Retrieval Performance in FERRET
A Conceptual Information Retrieval System

Michael L. Mauldin

Center for Machine Translation
Carnegie Mellon University
Pittsburgh, PA 15213

ABSTRACT

FERRET is a full text, conceptual information retrieval system that uses a partial understanding of its texts to provide greater precision and recall performance than keyword search techniques. It uses a machine-readable dictionary to augment its lexical knowledge and a variant of genetic learning to extend its script database.

Comparison of FERRET's retrieval performance on a collection of 1065 astronomy texts using 22 sample user queries with a standard boolean keyword query system showed that *precision* increased from 35 to 48 percent, and *recall* more than doubled, from 19.4 to 52.4 percent.

This paper describes the FERRET system's architecture, parsing and matching abilities, and focuses on the use of the the Webster's Seventh dictionary to increase the system's lexical coverage.

1. INTRODUCTION

Most information retrieval systems in use today are word-based, but such systems are inherently limited. Don Swanson has characterized these limits in his "Postulates of Impotence" [16]

> P5: Machines cannot recognize meaning and so cannot duplicate what human judgement can in principle bring to the process of indexing and classifying documents.
>
> P6: Word-occurrence statistics can neither represent meaning nor substitute for it.
>
> P9: [Therefore,] consistently effective fully automatic indexing and retrieval is not possible.

He goes on to say he hopes his postulates will start arguments. Although we whole-heartedly agree with his postulate P6, it is controversial (for example, see [4, 7] for descriptions of the use of co-occurrence statistics to extract "latent semantic structure").

We do take issue with P5: the inability of machines to recognize meaning. The FERRET project at Carnegie Mellon is an attempt to bring the power of natural language processing to bear on the information retrieval problem. We believe that current parsers are capable of recognizing meaning, and that future advances in NLP will allow machines to duplicate human judgments about text relevance.

There are two standard measures of performance in information retrieval: *recall* and *precision*. Recall is the proportion of relevant documents that are actually retrieved, and precision is the proportion of retrieved documents that are actually relevant. These two measures may be traded off one for the other, and the goal of information retrieval is to maximize them both.

Blair and Maron documented the poor performance of word-based retrieval in their study of the STAIRS retrieval system [1]. They concluded that it resulted from the false design assumption that "it is a simple matter for users to forsee the exact words and phrases that will be used in those documents they will find useful, and *only* in those documents." Two linguistic phenomena limit users' ability to choose effective search terms:

Polysemy, words having multiple meanings, reduces precision, because the words used to match the relevant documents may have other meanings that will be present in other irrelevant documents. For examples, a query about NASA launching a probe may retrieve texts about Congress launching a probe of NASA.

Synonymy, having multiple words or phrases to describe the same concept, reduces recall. If the author chooses one word or phrase to represent a concept, such as "liftoff," and the searcher looks for the word "launch," a simple word-based search will not retrieve the document.

	"stringer"	"ledger"	"post"	"copy"
Carpentry	stairway support	horizontal support	vertical support	
Accounting		book	to enter a transaction	
Computing			send a message	duplicate
Newpapers	employee		name of newspaper	text

Figure 1: Synonymy and Polysemy

Figure 1 shows just how promiscuous words can be. It shows four words and four domains that together have ten different meanings. A partial solution to the problem of polysemy is to index texts by word senses instead of words themselves [10, 14]. A partial solution to the problem of synonymy is to use a thesaurus to generate search terms that are synonyms of those given by the user [17].

2. CONCEPTUAL INFORMATION RETRIEVAL

We have built a more complete solution to the problems of polysemy and synonymy: the FERRET conceptual retrieval system. FERRET stands for "Flexible Expert Retrieval of Relevant English Text." Conceptual information retrieval is based on the "Relevance Homomorphism" assumption, depicted in Figure 2.

