NPSCS-91-012

NAVAL POSTGRADUATE SCHOOL
Monterey, California

EXPLOITING CAPTIONS FOR ACCESS TO MULTIMEDIA DATABASES

Neil C. Rowe
Eugene J. Guglielmo

April 1991

Prepared for:

Naval Postgraduate School
Monterey, California 93943

NAVAL POSTGRADUATE SCHOOL

Monterey, California

Rear Admiral R. W. West, Jr.
Superintendent

Harrison Shull
Provost

This report was prepared in conjunction with research funded by the Naval Postgraduate School under Direct Funding.

This report was prepared by:

UNCLASSIFIED

ECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
4. PERFORMING ORGANIZATION REPORT NUMBER(S) NPSCS-91-012	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Computer Science Dept. Naval Postgraduate School 6b. OFFICE SYMBOL (if applicable) CS	7a. NAME OF MONITORING ORGANIZATION Naval Ocean Systems Center
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943	7b. ADDRESS (City, State, and ZIP Code) San Diego, CA 92152
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Naval Postgraduate School 8b. OFFICE SYMBOL (if applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER O&MN, Direct Funding
8c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
EXPLOITING CAPTIONS FOR ACCESS TO MULTIMEDIA DATABASES

12. PERSONAL AUTHOR(S)
Neil C. Rowe and Eugene J. Guglielmo

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Progress	FROM 3/90 TO 3/91	April 1991	33

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Databases, Natural Language, Multimedia, Hashing

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Descriptive captions help organize noncompetitive media. But automated use of captions in retrieval from computerized multimedia databases has not been much examined because it would seem to require significant natural language processing. We argue that captions can be naturally expressed in a restricted language whose interpretations is easier than general natural-language understanding. We describe a multimedia database system that stores interpreted captions in predicate calculus for each media datum; it then interprets restricted-language queries, and finds matching media objects.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Neil C. Rowe	22b. TELEPHONE (Include Area Code) (408) 646-2462 / 22c. OFFICE SYMBOL CS

Exploiting Captions for Access to Multimedia Databases

Neil C. Rowe and Eugene J. Guglielmo[1]

Department of Computer Science
Code CS/Rp, U. S. Naval Postgraduate School
Monterey, CA USA 93943
(rowe@cs.nps.navy.mil, guglielm@cs.nps.navy.mil)

ABSTRACT

Descriptive captions help organize noncomputerized media. But automated use of captions in retrieval from computerized multimedia databases has not been much examined because it would seem to require significant natural language processing. We argue that captions can be naturally expressed in a restricted language whose interpretation is easier than general natural-language understanding. We describe a multimedia database system that stores interpreted captions in predicate calculus for each media datum; it then interprets restricted-language queries, and finds matching media objects. In exploring these ideas for two database applications, we have recognized three important issues. (1) Using a caption does not require deep understanding of it, just a comprehensive type hierarchy for concept types in it. (2) Captions can be accessed faster than media data because they are much smaller. So to access media data, we should map first to captions through a hash table. We argue that only nouns and verbs should be hashed, and that additional pointers should link subtypes to types. A "coarse-grain" search can intersect hash-table lists to find a candidate set of captions for a query; a "fine-grain" search can then carefully attempt matching the query to each, with variable binding, etc. (3) "Supercaptions", describing sets of other captions, can minimize caption redundancy; supercaptions can be of other supercaptions, etc. Pointers to supercaptions simplify the hash table. But now there is a conflict between exploring subcaptions and exploring supertypes of an entry in the hash table; we propose concurrent processing to solve this.

[1] This work was sponsored by the Naval Ocean Systems Center in San Diego, California, the Naval Weapons Center in China Lake, California, and the U. S. Naval Postgraduate School under funds provided by the Chief for Naval Operations.

Keywords: information retrieval, multimedia, captions, databases, natural language, parsing

1. Introduction

Descriptive captions have long been valuable in organizing and retrieving from multimedia data. Examples include English descriptions below newspaper photos, titles on slides, record jackets, and labels on videos. Captions can focus on only the important things in a media datum, but unlike a keyword list, a caption can have a complex structure mirroring the structure of a media datum. Although some multimedia database systems store captions, rarely have captions been used to aid retrieval since that seems to require a big dictionary and complex natural-language understanding routines. Nonetheless, we believe that software technology has now made sufficient progress to use captions routinely to help retrieval from many multimedia databases.

For example, the Naval Weapons Center at China Lake, California keeps a library of 36,000 photographs. Since many of the photographs appear very similar, each photograph has an associated caption that is stored in a computer database, such as this for Figure 1:

> **Air to air, TP87A209 Sidewinder AIM 9M test with F-15 aircraft USAF 82028 and F-16 aircraft ASAF 83131 of 422nd Test and Evaluation Squadron. Full side views of both aircraft and individually uploaded with missiles. Excellent. LHL 253149, 51 and 52 released S. B. Oster Pao, 5/29/89.**

Notice that this language is considerably more formal than everyday English, and thus is not as difficult to parse and interpret. However, the Naval Weapons Center currently uses such caption text only as a source of manually-selected keywords which are matched to keywords supplied by a user. Clearly, much valuable information in captions is being ignored, like the information in the above paragraph about the relationship of the aircraft to the squadron, the missiles to the aircraft, and the person to the rest of the caption. So keyword-based retrieval specifying "Sidewinder" and "aircraft" could also mistakenly find pictures of damage done to aircraft by the Sidewinder. Furthermore, some reasoning about a caption is necessary to match it, since many things are implied but not stated directly, like that a Sidewinder is a missile, "AIM 9M" is the version code of a missile, and that "excellent" refers to a clarity scale for photographs. So keyword-based retrieval specifying "Sidewinder", "test", and "excellent" to find excellent test results could mistak-

enly retrieve Fig. 1.

We propose that arbitrarily-long unformatted captions, in part from natural language, be created for every media datum in a multimedia database. Captions can also be created for sets of media data (like on all pictures taken during a particular missile test), and inherit to subcaptions. Then queries to the media data could first check all this caption information. This could save query processing time, because a caption can be stored in much less space than most media data, and a faster storage could be used than that for the media data; then failure to match a query to the caption eliminates the need to retrieve the media datum from slow storage. And users could browse the caption data to decide what they are interested in before costly media-data retrievals (keyword lists are not as helpful to browsers). To simplify matching, both captions and queries can be parsed and interpreted, then represented as "meaning lists" of semantic properties and relationships; this can be done long in advance for the captions.

Natural language (e.g. English) processing by computer has made slow but steady progress in recent years, and it is becoming increasingly efficient while at the same time allowing considerable subtlety of expression. Using natural language solves many of the ambiguity problems in the relationship of words in keyword lists, improving the precision of query matches. Thus we are including a natural-language processing component in our caption-oriented multimedia database system. Understanding natural-language descriptions of the contents of multimedia databases is usually a considerably simpler problem than that of general natural-language understanding, since the universe of discourse is usually quite constrained. Nouns tend to be concrete since they usually correspond to observables in the media data, and quantifiers and other logical operators are rare since often the easiest way to describe a media datum is to describe its separate pieces separately. And most multimedia databases emphasize still photographs and other fixed-time graphics to which few verbs can be applied, and verbs are one of the hardest aspects of natural language processing. But most importantly, we use natural language only to access entities in a database, and complete understanding of the words is not necessary for this goal. For instance, for the query "Air to air missiles mounted on aircraft", it is unnecessary to know exactly what "missile" and "mounted" mean to match the query to the above example caption, just that a Sidewinder is a missile and uploading is a kind of mounting.

