

Natural Language Technology in Precision Content Retrieval

Jacek Ambroziak and William A. Woods

Natural Language Technology in Precision Content Retrieval

Jacek Ambroziak and William A. Woods

SMLI TR-98-69

December 1998

Abstract:

This paper describes a new approach to information access that combines techniques from natural language processing and knowledge representation with a new technique for relevance estimation and passage retrieval. Unlike many attempts to combine natural language processing with information retrieval, these results show significant benefit from using linguistic knowledge. Subsumption technology is used to automatically integrate syntactic, semantic, and morphological relationships among concepts that occur in the material, and to organize them into a structured conceptual taxonomy that is efficiently usable by retrieval algorithms and also effective for browsing.

Keywords:

Natural Language Processing, Taxonomic Subsumption, Information Retrieval, Bridging Theory and Practice, Industrial Applications.

Note:

This paper appeared in the proceedings of the International Conference on Natural Language Processing and Industrial Applications (NLP+IA 98), August 18-21, 1998, Moncton, New Brunswick, Canada.



M/S MTV29-01
901 San Antonio Road
Palo Alto, CA 94303-4900

email address:
william.woods@sun.com
jacek.ambroziak@sun.com

© 1998 Sun Microsystems, Inc. All rights reserved. The SML Technical Report Series is published by Sun Microsystems Laboratories, of Sun Microsystems, Inc. Printed in U.S.A.

Unlimited copying without fee is permitted provided that the copies are not made nor distributed for direct commercial advantage, and credit to the source is given. Otherwise, no part of this work covered by copyright hereon may be reproduced in any form or by any means graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an information retrieval system, without the prior written permission of the copyright owner.

TRADEMARKS

Sun, Sun Microsystems, AnswerBook, SearchIt, and the Sun logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

For information regarding the SML Technical Report Series, contact Jeanie Treichel, Editor-in-Chief <jeanie.treichel@eng.sun.com>.

Natural Language Technology in Precision Content Retrieval

Jacek Ambroziak and William A. Woods

Sun Microsystems Laboratories
One Network Drive
Burlington, Massachusetts 01803-0902

1 Introduction

Increasingly, information retrieval systems are being called upon to support an information seeker who needs to quickly find some specific item of online information and then use it for some purpose. This need is different from that of a scholar/analyst looking for comprehensive references on a topic. Queries are typically only a few words and the information sought is often contained in a small passage of only a sentence or two. This need is not well served by traditional document retrieval systems, since the standard vector-based approach is not particularly effective for short queries, and returning entire documents can require a lot of subsequent reading to find the specific information sought. Often the information seeker fails to find what is wanted because the words used in the request are different from the words used by the author of the needed material.

Ideally, one would like to find useful information in response to spontaneously worded requests that do not require a lot of thought in order to be effective. The objective is to find the needed information with a minimum of effort spent thinking about how to express the request and a minimum of time spent reading through returned material. Meeting this objective requires rethinking the entire information retrieval process. This paper describes a project that has tackled this problem from first principles, evolving from a theoretical hypothesis, through subsequent experimentation and discovery, to a useful

technology with successful applications well-suited for fine-grained, specific searches.

The question of whether linguistic knowledge can make a useful contribution to information retrieval has been a debated topic in the information retrieval community, due to mixed and often negative results of previous attempts to use such knowledge (Fagan 89; Lewis & Sparck Jones 96; Sparck Jones 88; Varile & Zampolli 97). In particular, the use of multiword phrases has typically yielded no significant improvement.

This project began with the hypothesis that multiword phrases could be useful if the system knew enough about the meanings of such phrases to do more with them than simply look for exact matches. Specifically, the hypothesis was that by organizing recognized phrases by the relationship of generality, using subsumption technology derived from knowledge representation research (Woods 91), one could automatically relate more general words and phrases to more specific words and phrases that they subsume and that this would be useful for dealing with paraphrase relationships between query terms and index terms.

