer in the caption.

The response time for a query is the amount of time it takes the NLP subsystem to produce the logical forms plus the search/match times. Response time measures for MARIE are shown in Table IV for a network of SPARC 2 workstations running SunOS and the Oracle database management system in an isolated subnetwork with a database of 217 captions. Row 1 indicates the number of words in a query. Row 2 indicates the number of captions that were retrieved per query. Row 3 indicates the

Table III. Number of Relevant Images Retrieved Out of 20 for 20 User Queries

Queries	Secretary	Programmer	Engineer	Program Manager
Current Keyphrase Approach	10	9	9	4
MARIE Based on h_q Matches	10	3	5	1
MARIE Based on F Matches	16	10	7	3

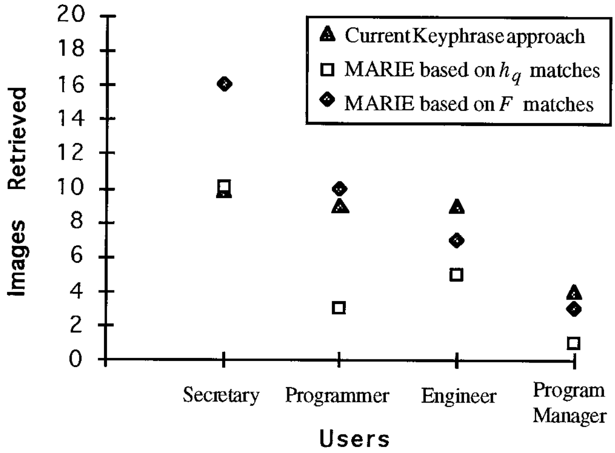


Fig. 6. Images retrieved for four different users.

Table IV. Parse Plus Match Response Time Measures

Queries	Mean	Median	Standard Deviation
Words in a Query	4.532	5.0	1.797
Captions Reviewed per Query	6.915	2.0	20.920
Relevant Captions per Query	2.152	1.0	2.319
NL Parse Time	3.431	3.084	1.493
Coarse-Grain Match Time	4.696	1.826	5.411
Fine-Grain Match Time (Same WS)	20.231	4.353	45.862
Fine-Grain Match Time (Separate WS)	10.855	2.964	23.042
Fine-Grain Match Time (2 WS)	6.166	2.217	11.537
Fine-Grain Match Time (3 WS)	4.892	2.254	7.928
Fine-Grain Match Time (4 WS)	4.182	2.111	6.098
Fine-Grain Match Time (5 WS)	3.930	2.049	5.145

WS = Workstation

number of relevant captions per query. Row 4 shows the query natural-language parsing time in real-time seconds. Row 5 indicates the real-time seconds to perform the coarse-grain match. Row 6 indicates the real-time seconds the fine-grain match took to run where the user interface, NLP subsystem, and one fine-grain match processes were all executing on the same workstation. Rows 7 through 11 show the real-time seconds for the fine-grain processes where the user interface and the NLP subsystem were executing on one workstation, and the fine-grain match processes were executing on one to five different workstations allowing some degree of parallelism in the match. The database was stored on a separate file server and accessed via the Network File System (NFS).

These times are considerably better than those for the existing keyphrase system used by the Photo Lab. Currently, it typically requires three minutes for personnel to extract sufficient information from a user to formulate a query, since the existing computer system only indexes a few of the possible words and phrases a user could use (for instance, "aircraft" but not "airplane"), and the query needs to be rephrased until it contains them. This is then followed by a lookup process averaging about five minutes, in which portions of captions exactly matching one or more of the keywords or keyphrases are presented to the user for approval, with staff using their years of experience to assist in ruling out false hits. Computer printouts (prepared offline) are commonly used and the computer directly consulted only on occasion. Thus a routine query takes about eight minutes. In fact the mean time is longer, because perhaps 10% of the time a phone call is required when the query is sent by mail or courier and lacks necessary clarifying information. So the existing Photo Lab system requires significant rewriting of queries to get them into a form acceptable to a rather restrictive keyphrase matching system. MARIE can work much faster by rewriting queries automatically.

We speculate that as we scale up to a larger database, we can be more selective in deciding what we are going to match by setting a higher-class threshold for the coarse-grain match. As we increase the number of workstations, however, we will have problems accessing one file system over NFS. This increase will force us to distribute the database over more than one server. We currently are using one scheduler to distribute the fine-grain match work to the workstations based on an initial distribution of work to all workstations, followed by waiting for whichever workstation finishes first and sending it the next match request. Using additional schedulers may not be advantageous, as the scheduler's list of image identifiers to be matched is stored in memory, and any waiting that occurs is on a socket waiting for a response.