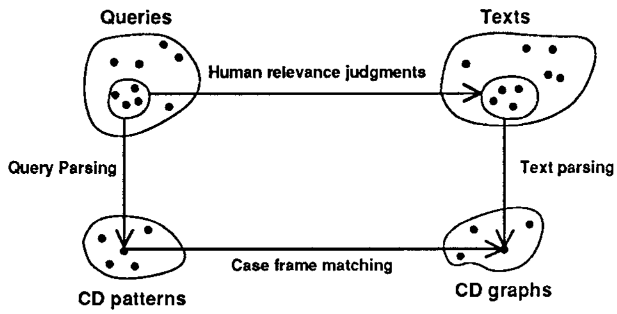

Figure 2: Relevance Homomorphism Assumption

This assumption states that the operations of query parsing, case frame matching, and text parsing form a homomorphism with human relevance judgements, and that parsing captures the "meaning" of the query and text as mentioned in Swanson's 5th postulate.

2.1. Related Work

The START system [9] parses the text and query into *T*-expressions, and matches them to answer questions about the text. But these expressions are not canonical, so if the query and the text use different words or phrases, the match will fail.

The SCISOR [8] and ADRENAL systems [3, 11] are both similar to FERRET in that they use case frames as their knowledge representation. Both systems use the TRUMP parser. The major difference between SCISOR and FERRET is that SCISOR is aimed more at question answering and at text extraction than information retrieval, and therefore it is difficult to compare SCISOR with traditional information retrieval systems. Recent work on ADRENAL has focused on "weak" NLP techniques, including syntactic phrases, phrase clustering, word sense disambiguation and sense-disambiguated thesauri.

2.2. Canonicality

Note that in Figure 2 the mappings from query to CD pattern and text to CD are many-to-one. This is a consequence of choosing a canonical knowledge representation. This uniqueness property greatly simplifies the matching process, or perhaps a more fair statement is that it improves the accuracy of matching for a given level of simplicity. This improved matching provides increased recall performance.

3. FERRET

FERRET is composed of three major modules: the text parser (a derivative of FRUMP [5, 6] called MCFRUMP), the case frame matcher, and the query parser (simulated for this study by MCFRUMP itself). For a detailed description of FERRET, see [12].

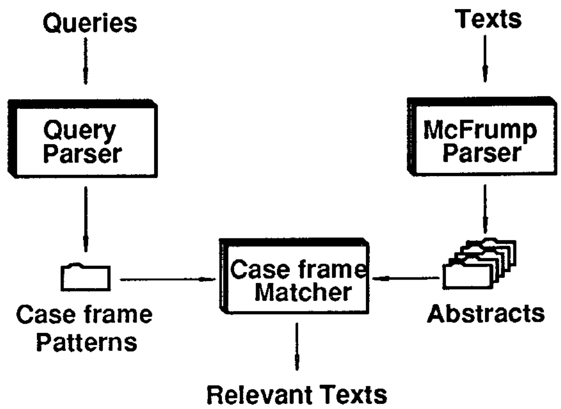

Figure 3: Simple Diagram of FERRET

3.1. FRUMP

The FRUMP parser forms the core of FERRET's text understanding ability; it was chosen over other kinds of parsers for several reasons. FRUMP demonstrated an impressive ability to read a broad class of real world texts. FRUMP's parsing "philosophy" of skimming to find the gist of a text seemed more appropriate for use in text retrieval than slower but more complete parsers. This style of parsing is also more robust, since the parser can tolerate skipping over complex material. And since it used case frames (in the form of conceptual dependency graphs and instantiated sketchy scripts) as its output, it fitted well to our model of text processing.

FRUMP stored its scripts in a discrimination tree (called an SSIDT). By clever use of this data structure, FRUMP was able to do script-based processing of inputs in time logarithmically related to the number of scripts in its database, and as a result it was very fast. FRUMP would "skim" its input texts to determine their main themes without slogging through each and every word.

Frump successfully analyzed 10% of the news stories on the UPI wire, requiring an average of 20 seconds per story. DeJong claimed that Frump is theoretically capable of understanding 50% of the UPI wire. "Successful" here means that it correctly understood the main theme of the text — some stories were not understood because although Frump correctly found the subthemes, it did not correctly identify the main point. Others failed because FRUMP misinterpreted the actions in the story, or simply lacked the necessary information to process the story.