Besides indexing by keywords, an alternative to caption matching is content analysis of media data at query time, but this usually requires too much computational effort. There are some exceptions, such as scanning a short block of text to find a particular name. But such purely syntactic analysis is inflexible and its utility is limited for pictures, video, and audio for which inferencing is often needed. For instance, we would have a hard time finding Fig. 1 for a query asking for pictures of Sidewinder missiles: The missiles are small in the photograph and easily confusable with the gas tanks hanging from the bottoms of the planes, and they cannot easily be found in the picture until the plane outlines are found first by processing of the entire picture. (A reasonable digital representation of this picture would be 500 by 500 bytes.) And additional information must always supplement content analysis: The name of the squadron to which the planes in Fig. 1 belong is not indicated in the photograph, nor the identity of what is being tested.

2. Previous work

Many researchers have worked on the problem of accessing multimedia data efficiently, although we know of no one who has tried to use captions in the central way that we do, nor anyone who has exploited captions on sets of captions. There is a variety of related research in information retrieval, database design, and artificial intelligence, for which we can cite some representative papers. Some researchers in information retrieval have investigated "semantic" representations of retrieval objects instead of the standard keyword lists. Kolodner's[1] pioneering work embedded facts for retrieval in a complicated semantic network, and used a variety of special heuristics suggested by human reasoning to intelligently search that network; the primary concern was computer-generated explanations of text data, a more difficult problem than ours. Cohen and Kjeldsen[2] proposed spreading activation over a semantic network to find qualitatively good associative matches. Rau[3] in SCISOR proposed a two-stage retrieval process from a semantic network in which the first stage was a spreading activation and the second was matching between a subgraph and a graph; input was English questions, so a significant portion of the implementation was devoted to natural-language processing and explanation of text data. Smith et al[4] in EP-X handled term-name differences between query and datum by using a hierarchy of concepts, where all levels could have pointers to retrieval objects.

Researchers in database design have been increasingly interested in multimedia databases. Some of this research concerns good ways of describing multimedia data for efficient retrieval, as the special summary data to describe pictures in Chang et al [5] and the special parameters for describing video in Nagel [6]. Such descriptive information should be part of a good caption on the media datum. Other research concerns efficient administration of a database system containing multimedia objects, which can often be difficult because of its highly varied and highly storage-intensive formats. Bertino et al [7] and Roussopolous et al [8] exemplify this work, with an emphasis on conceptual modeling and query languages.

A longtime concern of artificial intelligence has been manipulating descriptions of the world, and many of its results apply to our problem. A variety of books address practical issues in knowledge representation for artificial intelligence, as Rowe [9]. Grosz et al [10] exemplifies the current state of natural-language processing tools, in presenting a powerful design tool for creating natural-language parsers and interpreters for a wide variety of domains. Wilensky [11] provides an example of a powerful natural-language system that can be used to answer a wide range of English questions about the UNIX operating system; its success suggests that natural-language processing can be feasible and efficient for a surprisingly broad domain.

3. Overview of our caption-based access to multimedia data

Fig. 2 shows a block diagram of the data structures in our caption-based approach to efficient access of multimedia data, and Fig. 3 describes the blocks. Humans interact with our system at two places, the top left and the top right corners; on the left, human experts supply media data and their associated captions for storage in a multimedia database, and on the right, non-expert humans query the data. The actual media data (which comprise the *multimedia database*) are stored in a separate system on a separate processor, since media data generally require far more space than the access data structures discussed here. We expect pictures will usually be the most common form of media data, and each picture will be at least the complexity of a television picture (500 by 500 bytes), and we have a target of one million media data items in this design, so the multimedia database should be about 10^{11} bytes. This number and the generally read-only nature of the media data strongly suggest optical storage, which is slow for random access. Furthermore, multimedia data can come in many different formats, suggesting an object-oriented database

system. Previous work by B. Holtkamp, V. Lum, and the first author [12] proposed a details of it, and work is continuing on its implementation (although that work also proposed including captions and registration information in that database, we now think it a poor idea). Since the multimedia database will function mostly independently of the caption-based retrieval, we will not discuss its details further in this paper.

The main innovation of our design is the access to media data through semantically richer information--*meaning lists*, parsed and interpreted captions--instead of keywords. Meaning lists are lists of predicate-calculus expressions giving the "meaning" of captions, and are equivalent to semantic networks; Fig. 4 gives an example. Usually they can be written as lists of literals because logical conjunction is usually the primary operator necessary: A caption usually specifies the meaning of each part of a natural-language utterance, then requires that the "and" of all these meaning components must hold. Variables in the arguments to the literals can relate the parts of a caption description; in Fig. 4, the variables are the codes consisting of a letter followed by a number. Methods for obtaining meaning lists are described in section 4.1.

Besides the captions themselves, our system requires auxiliary information from a lexicon, a concept hierarchy for the domain, and frame recognition rules. The *lexicon* (or dictionary) is necessary for parsing, and gives for each possible natural-language word its "meaning": its part of speech, its grammatical forms, and the form of the literals needed to represent it. Ten thousand words exclusive of proper nouns is a reasonable lexicon size for most applications. Many of the hardest words to represent in a lexicon--for instance, conjunctions and quantifying adjectives--are consistent in meaning across a wide range of domains, so we can just borrow their interpretation from existing natural-language systems; the words that significantly change between applications are the nouns and a few verbs, and their representation is more straightforward. The *concept hierarchy* is a type hierarchy on the key concepts that can be included in meaning lists. It has both upward pointers (for semantic checking after parsing of natural language) and downward pointers (for finding captions with terms that are subtypes of those in the query); there can be more than one upward pointer from a concept. Lastly, the *frame-recognition rules* add generalization terms to meaning lists that reflect inferences beyond what the natural language actually said, like the implied firing of the missiles in the first caption of section 1.

The meaning lists for queries are used to find relevant media data by two phases, a coarse-grain search and a fine-grain search. The coarse-grain search does hash-table lookup of all occurrences of certain helpfully restrictive terms in the literals, those corresponding to nouns and verbs in the original natural-language input. This gives a set of *caption pointers* to all caption objects containing these identifying literals, and thus candidates for satisfying the query. Then a fine-grain search matches the full query meaning list against the candidate captions' meaning lists, binding variables as necessary.

A million media data items means a million captions. We expect an average caption will take 100 bytes; captions should summarize, not exhaustively catalog. So the caption database will be about 100 megabytes uncompressed, and compression techniques can make it significantly smaller. Note in Fig. 2 that some of the caption database is allocated to *supercaptions*. These are captions that describe a class of media data, eliminating redundancy; Fig. 4 shows some example supercaption information. Supercaptions are an important part of our design, and are a more user-friendly way of modeling hierarchical structure in data than an index on keywords; section 4.3 will discuss them further.