The present keyphrase system is also not accessible to the vast population of Center personnel, as the rules and attribute values for formulating the keyphrases are deemed too cumbersome for the naive user. Hence, we have no way to measure the amount of effort from the query user's perspective. Even though systems exist that allow a user to navigate through keyphrase hierarchies [Ragusa and Orwig 1990], the user must still formulate a keyphrase and restrict himself to the rigidity of the approach. Our goal has always been for the user to enter the same query as is verbally spoken to the retrieval specialists. As query users, the retrieval specialists themselves prefer the MARIE approach.

Another measurement of performance found in Mauldin [1991] is maintenance, the amount of human labor required to keep the system operational. For the keyphrase system, the retrieval specialists must manually create keyphrase(s) that they believe will best help to locate the records;

registration information, including captions, still needs to be entered. Most of this expertise comes from experience of previous queries encountered and how the present keyphrases are written. For example, a user querying information about an A-7 may need to search on A-7C and NA-7B as well. In the case of MARIE, lexicon definitions with accompanying type hierarchy information need to be created for new words, but this is easier to provide. Correlation information about new objects need be entered only when they differ from default values. As an example, all types of aircraft inherit the property that they have wings and cockpits, but only attack aircraft carry missiles and bombs. To alleviate some of these problems, we have used those type hierarchy concepts found in standard sources such as in the NASA Thesaurus [1988] and the Defense Technical Information Thesaurus [1990], as well as those defined locally in the NAWCWPNS Authority List used in the Center's Technical Library. Our use of WORDNET in the future to provide online lexicon access is expected to reduce most of the routine clerical work now done.

There are several shortcomings of our approach. First and foremost, the natural-language processor is predominately a syntactic approach and limited in its capabilities for handling arbitrary English constructs, thus requiring query restating. Second, there is no spelling checker and correction mechanism. Further, our choice of distinguishing proper nouns from common nouns by the use of capitalization has been more irritating than beneficial. Another shortcoming is the inability to analyze the query statement first and suggest improvements, such as in the way the query is phrased, emphasizing a greater use of verbs instead of numerous prepositions, elimination of redundancy (e.g., using just "AIM-9R" instead of "AIM-9R missile"), etc. Also, there is no user model with which to record a user's interaction with the system for further analysis as has been pursued by fellow researchers.

As a future enhancement, we mentioned earlier that each index file record contains the *image identifier* and *case*. This case information was included to eventually increase selectivity when performing the coarse-grain match. For example, when a query is entered, the user could specify the case for a word. This information could be used to filter the index records. Thus, if a search involves those captions that have an F-4 aircraft as the *destination* of a missile launch, those captions in which the F-4 aircraft was actually firing a missile (agent) or merely being a place for some other event (location) would be filtered out.

5. CONCLUSIONS

We have seen that our decision to use English queries and captions and the effectiveness of the matching strategy depends largely on the amount of natural-language processing performed. One could argue that simpler traditional methods might yield similar or slightly better performance and not require any natural-language processing at all. However, we have not encountered any such system that provides perfect recall and precision.

Further, such a system, even if it uses a graphical user interface, may be totally inadequate if such an interface cannot be accessed and interacted with easily. Although natural-language processing has shown high levels of recall and precision in limited domains, the real issue is using the same techniques across multiple domains. Future work will involve experimenting with other natural-language-processing strategies as we integrate different domains.

ACKNOWLEDGMENTS

We would like to acknowledge William F. Ball at the Naval Air Warfare Center Weapons Division without whose support this work would not have been possible. We would also like to acknowledge the support of Lynn Cowley who was the primary retrieval specialist at the Center's Photo Lab.

REFERENCES

ALLEN, J. 1987. *Natural Language Understanding.* Benjamin-Cummings, Menlo Park, Calif.

ANICK, P. G. 1991. Integrating "Natural Language" and Boolean query: An application of computational linguistics to full-text information retrieval. In *Proceedings of the AAAI-91 Workshop on Natural Language Text Retrieval.* AAAI, Menlo Park, Calif.

CHARNIAK, E. J. AND MCDERMOTT, D. 1987. *Introduction to Artificial Intelligence.* Addison-Wesley, Reading, Mass.

DLA. 1990. *Defense Technical Information Center Thesaurus.* AD-A226000, Defense Logistics Agency, Alexandria, Va.

DICK, J. P. AND HIRST, G. 1991. Intelligent text retrieval. In *Proceedings of the AAAI-91 Workshop on Natural Language Text Retrieval.* AAAI, Menlo Park, Calif.

FININ, T. W. 1986. Constraining the interpretation of nominal compounds in a limited context. In *Analyzing Language in Restricted Domains: Sublanguage Description and Processing.* Lawrence Erlbaum, Hillsdale, N.J., 163–173.