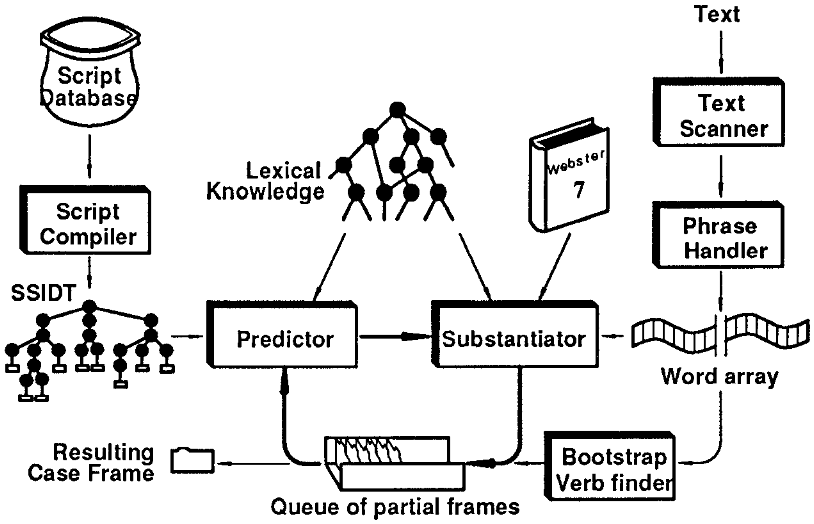

Figure 4: The MCFRUMP Parser

3.2. MCFRUMP

For the FERRET project, we re-implemented a simpler version of FRUMP in Franz Lisp (the original FRUMP was coded in M-expression Lisp, and compiled into UCI Lisp). The major difference is that MCFRUMP uses a simpler style of sketchy script. The original FRUMP script database was converted into the simpler form automatically, and scripts for newer domains (such as Astronomy and Dow Jones) were coded in the new script formalism. Figure 4 shows the internal structure of MCFRUMP, which is almost identical to that of FRUMP.

The *text scanner* is implemented as a pair of Lex programs that convert date, time and numeric expressions into tokens, to simplify later parsing. The *boot-strap verb finder* searches the word array left-to-right until a structure building word is found. That partial structure is added as the first entry to the queue of partial structures. This list is used by the *predictor* to perform a breadth first search of the possible concepts representing the input text. The predictor pops the top structure off of the queue and tries to extend it using the *substantiator*. If the substantiator can extend the structure, the new structure (or structures) is added to the end of the queue.

The substantiator uses the *lexical knowledge base* to determine whether a given word in the word array can have a particular meaning. For example, if the predictor is looking for a person, and the substantiator finds "Virginia," there is a match, and the substantiator builds a slot-filler containing a feminine first name, indicating a particular person. If the predictor had been looking for a location, the substantiator would have instead built a structure for the state of Virginia. The lexical knowledge is stored as a collection of FRAMEKIT frames [13]. If the word is not in the lexicon, the *dictionary interface* tries to find it in *Webster's*. If the word is found, the dictionary interface builds a FRAMEKIT frame and adds it to the lexical knowledge (for the current Lisp session only).

3.3. ASTRONOMY DOMAIN

We chose the Astronomy domain for the initial FERRET study, mainly because of the availability of texts from the UseNet newsgroup SCI.ASTRO (originally known as NET.ASTRO). Figure 5 shows two sample texts from this collection. These are scripts for the StarDate radio program heard daily on National Public Radio. We initially obtained 279 texts from the network news feed, and subsequently obtained another 1065 astronomy texts directly from the University of Texas McDonald Observatory. The average length of one text is 300 words or about 1800 characters.

The StarDate texts have one very desirable feature for natural language processing: they are narrative. Since they are meant to be read aloud, the information is conveyed sequentially without reference to charts, graphs, figures or references to other articles in the newsgroup. This matches most closely with the way FERRET reads text: one word at a time.

For the knowledge base, we started with the script database and lexicon from DeJong's FRUMP. Since this database was oriented towards the UPI newswire, there was very little conceptual overlap with the content of the SCI.ASTRO newsgroup. To handle astronomy, a domain description consisting of a script database and lexicon was written with 5 basic scripts (using 10 requests in all). This database took one graduate student about 40 hours to write. The basic scripts are:

- LAUNCH: sending a space probe to outer space
- ASTRO-POS: an astronomical body being in a particular relationship with another
- OUTER-SPACE-MOTION: an astronomical body moving in a given relationship to another body
- SPACETRAVEL: people traveling to or from an astronomical body
- ASTRO-VIEW: an astronomical body being viewed by or visible to someone

Article 1307 of 1351, Mar 3 02:00.
Subject: Pioneer 10
From: dipper@utastro
Organization: U. Texas, Astronomy, Austin, TX
Newsgroups: net.astro
Date: Mon, 3-Mar-86 02:00:18 EST

Pioneer 10 was the first spacecraft to venture
into the outer solar system. More -- coming up.