2 Conceptual Indexing

As suggested above, we attempt to go beyond standard word and phrase matching approaches by making use of knowledge of the conceptual structure and meanings of words and phrases and using formal subsumption techniques to relate more general

concepts to more specific ones. More general concepts are said to "subsume" more specific concepts, and algorithms are able to automatically organize words and phrases into conceptual taxonomies in which each concept is linked to its most-specific subsumers (the most specific concepts that are more general than the concept in question). These algorithms use the conceptual structures of phrases to align corresponding constituent elements and use semantic subsumption "axioms" drawn from a predefined lexicon to relate constituents to each other. For example, "automobile cleaning" subsumes "car washing" because of the way corresponding elements of these phrases are aligned and the facts (expressed as axioms) that a car is a kind of automobile and washing is a kind of cleaning.

As part of this project, a conceptual indexing engine known as ConceptStore was developed, which can automatically organize words and phrases into a conceptual taxonomy on the basis of the subsumption relationships between their conceptual structures. The resulting taxonomy is stored in a persistent knowledge base that remembers all of the words and phrases found in a body of material, together with their conceptual structures, the positions where they occur in the material, and their subsumption relationships to other words and phrases. We refer to this as a conceptual index of the material. ConceptStore is an efficient C++ program that has been used to organize collections of more than three million concepts. ConceptStore also contains algorithms for efficiently locating passages of material where concepts occur near each other.

In the experiments described here, ConceptStore was driven by a Lisp-based linguistic component that is part of a Pilot indexer that scans text, extracts words and basic noun and verb phrases, analyzes their structure and meanings, and passes the result

to ConceptStore. In other implementations of this technology, the Lisp-based linguistic component has been replaced by a different phrase extraction component written in C and C++.

The utility of a conceptual index can be illustrated by an example (shown in Figure 1) from the conceptual index of a collection of encyclopedia articles about animals. An initial request for "brown fur" retrieved the phrase (BROWN FUR) and the subsumed phrases (GRAY BROWN FUR), (RICH BROWN FUR), and (WHITE-SPOTTED BROWN FUR). However, a display of more general concepts showed that the query was subsumed by (BROWN COAT), revealing that the request was inadvertently more specific than intended. Generalizing the request to (BROWN COAT) produced the substantially more useful collection of concepts shown in Figure 1. By displaying the more general concepts in the taxonomy that subsume the stated request, the system unobtrusively suggested an exceedingly useful generalization of the request.

Note that morphological relationships are incorporated into the subsumption framework by treating derived and inflected forms of words as subsumed by their base forms. For example, "brownish" is subsumed by "brown." A lexicon of syntactic and semantic information about known words and an aggressive morphological analysis of unknown words are used to support this processing. Morphological variations and terminology variations are thus automatically related in the conceptual taxonomy, and syntactic relationships are incorporated into the structures of the parsed phrases. Different senses of words can be represented by concepts that have different places in the taxonomy but are subsumed by a concept corresponding to an abstract meaning of the undisambiguated word. When presented with a query, a retrieval algorithm in ConceptStore searches through the conceptual taxonomy for