We have applied our design to two important applications. We first built a prototype of some portions of this design for the domain of the military history of U.S. forces in the Pacific in World War II. We used media data of pictures digitized from published books about World War II, about 100 photos in all, plus some aerial photographs of an army base. We used the captions printed with those photos in the books, and some captions of our own for the aerial photographs, and we wrote a reasonably general augmented-transition network parsing and interpretation routine in Prolog; in the latest version the parsing times, on test sentences averaging fifteen words, are all less than ten seconds with uncompiled Prolog. The lexicon was 575 words, of which 227 were nouns and 46 were proper nouns. The meaning lists were then converted by code in C to an SQL-like language that accessed an INGRES database. The hardware was a Sun workstation. With the success of the prototype, we are now working with a significant existing database of records, both historical and current, of projects at the Naval Weapons Center at China Lake, California. Currently the database contains online captions of 36,000 photographs themselves stored offline; we are putting the photographs and other media data online in an optical jukebox. To demonstrate the generality

of our methods, we will also be including in our database the text of project reports, engineering drawings, viewgraphs for project presentations, video of project tests, and audio of test pilot dialogues. Our intention is to provide a multimedia database for proposal writing, public relations, and library purposes for the various development projects at China Lake, but the methods employed should apply to any research organization. The processing hardware will include a network of Sparc workstations. We intend to continue to use Prolog for some parts of the design, but the natural-language processing will be done by purchased software (see section 4.1).

4. Extraction of meaning lists

For efficient retrieval it is important that we store meaning-list representation of a caption and not the caption itself: natural language processing of captions at query time would enormously increase processing time. Following previous software development [9], we use meaning lists in Prolog linked-list format, lists of literals where most literals express properties or binary relationships. To simplify matching, we are trying to limit the properties and relationships to a small set of primitive properties and relationships; for instance, we will not distinguish between the relationships asserted by the terms "within", "inside", "part of", "containing", and "comprising". Again, to do efficient retrieval, it is not necessary that the meaning lists capture the full meaning and implications of an English expression, just that they express enough of the main intent to find obvious matchings.

4.1 Ways of obtaining meaning lists

We are exploring three ways of obtaining meaning lists for captions and queries about captions, each useful for certain kinds of information. One is a structured menu approach where we ask the user a series of questions derived from a decision tree. For instance for the picture described by the caption in Fig. 4, we could ask the user to look at the picture and give the main action; then who is doing the main action; then if there is any action object; then whether there are any modifiers that can describe the action (like adverbs); then if any adjective modifiers can describe the subject noun; and so on. With this approach, parsing is simple. To save time, the user can be asked to confirm default values (some words strongly imply others).

A second way of obtaining meaning-list information is by content analysis of the media data. Although we dismissed this for use at query time in section 1, it could be used in setting up the caption database if the analysis were not complex. For instance, we could compute the predominant color in a picture or the grain size of the predominant texture. But it is much easier to be told that a picture represents an F-16 aircraft than trying to trace the plane's outline and then identify it.

Third, we can actually parse the restricted English of a sentence representing a caption or query, and this is the friendliest approach for a user. Some powerful natural-language understanding software is appearing. After a survey of what was available, we have begun using DBG from Language Systems Inc. (Woodland Hills, California). We found its speed was reasonable on test sentences. Its lexicon must be supplied in part by us; some of this information is the type information we will discuss in section 4.2, and other is standard morphology (suffixes and prefixes of words). Generally speaking, the most difficult words in English are multi-domain multi-use words like conjunctions and prepositions, but their meanings do not vary much between domains and their lexicon entries can be copied from existing lexicons. Additional lexicon information can be obtained by structured menus addressed to the designer, as in the TEAM Project [10].

We will allow only descriptive captions, as opposed to background. For instance:

> **U. S. soldiers wading ashore in columns churn up the waters off Morotai Island, midway between western New Guinea and the Philippines. MacArthur wanted Morotai so Allied aircraft could operate from there and protect his Philippine landings. The Morotai invaders met no resistance.** (from R. Steinberg, *World War II: Island Fighting*, Time-Life Books, 1978)

Only the first half of the first sentence actually describes the photograph. This is a common convention; for instance, in randomly selected articles of *National Geographic* we found in 110 out of 120 caption paragraphs that the first sentence was the only descriptive one. We will also exclude captions whose associated pictures merely invoke a theme, as a caption about the Navy's budget for a picture of an aircraft carrier.

On the other hand, our Naval Weapons Center database has many multi-sentence captions in which all sentences are descriptive, like the example of section 1. Frequently these captions exemplify a kind of multi-

sentence grammar where the sentences occur in a particular order. For instance, this caption is typical of many in the database:

Skipper missile validation of A-6E aircraft loading check list. Closeup views of missile and MK 7 loader, and wire/electrical connections. LHL 226648 released D. Kline, 12/13/85.

First a testing action and its subject are described; then in a separate sentence, the focus and nature of the photograph; then in a separate sentence, the authorization for release of the photograph. The example of section 1 follows the same scheme. Thus a simple discourse grammar can parse many of these captions to make interpretation even easier.

4.2 Conceptual generalizations: type hierarchies and frames for stereotypical actions

To permit captions to be short, conceptual generalization on the contents of meaning lists must be possible. Conceptual generalization can exploit three kinds of information: a concept hierarchy, frames for domain stereotypes, and supercaptions. First, a complete and thorough type hierarchy for the concepts (nouns and verbs) in the domain of discourse must be created. For instance for military history, part would give geographical areas and locations, part would give the kinds of military ships, and part would give the different kinds of maneuvers a military ship can engage in. Fig. 5 gives the top of the hierarchy for the military history domain. Specifically in the Fig. 4 example, "U.S." is a country, "columns" is a kind of military formation, "Morotai Island" is a place in the western Pacific, "churn" is a side effect of physical motion in liquids and semi-liquid materials, and "wading" is a locomotion used by humans in crossing water of only a narrow range of depth. Such information can be obtained from domain experts using techniques of knowledge acquisition for expert systems. Obtaining all such information may seem considerable work for the designers of multimedia database system. But most of it can come from a natural-language dictionary, and it is necessary anyway for a good hierarchical indexing scheme on keywords, without which user-friendly access through keywords is impossible.

The second kind of generalization information we need is the "frame" or "script" abstraction that frequently occurs in describing often-stereotypical human activities. The terms "ashore," "wading," and "waters" in the caption of Fig. 4 together suggest that there is a beach-landing operation going on, a stereotypical kind

of military operation. Certainly, we can create a hierarchy of military operations that includes a beach landing. But we would not be able to recognize from the concept hierarchy alone that a beach landing is referenced in this sentence, since no single word indicates it, only the combination of clues. This kind of recognition is a "frame" or "script" problem and needs techniques like those in Schank and Abelson [13]. The abstractions and their clues must be obtained from an expert in the domain. We expect the number of different such abstractions to be small. For instance, military activities necessary to explain a World War II data base exemplify about ten concepts (see Fig. 7); each has stereotypical ways of accomplishing them with particular props, and each has associated preconditions and effects. So when we recognize these stereotypical concepts in meaning lists, we should insert extra summary terms into the lists, as additional terms to exploit in matching captions to queries.