March 3 Pioneer 10

On today's date in the year 1972, NASA's Pioneer 10 spacecraft was launched toward the outer solar system. It was to become the first craft to travel beyond the asteroid belt -- and the first to encounter mighty Jupiter...

Script by Deborah Byrd.

Article 192 of 4 Aug 85.
Subject: Jupiter at Opposition
From: dipper@utastro
Organization: U. Texas, Astronomy, Austin, TX
Newsgroups: net.astro
Date: 4-Aug-85 02:00:00 EST

The planet Jupiter today falls behind Earth in
the race around the sun. More -- after this.

August 4 Jupiter at Opposition...

You can probably spot Jupiter now, and for the next few months, as the very bright object in the east each evening. Jupiter shines brilliantly by virtue of its large size, its bright cloud cover which reflects sunlight so well, and its relative nearness to Earth right now...

Script by Deborah Byrd.

Figure 5: Excerpts from Two StarDate Texts

To support these five scripts, the lexicon from Frump was extended with additional frames for objects and actions that occur commonly in astronomy texts. There are:

- 965 frames
- 442 words
- 272 word senses
- 251 concepts

Common extensions included words for seeing things (image, photograph, glimpse), for certain kinds of motion (orbit, launch, rise), plus a great many proper names (of stars, comets, planets, nebulae, and constellations). The content of the lexicon was adjusted by running the parser several times on the initial set of 279 StarDate radio texts. The scripts database was fixed, however, before the larger dataset was obtained from McDonald Observatory.

4. LEXICAL COVERAGE

FERRET uses four levels of lexical knowledge:

- Hand-coded lexicon, this includes phrases, word senses and conceptual entries for 12,281 frames altogether (including 3,369 nouns, 2,148 names, and 376 basic verbs).
- Synonyms entries from Webster's, 58,197 entries altogether, including 28,323 word senses defined solely as synonyms (e.g.: flummox :: CONFUSE).
- Near-synonyms. These are words used in the definition of unknown words that are in the lexicon. A small set of syntax rules determines whether a noun or a verb is a near-synonym (similar to the templates used in [14]). Webster's contains 24,741 verb entries and 68,837 nouns entries that can be matched against these syntax rules.
- Proper name rules, including list of 313 male first names, 386 female first names, and 17 ambiguous first names, plus phrase rules for recognizing names including known and unknown words.

Figure 6 shows how some of these levels of knowledge relate. When parsing the sentence:

Pioneer 10 was the first spacecraft to venture into the outer solar system.

FERRET first tries to find a structure building word (usually a verb). Scanning left to right, it finds "venture," which is not in the lexicon. None of the synonyms for venture are in FERRET's lexicon, so it checks the definitions for near-synonyms, and finds two: "to offer" (*MTRANS) and "to proceed" (*PTRANS). FERRET uses the script constraints to invalidate the interpretation that the outer solar system is receiving an offer, and correctly chooses the physical motion sense of "proceed."

Figure 7 shows a portion of the Webster's entry for "spot," and Figure 8 shows how the dictionary synonym entry NOTICE, which has a link to word sense SEE1 (*PERCEIVE), is used to parse the sentence

"You can probably spot Jupiter now."

The dictionary can be a dangerous place to look for information about words. The Webster's entry for "spot" includes 39 different word senses, and the synonyms given are themselves ambiguous. For example, one of the definitions of "spot" is "to lie at intervals," which could easily be mistaken by FERRET to attribute a meaning of "bearing false witness" to the word "spot." FERRET is able to use this highly ambiguous information only because of the strong semantic constraints imposed by its script database. This extra lexical coverage comes at the price of CPU time spent considering and rejecting many irrelevant senses of the words found in the dictionary.