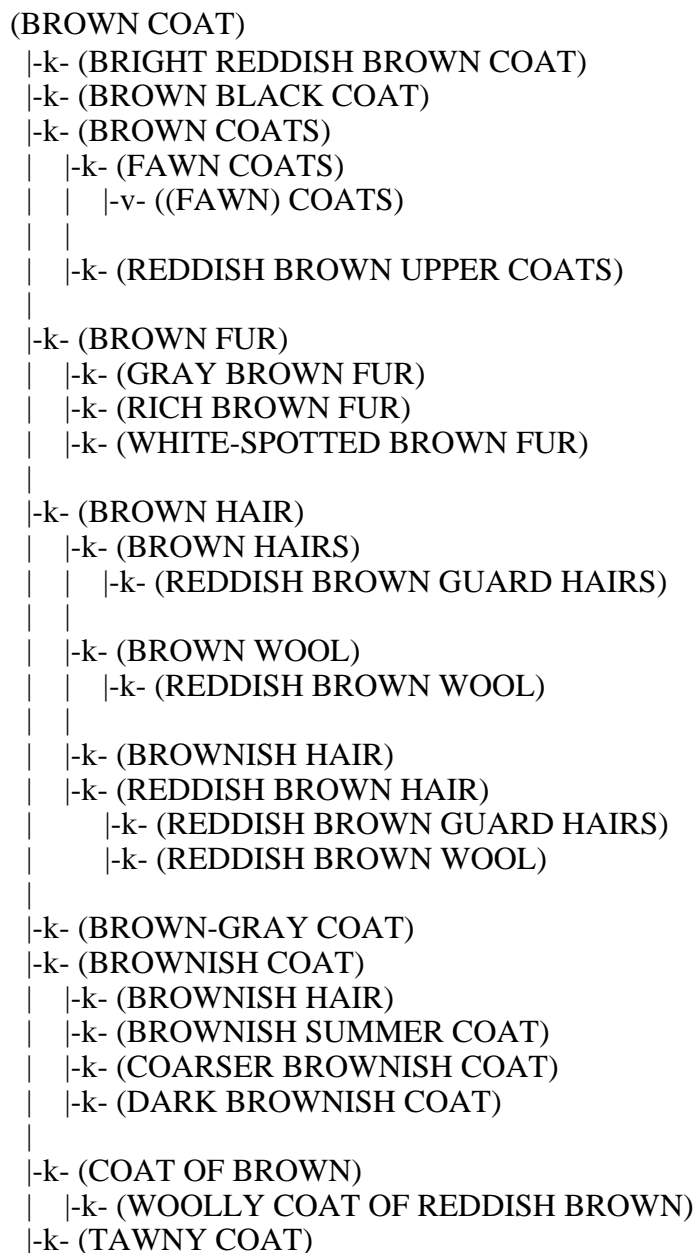


Figure 1: A fragment of a conceptual taxonomy, illustrating the utility of navigating in conceptual space.

subsumed concepts and uses the positions of those concepts in the indexed material to find specific passages that are likely to address the information needs of the request.

3 Specific Passage Retrieval

Conceptual subsumption by itself is able to

find specific locations in the indexed material where concepts subsumed by a query concept occur. However, the elements of a concept being sought are often not explicitly related in the indexed material in a single phrase that is formally subsumed by the query. For example, in the above "brown coat" example, the query did not subsume any phrase in the passage,

"The coat is reddish brown." In this case we might imagine a rule that would derive a phrase "x y" from any assertion of the form "The y is x," but this is just a special case of a general problem in which some of the relationships between elements of a concept are left implicit in a text and need to be inferred.

Since mastering the subtleties of implicit inference is well beyond the current state of the art of natural language processing, we developed a technique called "relaxation ranking" in which an algorithm looks for places where as many as possible of the different elements of a query occur near each other, preferably in the same form and word order and preferably closer together. Such passages are ranked by a penalty score that measures the degree of deviation from an exact match of the requested phrase, with smaller penalties being preferred. Differences in morphological form and formal subsumption of index terms by query terms introduce small penalties, while intervening words and sentence boundaries introduce more significant penalties. Elements of a query that cannot be found nearby introduce a substantial penalty that depends on the syntactic categories of the missing words.

We call the resulting approach "precision content retrieval." It consists of two parts: conceptual indexing builds structured conceptual taxonomies of words and phrases extracted from the indexed material, and specific passage retrieval identifies candidate passages in the material and ranks them according to their relaxation-ranking penalties. Like other passage retrieval approaches (Callan 94; Kaszkiel & Zobel 97; Salton et al. 93), this identifies relevant passages rather than simply identifying whole documents or Web pages, but unlike approaches that involve initially segmenting the material into paragraphs or other small passages, this approach dynamically constructs passages in response to requests, identifying passages

whose size may range from a single word or phrase to several sentences or paragraphs, depending on how much context is required to capture the various elements of the request. This penalty-based relaxation ranking approach turns out to produce exceptionally useful rankings of specific passages, supporting efficient location of relevant information in response to fine-grained specific requests.

