AIM: A New Approach for Meeting Information Needs

Rob Barrett and Ted Selker
IBM Almaden Research Center
650 Harry Rd.
San Jose, CA 95121
{barrett, selker} @ almaden.ibm.com

ABSTRACT

AIM (Automatic Informative Metadata) extends the existing keyword-matching approach to information retrieval by enabling users to express information needs as a set of AIM descriptors. These descriptors are chosen to fit the criteria that searchers use to judge search results. The descriptors can also be automatically derived from plain text documents, enabling the effective matching of information needs and documents. AIM seeks to enable users to interact more naturally with information collections by increasing the expressiveness of the communication. We present details of descriptors for language, subject, writing style, date, and profession together with scenarios which demonstrate how these descriptors can be used to enhance the information retrieval environment.

KEYWORDS: information retrieval, text analysis, user models, document indexing

INTRODUCTION

In this paper, we argue that the next generation information retrieval systems should support users through dialogs which address the multidimensional nature of their information needs. Information retrieval is a fundamental computer application that has been an active area of research since computers were first created. The expressiveness of the user interface for the information retrieval system is central to making the information accessible. We seek to join together an understanding of the user's needs and the capabilities of text processing to produce more usable information retrieval systems.

We have been developing a new system, called AIM (Automatic Informative Metadata), that enables users to express their information needs more fully than the traditional keyword approach alone. AIM forms a multi-faceted bridge between a user's information need and the computer's vast storehouse of data (see Figure 1). We connect user-meaningful information descriptions with computationally-tractable textual analysis. User's needs and electronic documents are matched using a common set of AIM descriptors. AIM descriptors are both automatic, in that they are generated from documents without human intervention, and informative, in that they express information in ways that are useful to users. We hope that these methods will stimulate the development of information retrieval systems that converse with users in terms closer to the users' natural descriptions of their information needs. In what follows, we describe 1) why AIM's method of information description is necessary, 2) the descriptor generation techniques we have developed, 3) results of preliminary tests, and 4) example scenarios which use these techniques.

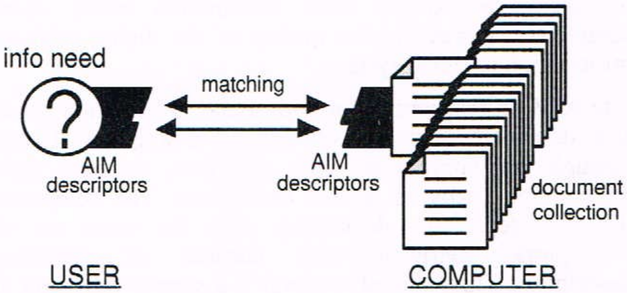

Figure 1: A user's information need is expressively represented in terms of a set of AIM descriptors. Documents in a collection are represented likewise. Multidimensional matching between the information need and the documents can then be accomplished.

MOTIVATION

What makes a document useful?

Many factors contribute to an information seeker's decision as to whether a document is useful. Existing keyword-based systems inherently assume that document usefulness can be derived from the presence or absence of keywords, either through boolean operators or statistical and probabilistic calculations [1]. Cool *et al* studied which aspects of documents are important to searchers engaged in a particular task [3]. In particular, college students were asked to write an essay, and were required to specify their reasons for selecting the documents in their bibliographies. The results indicate that many different facets of judgment are used to determine a document's usefulness. Besides the obvious facet of topic, Cool *et al* identified features such as document content, format, age, authority and style. One significant conclusion of this work is that searchers have a relatively small set of 'dimensions' in mind when determining a document's usefulness. These dimensions provide a possible way to organize the documents within an information space. We propose, along with Cool *et al,* that a strong improvement in the effectiveness of document retrieval can be accomplished by implementing user interfaces for information search that allow users to specify their information needs in terms of the dimensions that people naturally use to determine document usefulness.

The AIM method

We created AIM to demonstrate the feasibility of expanding information retrieval systems to consider a wide range of characteristics when matching documents to queries. To match documents and queries together, a common representation must be established (Figure 1). In keyword retrieval systems, this common representation is a list of words. A document is represented by the list of words that constitute the document, possibly including position and frequency information. Queries consist of a list of user-supplied words, possibly including operators and weights. Documents and queries are matched together by comparing the document and query word lists. AIM expands this approach by extending the representations of documents and queries to include more meaningful descriptions. Extended representations offer the possibility for users to express their information needs more completely, increasing the quality of the dialog with an information retrieval system.

The AIM representation for both users' information needs and documents consists of sets of descriptors. Users, through an appropriate user interface, express their information needs as sets of descriptors. The computer, likewise, represents documents with the same set of descriptors. Retrieval then consists of matching descriptors. This view of retrieval is a simple extension to existing keyword systems.