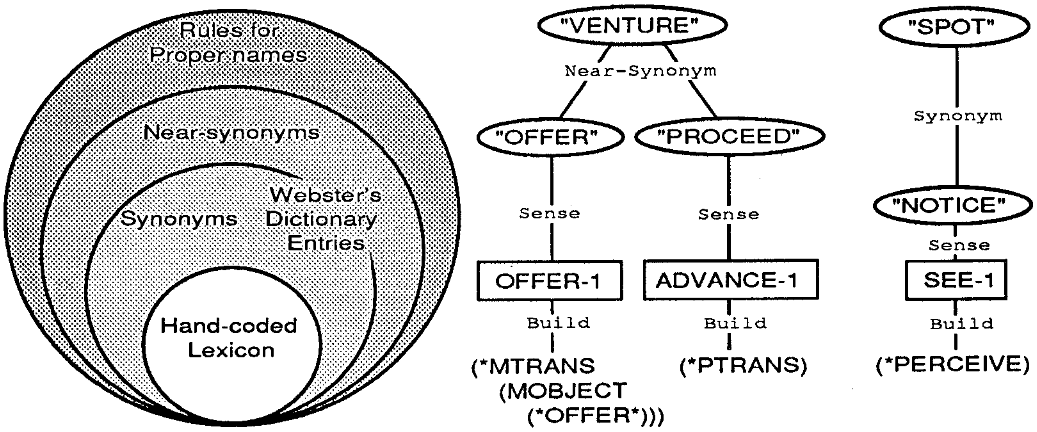

Figure 6: Levels of Lexical Knowledge

```
% lookup spot | dtl

(w7-spot-1 2 synonyms and 17 definitions omitted)

(w7-spot-2
    (pos verb)
    (variant
        (spotted ppart)
        (spotting ger))
    (synonym
        (disgrace) (stain) (identify)
        (detect notice) (stud))
    (definition
        (to stain the character or
         reputation of %colon %colon
         disgrace)
        (to mark in or with a spot
         %colon %colon stain)
        (to lie at intervals in or
         over %colon %colon stud)
        12 other definitions omitted)
    (related (spottable adjective)))

(w7-spot-3 7 definitions omitted)
```

Figure 7: Automatically formatted entries from Webster's

```
Sentence:
(p-2978 %cpt you can probably spot %cpt jupiter
 now %comma)

Using synonym of 'spot' => seel

******* spot (seel) builds:
    (cd (<=> (*perceive*)))

With modifiers (cd (<=> (*perceive*))
                   (mode (*probable*))
                   (tense (future)))

Parse Rule 2 fills 'actor' with (*you*)
Parse Rule 2 fills 'object' with (*jupiter*)

Accepting script
(astro-view-rl
    (&viewer (*you*))
    (&view-object (*jupiter*))
    (&view-date (*now*))
    (&view-mode (*probable*)))

Parsing time: 41.36 seconds
```

Figure 8: Using a synonym from Webster's to parse

5. RETRIEVAL PERFORMANCE

We evaluated the retrieval performance of FERRET on the astronomy texts described in Section 3.3. The collection of 1065 texts was sorted by creation date, and split into a training and an evaluation set. The training set contained the 533 odd-numbered texts. This set was used to refine and train the parser and its script learning component. The second set of 532 even-numbered texts was used to evaluate the retrieval performance, and was not seen by the author until after the parser and lexicon were complete.

After the parser and lexicon were tuned on the training set, 44 user queries were obtained from 22 computer science graduate students. Neither the parser nor the lexicon was changed after the queries were obtained. Each student was sent a survey form with three different paragraphs automatically chosen at random from both the training and evaluation test sets. The students were asked to think of a natural language query that would retrieve at least one of the paragraphs, to mark each paragraph that should be retrieved by the query, and to provide a keyword version of their query as well.

Since the FERRET project focused on parsing *texts* rather than *queries*, the MCFRUMP parser was used to substitute for the unimplemented query parser (query parsing has also been extensively studied, see for example [2]). Of the 44 queries, 22 were not parseable at all by MCFRUMP, and 16 required paraphrasing, while 6 were parsed as is (the complete list of parses and paraphrases is given in [12]). This way, any experimenter bias would be visible in the English paraphrases instead of hidden in parenthesis-laden case-frames. These patterns obtained from the 22 parsed sentences were matched against the parser output from the training and evaluation sets. The results are shown in Figures 9 and 10.