In a user interface to such a system, retrieved passages will be reported to the user in increasing order of penalty, together with the penalty score, information about which target terms match the corresponding query terms (since they may be different), and the content of the identified passage with some surrounding context. In a typical application, results are presented in a hypertext interface that allows the user to click on any of the presented items to see that passage in the context of its source document. For example, in one application of this technology to a data base of transcripts of TV news broadcasts, the best match found in response to a request for "recent arson" is displayed as follows:

1. -0.61 RECENT (MARIN ARSONIST)
Morn.25.Oct@0-Caption.data

39 THE HUNT FOR THE MARIN
40 ARSONIST.
41 THIS MORNING.. THE SEARCH IS
42 FOCUSING ON NOVATO.. WHERE THE
43 MOST RECENT FIRES WERE SET.

The format of this display is:
<rank number> <penalty score> <term matches>
<document reference name>
<empty line>
<passage excerpt of information content>

This says that the passage is ranked number 1 and has a penalty of 0.61. The rest of the

first line shows that the first term "recent" in the request is matched exactly by the term "recent" in the text, and that the second term "arson" in the request is matched by the phrase "Marin arsonist" in the text. (Individual word matches are displayed here as single words, while phrases are displayed as sequences of words enclosed in parentheses.) A standard reference to the news broadcast in which the passage was found is then given [Morn.25.Oct@0-Caption.data], which is also active as a Hypertext link to take you to the corresponding passage in the indicated transcript. In addition, the actual content of the passage in which the terms were found is displayed in order to provide more information about the match to help you to decide whether you want to actually go see the passage in context.

Notice that individual terms in the request can be matched by phrases in the text that are judged more specific than the requested term and that the query term "arson" is automatically matched against a different form of the word ("arsonist") in the text phrase, without the user having to specify "wild card" or "truncation" or "stemming" operators. Notice also that the query term "recent" does not directly modify the word "arson" in the passage, but that the relative proximity of the two terms in the passage does reflect an inferable relationship between the two. (The deictic interpretation of "recent" relative to the date of the request is beyond the current state on the system, but is planned for future versions.)

In addition to the retrieved passages, the user can be presented with a display of portions of the conceptual taxonomy related to the terms in the request. As illustrated earlier, this frequently reveals generalizations of the request that will find additional relevant information, and it also conveys an understanding of what concepts have been found in the material that will be matched by

the query terms. More details describing this technology can be found in (Woods 97).

4 Experimental Evaluation

In order to evaluate the effectiveness of the above techniques, we collected a set of 90 queries from a naive user of UNIX®, 84 of which could be answered from the online UNIX documentation known as the *man* (for "manual") pages. A set of "correct" answers for each of these 84 queries was manually determined by an independent UNIX expert, and a snapshot of the UNIX man pages collection was captured and indexed for retrieval. Since the relaxation ranking algorithm retrieves passages within the man pages, rather than whole man pages, in order to compare this methodology with classical document retrieval techniques, we assign a ranking score to each man page equal to the penalty score of the best ranked passage that it contains. In this fashion, we are able to compare different versions of the relaxation ranking algorithm with different classical document retrieval methods, at least as far as the retrieval of whole documents is concerned.

In rating the performance of a given method, in addition to the traditional average recall and precision values, we also compute a measure we call the "success rate," which is simply the percentage of queries for which there is an acceptable answer in the top ten hits. The latter is the principal factor on which we base our evaluations, since for this application, the user is not interested in subsequent answers once an acceptable answer has been found (indeed for most of the 84 requests there is only one correct answer), and it has been observed in applications such as Sun's AnswerBook™ that users will generally stop looking if an answer is not found in the first ten hits. Success rate is a more relevant measure than recall for this kind of application, since finding one answer for each of two requests is a substantially better

result than finding two answers to one request and none for another, and for this application it doesn't matter how many additional relevant answers there might be after an acceptable one is found.