4.3 Conceptual generalizations: supercaptions

Our third kind of conceptual generalization seems to be an idea unique with us: the *supercaption*, a caption that describes more than one media datum. For instance (see Fig. 6), the Morotai Island caption in Fig. 4 could be a subcaption for the supercaption "Black/white photographic record of U.S. in World War II in the Pacific", which in turn could be a subcaption of the supercaption "Historic black/white photographs of combat". Supercaptions can be obtained from a domain expert just like captions, and are most useful when they give complex meaning-list information unobtainable from the concept hierarchy, like the dates, times, and places common to a set of photos of a battle. Supercaptions can create a hierarchical structure different from the type hierarchy of domain concepts, as in Fig. 6. Supercaptions can represent how an expert clusters media data, unlike groupings based on single data features.

"Stub" or "registration" information, about how a set of media objects were created, is naturally expressed with supercaptions. For instance for a photograph or video, this includes the photographer, the type of film, the exposure, the date and time the picture was taken, the place where the picture was taken, and so on. These properties usually apply to classes of pictures, and would require unfairly tedious labor to enter separately for every picture.

Parsing and interpretation of supercaptions involves some issues not addressed with captions. One is universal quantification: can the supercaption information be appended to each of its subcaptions? For instance, does each picture in the series of pictures entitled "Morotai Island actions by U.S." show an event on Morotai, or do some pictures show background, preparations, or aftermath? The alternative is to treat the supercaption as a "theme" for conceptual clustering. Another important question is whether the multimedia data referred to by the supercaption represents an complete enumeration; if so, we can make several powerful inferences. For instance, the supercaption "The American naval ship types of World War II" implies that every ship type is shown in at least one media datum, and furthermore every media datum contains at least one ship type.

Linguists have not devoted attention to this specialized issue, so we have developed our own heuristics for their semantics. The key in most single-sentence captions seems to be the nature of the grammatically central noun in the caption, and usually that is the noun in the subject noun phrase; for instance, "types" in "The American naval ship types of World War II". Let the variable corresponding to the grammatically central noun be x, and a predicate asserting the truth of the conjunction of all the meaning-list literals linked to it be $p(x)$. Then:

--Rule 1: If $p(x)$ is a plural noun equivalent directly depictable in the media data referred to by the supercaption, or represents a supertype of something depictable, then $\forall c \in subcaptions(\exists x\ p(x,c))$. This follows from the idea that each subcaption must advance the "argument" of the supercaption, and if a subcaption referred to a picture that did not contain the main noun of the supercaption, it would in some sense be inadequate in supporting the claim of the supercaption. For instance, for "The American naval ship types of World War II."

--Rule 2: If the main type of the caption is a single event (events are not "directly depictable"), then interpret all events in the subcaption as parts of the larger event in the supercaption. That is, $p(s)/\backslash\forall c \in subcaptions(\exists e \in events(c)[part\_of(e,s)])$. For instance, if the supercaption is "Morotai Island actions by U.S." then all verb forms in subcaptions denote actions that are part of Morotai Island actions by the U.S.

--Rule 3: If the main type of the supercaption is accompanied by the determiner "the" or is itself a non-picturable type referring to an aggregate (as denoted by the English words "catalog," "gallery," "display," "index," etc.), followed by the word "of" and a prepositional phrase, then completeness of the subcaptions in representing the supercaption can be assumed. That means, taking the noun type in the "of" prepositional phrase as $p(x)$, that $\forall x\,(p(x) \rightarrow \forall c \in subcaptions(\exists z\,[in(z,c) \wedge p(z)]))$. For instance, "The American naval ships of World War II" implies that there exists a caption pointed to by the supercaption that contains every possible type.

--Rule 4: If none of the preceding rules apply, the supercaption must be interpreted as a theme invoked only for indexing of supercaptions, and it has no implications for its subcaptions.

All other terms in meaning list that are linked to the main variable follow similar quantification to that in the above rules.

5. Retrieval using captions

Given a query on our multimedia database, we will translate it into a meaning list. Exploiting the captions for retrieval means first finding captions whose meaning lists match key terms of the query meaning list (*coarse-grain search*): then for each that matches the whole caption, we retrieve the corresponding media object (*fine-grain search*). This two-stage search postpones the handling of the usually-bulky media data. To further simplify matters, we assume the query contains no quantifiers.

There are many ways to use semantic information such as captions for retrieval, not all efficient. The approach of Kolodner [1] used special-purpose heuristics good for modeling everyday human reasoning but not necessarily good for technical domains. The approaches of Cohen and Kjeldsen [2] and Smith et al [4] explored a semantic network, but only a uniformly structured one (by topic associations in the first, and a type hierarchy in the second); thus they cannot exploit the full range of knowledge that we do with our three kinds of conceptual generalization. So Rau's SCISOR [3] is the closest to what we want to do, with its emphasis on a variety of knowledge for different purposes; it used a two-phase search process like ours.

5.1 Fine-grain search

Our fine-grain search is by definition matching done with the full captions. This inevitably requires a subgraph-matching algorithm, that tries to match pieces of a caption by binding variables and backtracking as necessary. Subgraph matching is much addressed in computer science, and there are many algorithms for the many special cases of it. In all algorithms, combinations must be tried until a match is found. In the worst case, the general subgraph-matching problem is exponential in complexity since the general algorithms are NP-hard. The worst case will not often be approached in real databases with real user queries, as it requires a few predicate names to be used repeatedly in meaning lists, which is unlikely considering the human origins of captions and queries.

5.2 Coarse-grain search

The coarse-grain search must map from key terms of the meaning lists to caption pointers. Since we have one million captions, we will need $\log_2 10^6 = 20$ bits for each caption pointer; since we will have about 50 indexable items per caption based on our examination of good human captions, we will need about 125 megabytes for hash-table pointers alone to the captions. This suggests the pointers be in secondary storage. Since we expect to use widely scattered portions of the caption access data at any one time, a hashing scheme is better than an index.

So we identify key terms in the meaning list translation of a user query, hash these to a secondary-storage hash table of caption pointers, intersect the pointer lists (we assume by default that a user wants captions exactly matching the whole query), and look up the corresponding captions. But what are the "key" terms? After analysis of sample captions, we concluded that only the equivalents of nouns and verbs as they appear in meaning lists provide sufficiently restrictive information on the set of target data to make them worthwhile to exploit in a coarse-grain search. Conjunctions, auxiliaries, expletives, and pronouns do not translate directly into meaning lists. Prepositions and adverbs usually provide only weak restrictive information and can be fuzzy (for instance, when is one object north of another in a picture?) Some adjectives like "U.S." are helpful, but usually only non-abstract adjectives; the "alert" in "alert soldiers" contributes

far less. Verbs can be useful, but probably less so than nouns because they are hard to depict in media data.

Our hash table gives only exact matches for a query term, since the hash table is necessarily large. For instance, if a caption mentions Morotai Island, then only the hash table entry for "Morotai Island" points to it, not the entry for "Western Pacific" or "Battle sites of World War II". So a query that does mention "Western Pacific" must use the concept hierarchy to reach other hash-table entries to find the Morotai Island caption. This will save much space at the expense of time to follow the downward pointers of the concept hierarchy (but significant clustering of these references on hash-table pages can probably be done). We can also save space by using supercaption pointers in the hash table as well as caption pointers. A supercaption pointer can represent many subcaption pointers, and the linkage can be specified in another table.