GAY, L. S. AND CROFT, W. B. 1990. Interpreting nominal compounds for information retrieval. *Inf. Process. Manage. 26,* 1, 21–38.

GUGLIELMO, E. J. 1992. Intelligent information retrieval for a multimedia database using captions. Ph.D. dissertation, Naval Postgraduate School, Monterey, Calif.

HAAS, S. W. 1991. A feasibility study of the case hierarchy model for the construction and porting of natural language interfaces. *Inf. Process. Manage. 26,* 5, 615–628.

KATZ, B. 1988. Using English for indexing and retrieving. AI Memo 1096, MIT, Cambridge, Mass.

KOLODNER, J. L. 1983. Indexing and retrieval strategies for natural language fact retrieval. *ACM Trans. Database Syst. 8,* 3 (Sept.), 434–464.

KROVETZ, R. AND CROFT, W. B. 1992. Lexical ambiguity and information retrieval. *ACM Trans. Inf. Syst. 10,* 2 (Apr.), 115–141.

MAULDIN, M. L. 1991. *Conceptual Information Retrieval: A Case Study in Adaptive Partial Parsing.* Kluwer Academic, Norwell, Mass.

MILLER, G., BECKWITH, R., FELLBAUM, C., GROSS, D., AND MILLER, K. 1990. Five papers on Wordnet. *Int. J. Lexicogr. 3,* 4 (Winter).

MINSKY, M. 1981. A framework for representing knowledge. In *Mind Design,* J. Haugeland, Ed. The MIT Press, Cambridge, Mass., 95–128.

MONTGOMERY, C. A., BURGE, J., HOLMBACK, H., KUHNS, J. L., STALLS, B. G., STUMBERGER, R., AND RUSSEL, R. L., JR. 1989. The DBG message understanding system. In *Proceedings of the Annual AI Systems in Government Conference* (Washington, D.C., Mar. 27–31). IEEE Computer Society Press, Los Alamitos, Calif.

NASA. 1988. *NASA Thesaurus, Hierarchical Listing.* NASA SP-7064, NASA Scientific and Technical Information Division, Washington, D.C.

RAGUSA, J. AND ORWIG, G. 1990. Attacking the information access problem with expert systems. *J. Expert Syst. 4* (Winter), 26–32.

RAU, L. F. 1987. Knowledge organization and access in a conceptual information system. *Inf. Process. Manage. 23,* 4, 269–284.

ROWE, N. C. 1994. Inferring depictions in natural-language captions for efficient access to picture data. *Inf. Process. Manage. 30,* 3, 379–388.

SAGER, N. 1986. Sublanguage: Linguistic phenomenon, computational tool. In *Analyzing Language in Restricted Domains: Sublanguage Description and Processing,* Lawrence Erlbaum, Hillsdale, N.J., 163–173.

SALTON, G. AND MCGILL, M. J. 1983. *Introduction to Modern Information Retrieval.* McGraw-Hill, New York.

SCHANK, R. 1977. *Scripts, Plans, and Goals.* Lawrence Erlbaum, Hillsdale, N.J.

SCHANK, R. 1975. *Conceptual Information Processing.* Elsevier Science, New York.

SEMBOK, T. M. T. AND VAN RIJSBERGEN, C. J. 1990. SILOL: A simple logical-linguistic document retrieval system. *Inf. Process. Manage. 26,* 1, 111–134.

SMEATON, A. F. 1992. Progress in the application of natural language processing to information retrieval tasks. *Comput. J. 35,* 3, 268–278.

SRIHARI, R. AND RAPAPORT, W. 1989. Combining linguistic and pictorial information using captions to interpret newspaper photographs. In *Current Trends in SNePS—Semantic Network Processing System.* In Lecture Notes Artificial Intelligence, vol. 437. Springer-Verlag, Berlin, 85–96.

TAGUE-SUTCLIFFE, J. 1992. The pragmatics of information retrieval experimentation, revisited. *Inf. Process. Manage. 28,* 4, 467–490.

VAN RIJSBERGEN, C. J. 1979. *Information Retrieval*. 2nd ed. Butterworth and Co., London.

VICKERY, A. AND BROOKS, H. M. 1987. PLEXUS—The expert system for referral. *Inf. Process. Manage. 23,* 2, 99–117.

YOKOKOTA, M., TANIGUCHI, R., AND KAWAGUCHI, E. 1984. Language-picture question-answering through common semantic representation and its application to the world of weather report. In *Natural Language Communication with Pictorial Information Systems,* L. Bolc, Ed. Springer-Verlag, Berlin, 203–253.

Received February 1993; revised April 1995; accepted April 1995