Training set, 22 queries			
	Keywords	FERRET	Learning
Precision	34.9%	44.8%	43.3%
Relative Recall	33.0%	43.4%	65.6%

Figure 9: Retrieval Performance on Training Set

Evaluation set, 22 queries			
	Keywords	FERRET	Learning
Precision	35.2%	49.4%	47.9%
Relative Recall	31.6%	45.3%	79.9%

Figure 10: Retrieval Performance on Evaluation Set

Evaluation Set, 5 queries			
	Keywords	FERRET	Learning
Precision	79.0%[1]	73.4%[2]	65.2%
Absolute Recall	19.4%	20.0%	52.4%

1. Keywords failed to retrieve any texts for 3 of the 5 queries, so this is the average precision for the other 2 queries.
2. FERRET failed to retrieve any texts for 1 of the 5 queries, so this is the average precision for the other 4 queries.

Figure 11: Absolute Recall Performance

The keyword versions of the queries were evaluated using a simple boolean keyword query system that provided functions for AND, OR, NOT, stemming, and ADJACENCY. In a few cases where the users had been overly specific, the keyword queries were modified to improve their performance. The *relative recall* performance was the proportion of the total number of documents retrieved that were retrieved by one method or the other. These are only upper bounds on the actual recall performance.

To estimate the *absolute* level of recall of both systems, a random sample of 5 of the 44 queries was chosen, and the entire evaluation set was manually searched for texts relevant to these 5 queries, and the recall rates were recomputed. The results of this study are shown in Figure 11. Regardless of the actual recall performance, the FERRET system retrieved 2.53 times as many relevant documents as the boolean keyword query. For more detailed discussion of this experiment, and especially for more discussion of FERRET's script learning component, see [12].

6. DICTIONARY PERFORMANCE

To determine how the dictionary interface affected parsing, a comparison study was done by parsing a sample of 50 texts both with and without the dictionary and with and without time limits. This was done separately from the retrieval study; the retrieval study was done with time limits of 8 minutes per text, 3 minutes per sentence, and with the dictionary enabled.

6.1. With Time Constraints

The normal processing mode for MCFRUMP includes a time limit of 8 minutes per text and 3 minutes per single sentence. Because the parser attempts to resolve all possible meanings of a sentence, a highly ambiguous sentence, or a sentence with many pronouns, can require hours to parse. These limits keep MCFRUMP from getting stuck on any one sentence or article.

Early tests of the parser were done on the training set with tighter limits of 4 minutes per story and 2 minutes per sentence. As a control, the parser was run without the dictionary interface using these same limits. The results are shown in Table 12. To our surprise, the parser understood more sentences *without* using the dictionary.

Closer investigation of parser traces showed that using the dictionary drastically increased the amount of ambiguity resolution required by the parser. Since more words now had plausible meanings, more work was required to confirm or disconfirm possible interpretations of any one sentence.

With the dictionary, sentences appearing earlier in the text were more likely to be parsed, but the parser ran out of time before it read very far into the text. Without the dictionary, early sentences were less likely to be parsed correctly, but there were slightly fewer misunderstood sentences, and the parser often found simpler sentences later in the text that it could understand.

Effect of Dictionary, With 4 minute time limit per text			
Number of texts: 533	With W7	Without W7	Change
Total sentences Misses	3312 2704	7492 5981	-55.8%
Sentences with some parse	608	1511	-59.8%
Percent "understood" Sentences with script (percent of sentences read)	22.5% 449 14%	25.3% 1194 16%	-11.1%
Sentences with partial parse Size of inst. scripts CPU hours to parse texts	159 408kbytes 56.6	317 820kbytes 43.3	 -50.2%

Figure 12: Effect of Dictionary with time limits

Effect of Dictionary, With no time limit			
Number of texts: 50	With W7	Without W7	Change
Total sentences Misses	986 704	986 784	
Sentences with some parse Percent "understood"	282 28.6%	203 20.6%	+38.9
Sentences with script (percent of sentences read)	168 17.0%	147 14.9%	+14.3
Sentences with partial	114	56	+103.6
Size of inst. scripts CPU hours to parse	151kbytes 43.5	109kbytes 9.7	+40.0%

Figure 13: Effect of Dictionary without time limits

The net result was that more sentences were understood with slightly higher precision without the dictionary. But in some cases, these sentences appeared much later in the text and sometimes dealt with secondary topics.