Although our prototype implementation was for a single processor, the coarse-grain search can use concurrent processing for the conjunctive portions of queries, where each processor writes to a shared memory of candidate caption pointers. We intend to do this on a network of Sparc workstations. Initially, each key term in the query meaning list can be assigned a separate processor with its own list of caption pointers it has found so far in its designated area of the shared memory. Each processor can use the concept hierarchy to find subtypes of its term in the concept hierarchy, and the supercaption-subcaption table to find subcaptions. Whenever a processor exhausts all possibilities for its pointers, it goes through the pointer lists generated by the other processors and (1) eliminates all those that do not appear in its own list, and (2) eliminates from its own list all pointers not appearing in lists of other exhausted processors. The first processor to finish will tend to be the one finding the fewest pointers and hence having the most restrictive terms, and this processor will eliminate possibilities first, the most efficient way of doing a set intersection. Note that this approach permits the first few media datums found to be supplied to the user while processing continues to find others: this can keep the user happy during a long search.

Fig. 8 shows an example of concurrent coarse-grain search. An English question is parsed and interpreted to create a meaning list. At the same time frame recognition rules infer that a beach-landing frame is

applicable, and add a term for it to the meaning list. Three key terms in the meaning list are each assigned a separate process to look up caption pointers: "photos", "landings", and "Philippines campaign". We also establish a processes for the beach-operation frame. Now "photos" has subtypes of military and civilian photos in the concept hierarchy. So we can establish separate processes for these to find captions that reference them explicitly; and for all subtypes of these subtypes; and so on. "Philippines campaign" is a term likely to be in a supercaption; so one pointer for it in the hash table could be to a supercaption in the caption database (as we could quickly identify if supercaptions had a designated range of pointer codes). Then we could establish separate processes to find the subcaptions of the supercaption, while still trying to find direct pointers to "Philippine campaign." Here the subcaptions would be for the various battles involved in that campaign; we could explore them and their subevents, returning all caption pointers encountered as we find them.

5.3 Further details of the coarse-grain search

The only detail in Fig. 8 as yet unexplained is the relation to beach operations to amphibious actions. This is an example of our *alias* handling, cross-referencing from one equivalent term to another. Aliases are common in natural language and are important to the user-friendliness of text-based interfaces. For instance, "plane", "airplane", and "aircraft" mean the same thing. Most of this can be handled in the lexicon by assigning the same literals to represent the meaning of the aliases. But when aliases are near but not exact, like "beach operation" and "amphibious action", it makes more sense to postpone their handling to the coarse-grain search when they can serve as search heuristics. We can designate one alias as primary, and store pointers with it: all other aliases can just have a special flag and a pointer to the primary alias.

We expect that negated terms will be rare in captions, since the point of a caption is to describe presences, not absences. But negatives can occur in queries, as for instance "Non-U.S. soldiers in the Philippines campaign." We can retrieve the pointers of the negated term with a separate processor just as before, but now we eliminate pointers in other processors' lists that *do* occur in the pointer list for this negation processor. Also, non-negation processors should delete pointers from their own lists that occur in the list of any negation processor, even if the negation processors are not done.

If we write the query as a conjunction of disjunctive expressions, the algorithm of the last section can be applied separately to each item in the conjunction. Then the disjunctions can be treated just like the subtypes and subcaptions, which are implicit disjunctions. Disjunctions in captions should be rejected as too vague to be a good description. Again, we assume no quantifiers in queries.

The concept hierarchy of section 4.2 is an "a_kind_of" or "generalization/specialization" hierarchy, and the coarse-grain search algorithm exploits the predominantly downward nature of inheritance with respect to these links. However, several other kinds of inheritance can also occur, as discussed in Rowe [9], and can be exploited by a smarter algorithm. One classic example is with the "part_of" or "containment" relationship between concepts. For instance, if query asks for pictures of planes with ceramic-composite wings, that should match a caption describing a ceramic-composite plane, since a wing is part of a plane. This kind of inference won't work at all for certain properties (like cost) and works in the opposite direction for other properties (like defectiveness of a part, which inherits upwards to give defectiveness of a plane containing the part). A rule-based inference system is necessary to specify all the cases.

5.4 Time efficiency of our approach

To show that our media data search is efficient in its use of time, we must compare it to other methods of information retrieval. To be fair, we cannot compare it to the methods that store media data in main memory or secondary storage [1,2,3], since the total amount of media data we want to store is too large; the "spreading activation" idea used in that work could require enormous numbers of optical-jukebox disk fetches, since significant clustering of access pointers is hard to achieve. So the best comparison is to EP-X [4] with its media-data pointers embedded in a type hierarchy of keywords. For EP-X, fine-grain search must be done by the user, so unnecessary extra media data is retrieved compared to our approach. On the other hand, our approach requires that all queries go through a new secondary-storage structure, the captions database. Since we are talking about slow secondary and tertiary storage, and algorithms requiring little main-memory processing (except perhaps for parsing, which preliminary experiments convince us can be done in at worst a few seconds), page access time will greatly override all other time costs. Let c_S be the cost of secondary storage page fetches for the captions, c_T the cost of media data page fetches, n the

number of media datum pointers produced by EP-X or the number of caption pointers we produce, and p the probability that a medium datum pointer in EP-X or a caption pointer in our system will satisfy the fine-grain search criteria. Assume all other secondary storage page fetches are negligible in cost (concept-hierarchy and supercaption-hierarchy pointers will show a high degree of clustering, and their page fetches can be done concurrently with the caption and media-data fetches). Then our approach will be better than EP-X if $nc_T > nc_S + npc_T$. or when $p < 1-(c_S/c_T)$. (Actually, we are being conservative in assuming that the same number of caption pages and media object pages will be needed; otherwise the n on the left side must be increased.) In our system currently under development, we estimate the paging cost ratio will be about 0.1 based on claimed times of the hardware we are using (18 msec. seek for magnetic disk, 90 msec. seek for optical disk, 10 seconds for exchanging disks in the jukebox) and the assumption that enough clustering of media data references on optical disks can be done so that exchanging disks is necessary only once in about 100 page fetches. So the fine-grain search need only exclude one caption in ten in order that our caption-based approach be faster; thus fine-grain search does not have to rule out much in order that our approach be better. At the same time, our approach will be more user-friendly since the user can work in natural language.

5.5 Partial matching to a query

A common user error is putting so many restrictions in a query that its answer set is empty. With our caption-based approach, this circumstance can be identified without going to the multimedia database, at worst in the fine-grain search, or at best in the coarse-grain search without going to the caption database either. When this happens, it is helpful for the system to automatically try *partial matching*, finding captions that satisfy some generalization of the query. Three modifications of our processing algorithm make this not difficult to do. First, we can find pointers that occur in all but at most K of the pointer lists intersected, the lists corresponding to the key query terms. Second, we can search upward in the concept hierarchy as well as downward: to supertypes of terms, or to supercaptions of captions. Third, we can follow less exact aliases of terms in the concept hierarchy.