It is important to keep in mind that the major benefit of retrieving specific relevant passages is not being tested by this experimental methodology. Moreover, the objective of comparing the method to classical document retrieval methods cuts at cross purposes with the objective of evaluating the quality of the retrieved passages. For example, for one query, "print a file," the relaxation ranking algorithm found a passage in the man page for the file manager that is undeniably a correct answer to the query, even though the file-manager man page is not primarily about printing files and had not been judged by the human evaluator to be a correct document to retrieve in response to this query. Evaluating the quality of the retrieved passages themselves requires a different methodology that is not presented here.

These experiments were conducted using an experimental retrieval system that we named Recall II. The linguistic knowledge sources used in these experiments included a core lexicon of approximately 18,000 words, a substantial set of morphological rules and specialized morphological algorithms covering inflections, prefixes, suffixes, lexical compounding, and a variety of special forms such as numbers, ordinals, roman numerals, dates, phone numbers, acronyms, etc. In addition, we made use of a lexical subsumption taxonomy of approximately 3000 lexical subsumption relations, and a small set of semantic entailment axioms. The database was a snapshot of the UNIX man pages, consisting of approximately 1800 files of varying lengths and constituting a total of approximately 10 megabytes of text.

Table 1 shows the results of comparing three versions of this technology with a textbook implementation of the standard tf.idf

algorithm, a state-of-the-art commercial search engine and a Bayesian probabilistic retrieval algorithm. In this table, tf.idf refers to an implementation of a term weighted inverse document frequency algorithm taken directly from Salton's textbook (Salton 89). SearchIt™ refers to a search engine developed at Sun Microsystems, Inc. that combines a simple morphological query expansion with a state-of-the-art commercial search engine, and TWITF is an implementation of a simple Bayesian probabilistic retrieval algorithm (named for "term weighted inverse term frequency"). Recall II is the full system described above, using all of its knowledge sources and morphological analysis rules, while -morph refers to this system with its dynamic morphological analysis rules turned off, and -knowledge refers to this system with all of its knowledge sources and rules turned off. The table presents the success rate and the measured recall and precision values at the cutoff point for 10 retrieved documents.

Table 1. A comparison of different retrieval techniques

System	Success Rate	Recall	Precision
tf.idf	28.6%	14.8%	2.9%
SearchIt	44.0%	28.5%	7.4%
TWITF	47.6%	28.4%	7.4%
Recall II	60.7%	38.6%	7.3%
-morph	50.0%	(not measured)	
-knowledge	42.9%	(not measured)	

5 Discussion

The table shows that for this task, the relaxation ranking passage retrieval algorithm without its supplementary knowledge sources (Recall II -knowledge) is roughly comparable in performance (42.9% versus 44.0% success rate) to a state-of-the-art commercial search engine (SearchIt) at the pure document

retrieval task (neglecting the added benefit of locating the specific passages). Adding the knowledge in the core lexicon (which includes morphological relationships, semantic subsumption axioms, and entailment relationships), but without dynamic morphological analysis of unknown words (Recall II -morph), significantly improves these results (from 42.9% to 50.0%). Further adding the dynamic morphological analysis capability that automatically analyzes unknown words (deriving additional morphological relationships and some semantic subsumption relationships) significantly improves that result (from 50.0% to 60.7%). In contrast, we found that adding the same semantic subsumption relationships to the commercial search engine, SearchIt, using its provided thesaurus capability degraded its results, and results were still degraded when we added only those facts that we knew would help find relevant documents. It turned out that the additional relevant documents found were more than offset by additional irrelevant documents that were also ranked more highly.

It is worth noting that these results are not monotonic. That is, when we compare one method versus another and obtain a higher success rate, it is not correct to assume that the higher performing method retrieved all of the results of the lower one plus more. Rather, a change in method tends to give up some results and gain others. The key issue is whether the change nets out positive or negative. Generally, using more knowledge with the relaxation ranking algorithm picks up better passages and pushes lesser quality passages down in the rankings, resulting in lower penalty scores for the document as a whole.