6.2. Without Time Constraints

A second comparison study was performed, this time without any time limits on the parser. This way, the effect of the dictionary on understanding could be studied in isolation from time effects. Because of the increased time needed, a smaller sample of 50 texts was randomly chosen from the training set and used for this study. The results are shown in Table 13. This time, the parser understood more sentences with the dictionary than without.

Investigation of parser traces from the second study showed that, like the first test, early sentences were more likely to be understood by using the dictionary, and that the abstracts produced were somewhat more representative of the texts (approximately one fifth of the texts had much more accurate abstracts because the early sentences were better understood).

7. SUMMARY AND FUTURE WORK

Before summarizing, we present two directions for future work: generalizing the skimming parser to languages other than English, and investigating the performance of conceptual retrieval on standard IR corpora.

7.1. Generalizing to other languages

We are investigating the ability of a FRUMP style parser to process text in languages other than English. Figure 14 shows a portion of an article from the OPS 27Reunión. The English translation is

> ''As of July 1, 1989, a cumulative total of 167,373 AIDS cases had been officially reported to the the World Health Organization Global Program on AIDS, originating in 149 countries.''

Figure 15 shows the same sentence translated into Japanese. Figures 16 and 17 show the actual instantiated sketchy scripts produced by the parser (the parse shown in Figure 17 was produced from the segmented Romaji text). Both texts were parsed using the same script database, and

Spanish:

From: OPS 27Reunión
Date: Sep 5, 1989

Al 1 de julio de 1989, un total acumulativo de 167.373 casos de SIDA se notificaron oficialmente al Programa Global de la Organización Mundial de la Salud sobre el SIDA (PGS), procedentes de 149 países.

Figure 14: Sample Spanish Text

Kanji:

1989年7月1日現在、149ヶ国で発生したトータルエイズ ケース数、167373、が PGS に報告されました

Romaji:

```
1985-7-1 genzai, 149 kakoku de hasseishita
tootaru eizu keesu suu, 167373, ga PGS ni
hookoku sare masita.
```

Figure 15: Sample Japanese text

```
Activated
(incidence-rl
    (&agency (*pgs*))
    (&disease
        (*case* (quantity (167373.0))
                (disease (*aids*))))
    (&origin (*country* (quantity (149.0))))
    (&date (*date* (month july) (day 1)
                   (year 1989))))

Parsing time: 32.48 seconds
```

Figure 16: Instantiated script from Spanish parse

```
Activated
(incidence-rl
    (&agency (*pgs*))
    (&disease
        (*case* (disease (*aids*))
                (quantity (167373.0))))
    (&origin (*country* (quantity (149.0))))
    (&date (*date* (month july) (day 1)
                   (year 1986))))

Parsing time: 29.43 seconds
```

Figure 17: Instantiated script from Japanese parse

the resulting frames differ only in the order of the slots in the *CASE frame (the order is different because in the Spanish text the quantifier "167.373" occurs before the word "casos," and in the Japanese example the quantifier comes after the word "keesu").

Since order of slots is unimportant in case frames, the two output frames are identical. One can see that an information retrieval system based on this kind of matching can allow a user to retrieve documents written in one language using queries written in a different language. The changes in the Lisp source for MCFRUMP to parse these examples were less than 5% of the total lines of code in the system.

7.2. Future Evaluation

The FERRET project is only a single example of a conceptual retrieval system outperforming a standard information retrieval technique (boolean keyword query). Furthermore, boolean systems are not the best keyword-based retrieval systems: a boolean system was used here mainly because of its convenience as a test standard. The next obvious step is to evaluate FERRET's performance on one or more standard IR corpora. Such a study was beyond the scope of the initial FERRET effort, but is necessary to allow comparison of conceptual retrieval with more sophisticated IR techniques such as relevance feedback [15] or latent semantic indexing [4, 7].

7.3. Conclusion

Conceptual information retrieval using an effective, canonical knowledge representation and case frame matching is an alternative to word-based retrieval methods, and was shown in this study to increase both recall and precision over standard boolean keyword query.