All three ideas are quantifiable, and they trade off with one another, so an A* search is strongly suggested

to find the best "near miss" media data. Then the cost used in the A* search can be the sum of (1) the number of (previously intersected) pointer lists in which the term does not occur; (2) $\log_2$ of the ratio of the generalization set size to the starting set size (set sizes being determined by counting corresponding multimedia objects in advance); and (3) $-\log_2$ of the subjective probability that an item satisfying the alias term will satisfy a user requesting the original term. The weighting of these three cost factors will need to be determined by trial and error; analogous weighting problems arise frequently in information retrieval and many methods have been developed for them.

6. Customization for the user

There are many opportunities for optimization to the needs of a particular user in our system. Lexicon, caption-pointer, caption, and media-data pages can all be cached with a least-recently-used replacement policy. Hence, we should place the most closely related items together on pages wherever possible. It may also be good to cache results of caption-pointer intersections, which amounts to caching of structures rather than keywords, a high-level form of caching. User customization of the natural-language processing is not as important, but information as to particular word senses of ambiguous words that the user employs can be stored.

7. Conclusion

Captions are a natural way to organize multimedia data. But using captions in a significant way in an automated retrieval system is a difficult problem which requires conceptual innovations as well as the sort of significant effort we have described our project, which we believe is the first frontal assault on caption-based data retrieval. Much work remains to be done. We are confident now we have a design that can work.

References

[1] J. L. Kolodner, "Indexing and Retrieval Strategies for Natural Language Fact Retrieval," *ACM Transactions on Database Systems*, *8*, 3 (September), 1983, 434-464.

[2] P. R. Cohen and R. Kjeldsen, "Information Retrieval by Constrained Spreading Activation in Semantic Networks," *Information Processing and Management, 23*, 4, 1987, 255-268.

[3] L. F. Rau, "Knowledge Organization and Access in a Conceptual Information System," *Information Processing and Management, 23*, 4, 1987, 269-284.

[4] P. J. Smith, S. J. Shute, D. Galdes, and M. H. Chignell, "Knowledge-Based Search Tactics for an Intelligent Intermediary System," *ACM Transactions on Information Systems, 7*, 3 (July 1989), 246-270.

[5] S. K. Chang, C. W. Yan, D. C. Dimitroff, and T. Arndt, "An Intelligent Image Database System," *IEEE Transactions on Software Engineering, 14*, 5 (May), 1988, 681-688.

[6] H. Nagel, "From Image Sequences Towards Conceptual Descriptions," *Image and Vision Computing, 6*, 2 (May), 1988, 59-74.

[7] E. Bertino, F. Rabitti, and S. Gibbs, "Query Processing in a Multimedia Document Systems," *ACM Transactions on Office Information System, 6*, 1 (January 1988), 1-41.

[8] N. Roussopoulous, C. Faloutsos, and T. Sellis, "An Efficient Pictorial Database System for PSQL," *IEEE Transactions on Software Engineering, 14*, 5 (May), 1988, 639-650.

[9] N. C. Rowe, *Artificial Intelligence through Prolog*, Englewood Cliffs, N.J.: Prentice-Hall, 1988.

[10] B. J. Grosz, D. E. Appelt, P. A. Martin, and F. C. M. Pereira, "TEAM: An Experiment in the Design of Transportable Natural Language Interfaces," *Artificial Intelligence, 32* (1987), 173-243.

[11] R. Wilensky, Y. Rens N.,and D. Chin, "Talking to UNIX in English: An Overview of UC," *Communications of the ACM, 27*, 6, (June 1984), 574-593.

[12] B. Holtkamp, V. Y. Lum, and N. C. Rowe, "DEMOM--A Description Based Media Object Data Model", IEEE Computer Software and Applications Conference (COMPSAC), Chicago Ill., October 1990.

[13] R. Schank and R. Abelson, *Scripts, Plans, Goals, and Understanding*, Hillsdale, N.J.: Lawrence

Erlbaum, 1977.