6 Conclusion

We have described an information retrieval method in which specific passages within texts are dynamically found in response to a query, and these passages are ranked based on a penalty-based score that measures the degree of relaxation from an ideal standard required for a match. This is a different approach from previous methods of passage retrieval. This algorithm has been implemented, an initial knowledge base of linguistic information has been developed for its use, and the technology has been applied in several applications at Sun. Experiments have shown that the use of linguistic knowledge can significantly improve performance in finding appropriate answers to specific queries when incorporated into this relaxation ranking algorithm. The differences in performance supporting this claim are statistically significant at a better than .005% level. It appears that the relaxation ranking algorithm figures crucially in this success, since the addition of such linguistic knowledge to standard information retrieval models typically degrades retrieval performance rather than improving it. The linguistic knowledge used in these experiments includes morphological relationships between words, taxonomic relationships between concepts, and general semantic entailment relationships between words and concepts.

Acknowledgments

Many people have been involved in creating the conceptual indexing and retrieval system described here. These include: Gary Adams, Lawrence Bookman, Cookie Callahan, Chris Colby, Jim Flowers, Ellen Hays, Robert Kuhns, Patrick Martin, Peter Norvig, Tony Passera, Philip Resnik, Robert Sproull, and Mark Torrance.

References

- (Callan 94) J. P. Callan: "Passage-level evidence in document retrieval," *SIGIR 1994*: 302-309.
- (Fagan 89) J. L. Fagan: "The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval," *Journal of the American Society for Information Science*, 40(2):115-132, March 1989.
- (Kaszkiel & Zobel 97) Marcin Kaszkiel and Justin Zobel, "Passage retrieval revisited," *SIGIR 1997*: 178-185.
- (Lewis & Sparck Jones 96) David D. Lewis, Karen Sparck Jones: "Natural Language Processing for Information Retrieval," *CACM* 39(1): 92-101, 1996.
- (Salton 89) Gerard Salton: *Automatic Text Processing*, Addison Wesley, Reading, MA, 1989.
- (Salton et al. 93) G. Salton, J. Allan, and C. Buckley, "Approaches to passage retrieval in full text information systems," *SIGIR 1993*: 49-58.
- (Sparck Jones 88) Karen Sparck Jones: "A Look Back and A Look Forward," *SIGIR 1988*: 13-29.
- (Varile & Zampolli 97) Giovanni Varile and Antonio Zampolli (Editors), *Survey of the State of the Art in Human Language Technology*, Cambridge Univ Press, 1997.
- (Woods 91) William A. Woods; "Understanding Subsumption and Taxonomy: A Framework for Progress," in John Sowa (Editor), *Principles of Semantic Networks*, Morgan Kaufmann, San Mateo, CA., 1991.
- (Woods 97) William A. Woods: *Conceptual Indexing: A Better Way to Organize Knowledge*, Technical Report SMLI TR 97-61, Sun Microsystems Laboratories, Mountain View, CA., 1997.

About the Author

William A. Woods is a Principal Scientist and Distinguished Engineer at Sun Microsystems Laboratories in Burlington, Massachusetts. He is internationally known for his research in natural language processing, continuous speech understanding, and knowledge representation, and is currently interested in technology for improving people's access to online information. He earned his doctorate at Harvard University, where he served as an Assistant Professor and later as a Gordon McKay Professor of the Practice of Computer Science. He is a past president of the Association for Computational Linguistics, a Fellow of the American Association for Artificial Intelligence, and a Fellow of the American Association for the Advancement of Science.

Dr. Woods worked at Bolt Beranek and Newman Inc. (BBN) when the Internet (then Arpanet) was being invented, and he built one of the first natural language question answering systems to answer questions about the Apollo 11 moon rocks for the NASA Manned Spacecraft Center. He was Principal Investigator for BBN's work in natural language processing and knowledge representation and for its first project in continuous speech understanding. Subsequently, he was Principal Scientist for Applied Expert Systems, Inc. and Principal Technologist for On Technology Inc.