This study shows the ability of text skimming parsers to extract semantic content from a medium-sized corpus of unedited English text (1065 texts is large by AI standards, and minuscule by IR standards). FERRET also demonstrates the use of machine-readable dictionaries and machine learning to improve the parser's performance.

8. Acknowledgments

The StarDate radio texts used in this study are Copyrighted 1985 and 1986 by the McDonald Observatory, University of Texas at Austin, and were used with permission. The Japanese parsing example resulted from joint work with Dr. Teruko Mitamura of the Carnegie Mellon Center for Machine Translation.

This work was supported in part by the Office of Naval Research (ONR) under contract number N00014-84-K-0345, in part by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. 4976, monitored by the Air Force Avionics Laboratory under contract F33615-87-C-1499, and in part by a grant from Dow Jones and Company.

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the ONR, DARPA, DOD, Dow Jones and Company, or the U.S. Government.

9. REFERENCES

1. D. C. Blair and M. E. Maron. "An Evaluation of Retrieval Effectiveness for a Full-Text Document-Retrieval System." *Comm. ACM 28*, 3 (March 1985), 289-299.

2. J. G. Carbonell, W. M. Boggs, M. L. Mauldin and P. G. Anick. "The XCALIBUR Project, A Natural Language Interface to Expert Systems." *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, 1983. (IJCAI-83).

3. W. B. Croft. "Approaches to Intelligent Information Retrieval." *Information Processing and Management 23*, 4 (1987), 249-254.

4. S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer and R. Harshman. "Indexing by Latent Semantic Analysis." *Journal of the American Society for Information Science 41*, 6 (September 1990), 391-407.

5. G. F. DeJong. "Prediction and Substantiation: a New Approach to Natural Language Processing." *Cognitive Science 3* (September 1979).

6. G. F. DeJong. "An Overview of the FRUMP System." In *Strategies for Natural Language Processing*, W. G. Lehnert and M. H. Ringle, Ed., Lawrence Erlbaum Associates, Hillsdale, NJ, 1982, ch. 5, pp. 149-176.

7. G. W. Furnas, S. Deerwester, S. T. Dumais, T. K. Landauer, R. A. Harshman, L. A. Streeter and K. E. Lochbaum. "Information Retrieval using a Singular Value Decomposition Model of Latent Semantic Structure." *Proceedings of the Eleventh Annual International ACMSIGIR Conference on Research and Development in Information Retrieval*, June, 1988, pp. 465-480. (SIGIR-88).

8. P. S. Jacobs and L. F. Rau. "SCISOR: Extracting Information from On-line News." *Comm. ACM 33*, 11 (November 1990), 88-97.

9. B. Katz. Using English for Indexing and Retrieving. A. I. Memo 1096, Massachusetts Institute of Technology, October, 1988. A shorter version of this paper appears in the *Proceedings of the Conference on User-oriented Content-based Text and Image Handling, RIAO 1988.*

10. R. Krovetz and W. B. Croft. "Word Sense Disambiguation Using Machine-Readable Dictionaries." *Proceedings of the Twelfth Annual International ACMSIGIR Conference on Research and Development in Information Retrieval*, June, 1989, pp. 127-136. (SIGIR-89).

11. D. D. Lewis, W. B. Croft and N. Bhandaru. "Language-Oriented Information Retrieval." *International Journal of Intelligent Systems 4*, 3 (Fall 1989).

12. M. L. Mauldin. *Information Retrieval by Text Skimming*. Ph.D. Th., School of Computer Science, Carnegie Mellon University, August 1989.

13. E. Nyberg. The FrameKit User's Guide. Center for Machine Translation, Carnegie Mellon University, 1988.

14. Y. Ravin. "Disambiguating and Interpreting Verb Definitions." *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, June, 1990, pp. 260-267. (ACL-90).

15. G. Salton, E. A. Fox and E. Voorhees. "Advanced Feedback Methods in Information Retrieval." *Journal of the American Society for Information Science 36*, 3 (1985), 200-210.

16. D. R. Swanson. "Historical Note: Information Retrieval and the Future of an Illusion." *Journal of the American Society for Information Science 39*, 2 (1989), 92-98.

17. Y.-C. Wang, J. Vandendorpe and M. Evens. "Relational Thesauri in Information Retrieval." *Journal of the American Society for Information Science 36*, 1 (1985), 15-27.