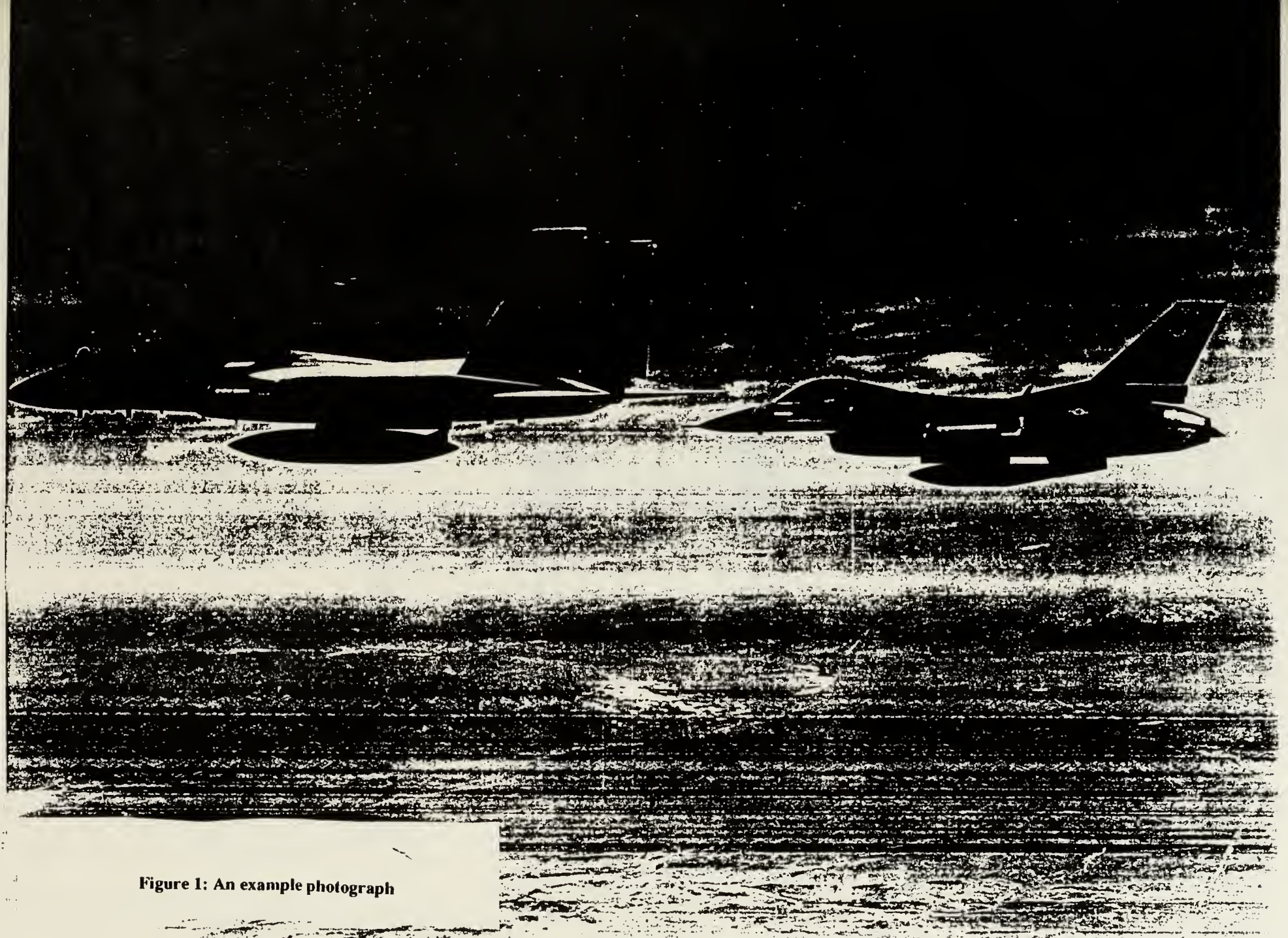

Figure 1: An example photograph

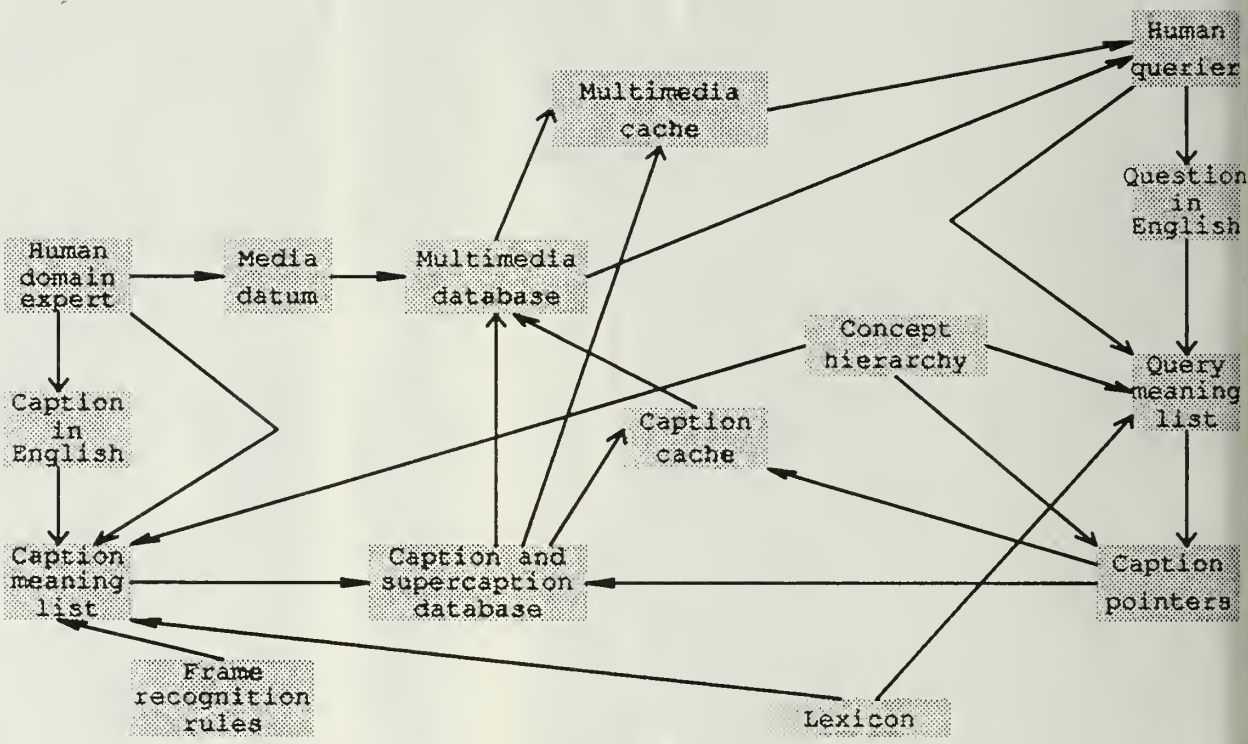

Figure 2: Block diagram of
our caption-based multimedia system

Data structure	*Description*	*Megabytes (approx.)*	*Storage type*
lexicon	dictionary for natural language processing	1	main memory
concept hierarchy	type hierarchy on domain concepts	0.1	main memory
frame-recognition rules	recognizes higher-level plans in the captions	0.1	main memory
hash table of caption and supercaption pointers	maps from interesting terms in meaning lists	100	secondary storage, magnetic disk
caption and supercaption database	meaning lists for captions and supercaptions	100	secondary storage, optical or magnetic
caption and supercaption cache	most-recently used captions or supercaptions	1	secondary storage, magnetic
multimedia database	the real data	>100,000	tertiary storage, optical, jukebox
multimedia cache	most-recently used objects	100	secondary storage, magnetic

Figure 3: Our data structures, with sizes for a million-object multimedia database with media datum items at least 10000K bytes each

Caption: **US soldiers wading ashore in columns churn up the waters off Morotai Island.**

Parse tree (a summarization of program procedure calls):

```
sentence
        (nounphrase
                (nounphrase
                        (nounphrase(adjectivelist(adjective("US")),noun("soldiers")),
                        participlephrase
                                (participle("wading"),adverb("ashore")) ),
                prepositionalphrase(preposition("in"),noun("columns")) ),
        verbphrase
                (verbgroup(verb("churn"),particle("up")),
                nounphrase
                        (nounphrase(determiner("the"),noun("waters")),
                                prepositionalphrase
                                (preposition("off"),
                                propernoun("Morotai Island") ) ) ) )
```

Meaning list (actual program output):

```
[plural(f2), soldier(f2), name(f2,U.S.), place(f2),
wade(f2), action(wade,g2), tense(g2,present), transitive(g2),
place(f2,0), inside(f2,i2), plural(i2), column(i2),
churn(f2,h2), action(churn,d2), plural(d2), tense(d2,present), direction(f2,0),
plural(h2), water(h2), definite(h2),
location(h2,l2), name(l2,Morotai Island), place(l2)]
```

Frame inferred: beach-landing

Example meaning terms inheritable from supercaptions:

```
[photograph(a3),focus(a3,medium-range),colorrange(a3,blackwhite),
war(a3,"World War II"),area(a3,"Pacific Ocean"),
campaign(a3,"Philippines recapture")]
```

Figure 4: An example parse tree and corresponding meaning list obtained by our parsing and interpretation program, plus examples of additional information inferrable or inheritable

1. physical objects
  1.1. geographical locations
    1.1.1. land units
    1.1.2. water
    1.1.3. air
    1.1.4. mixed units
  1.2. vehicles
    1.2.1. land
    1.2.2. water
    1.2.3. air
  1.3 weapons
  1.4. other military equipment
  1.5. people
    1.5.1. military
    1.5.2. civilian
  1.6. organizations
    1.6.1. military
    1.6.2. civilian
  1.7. terrain
  1.8. weather
2. abstract objects
  2.1. facts
    2.1.1. observations
    2.1.2. measurements
    2.1.3. thoughts
  2.2. events
  2.3. plans
  2.4. directions
  2.5. communications networks
  2.6. responsibility
  2.7. military actions
    2.7.1. aggression
    2.7.2. defense
    2.7.3. preparation

Figure 5: Top levels of the concept hierarchy for a multimedia database of World War II military history

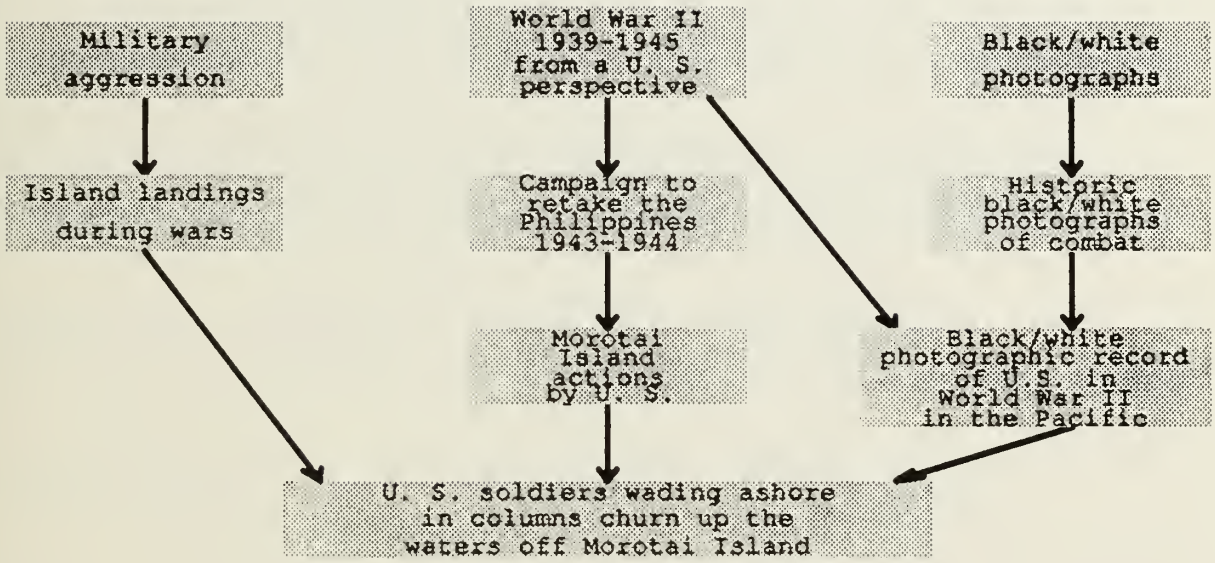

Figure 6: An example supercaption hierarchy

Plan: To find a sequence of steps that will achieve a goal
Order: To command someone to do something
Secure: To achieve a goal
Attack: Aggression from one entity on another
Defend: To act to minimize aggression by another
Attempt: To try to perform some action
Maneuver: To move in steer and an object through air, sea, or land
Neutralize: To make a strategic asset important
Disguise: To make a strategic object more difficult to recognize
Fortify: To make a strategic object more difficult to aggress upon

Figure 7: The ten basic military-history frames we use

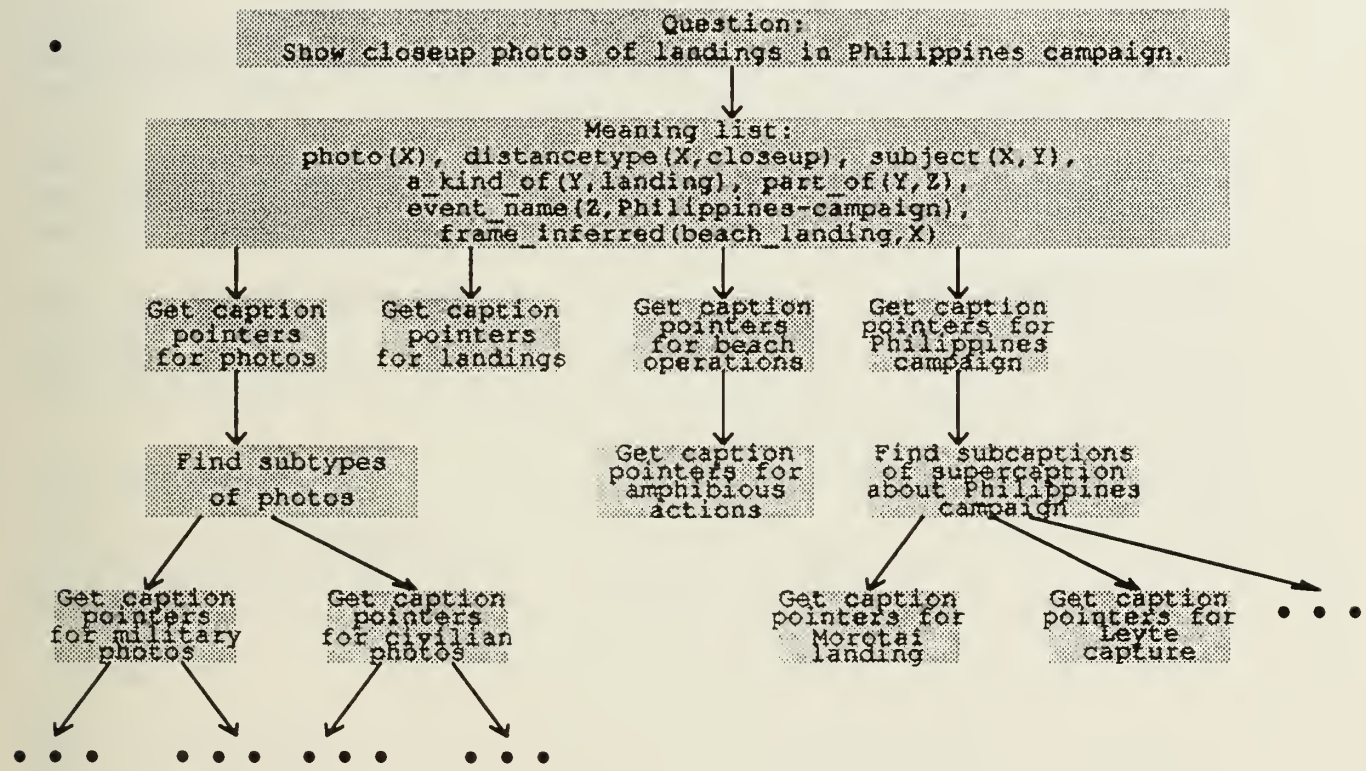

Figure 8: An example of processes created in a coarse-grain search with concurrency

Distribution List

SPAWAR-3242
Attn: Phil Andrews
Washington, DC 20363-5100 1

Defense Technical Information Center,
Cameron Station,
Alexandria, VA 22314 2

Dudley Knox Library, Code 0142,
Naval Postgraduate School,
Monterey, CA 93943 2

Center for Naval Analyses
4401 Ford Ave.
Alexandria, VA 22303-0268 1

Director of Research Administration,
Code 08,
Naval Postgraduate School,
Monterey, CA 93943 1

John Maynard
Code 402
Command and Control Departments
Naval Ocean Systems Center
San Diego, CA 92152 1

Dr. Sherman Gee
ONT-221
Chief of Naval Research
800 N. Quincy Street
Arlington, VA 2217-5000 1

Leah Wong
Code 443
Command and Control Departments
Naval Ocean Systems Center
San Diego, CA 92152 1

Bernhard Holtkamp
University of Dortmund
Dept. of Computer Science
Software-Technology
P.O. Box 500 500
D-4600 Dortmund 50
West Germany 5

Vincent Y. Lum
Code CSLu
Naval Postgraduate School
Monterey, CA 93943 5

Dr. Neil C. Rowe, Code CSRp
Computer Science Department
Monterey, CA 93943 20

Klaus Meyer-Wegener
University of Kaiserslautern
Computer Science Department
P.O. Box 30 49
D-6750 Kaiserslautern
West Germany 1

Professor Robert B. McGhee, Code CSMz
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93943 1