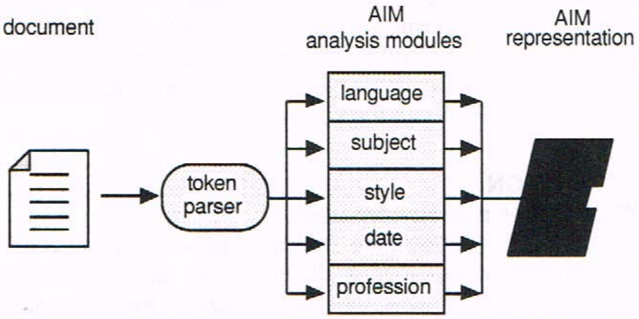

Figure 2: AIM generates a representation of a document by parsing it into tokens and processing the token stream with a set of analysis modules. The resulting set of AIM descriptors constitute the representation.

To generate descriptors for a document, AIM takes the plain text of the document as input, parses it into words, and feeds the words to a set of analyzers which generate AIM descriptors (see Figure 2). These descriptors are stored as a representation of the document. In our discussions, we will refer to any unit of text as a 'document'. This document could represent a single entry in a bibliographic database, an e-mail message, a complete journal article, the full-text of a book, or even a complete database of many articles. AIM analyzes whatever text is given to it and generates a concise (~100 byte) set of descriptors which represent the text. The AIM descriptor of a 2 MB collection of journal abstracts is still 100 bytes and contains all of the AIM metadata for the average qualities of the collection. This conciseness allows many different document representations to be stored and examined efficiently using AIM descriptors.

The definition of a 'document' depends upon how AIM is being used. For example, to enhance *ad hoc* retrieval of world-wide web pages, each page might be considered as a document. Retrieval would then depend on AIM descriptors in addition to keyword matching. Another use of AIM might be to assist a user in selecting an appropriate text database for a query. In that case, the entire database (or suitable subset of it) could be used as an AIM document. The user's information need can be compared with the AIM descriptors for each database to determine high-quality targets for the full query. These and other scenarios will be described in a later section.

We constrained our choice of methods for generating document metadata by requiring that each be computed automatically (*i.e.*, with no human intervention) and have low-computational overhead (*i.e.* only linear-time algorithms were considered). Given the observations of Cool *et al* and these constraints, our first implementation of AIM has a representation consisting of five basic descriptors: language, subject, writing style, date, and profession. These descriptors are both useful to information searchers for describing their information needs, and can be automatically derived by computers from natural language text. AIM is extendible in the sense that additional descriptors can be defined as new analysis modules are developed and as different communities seek to specialize the descriptors for their unique needs.

Why Keywords are Not Enough

The most common technique for information retrieval is keyword matching. Keywords can be either manually-derived words which describe the text, or simply all of the words contained within the document. Most functions that map documents to queries fit within two categories: (1) boolean operations acting on the presence/absence of keywords [12] and (2) weighting functions which generate numerical quality-of-match scores between documents and queries depending on keyword usage statistics [10]. The queries are often expanded through a variety of extensions. Proximity operators allow the specification of how close together various keywords must be, therefore allowing the specification of multi-word phrases as search terms. Wildcard operators allow the specification of word-forms which will match a number of keywords. Morphological analysis allows the matching of variant forms of the same word. And thesauri are used to expand queries based on word relationship data such as synonyms.

Keyword matching is a good starting-point for information retrieval systems, after all, the words of our languages are the atoms of communication. However, a major limitation in keyword searching is that the search language essentially allows only a single type of request: "Find me documents containing significant occurrences of the following words". Users have a difficult time producing effective queries with this form of request. We mention two specific problems.

The first problem with keyword queries is that single words have multiple meanings. For example, consider a query consisting of the word "mouse". Does this mean a computer pointing device or a small rodent? In normal language use, the context of the word aids the listener in disambiguating the term. However, most query systems do not have a good mechanism for communicating context. The searcher can use tricks, such as specifying collections of additional words which describe the context of the primary word. In this example, a boolean query such as "mouse AND (computer OR cursor OR pointing OR pointer)" might be used. But this query is much more difficult to specify and may not be effective if the context-terms are not chosen carefully.

In free-text searching systems, the expanded query might be: "tell me about a mouse for pointing a cursor on a computer system". In such search systems, however, it is difficult for the computer to determine that the main concept of the query is "mouse" and that the rest of the terms are only setting the context. Therefore documents which concern computers and cursors will score just as highly as those which concern mice and computers. Thus, ambiguity is not easily overcome by the limited expressiveness of conventional keyword searching.

A second problem with keyword queries is in specifying the genre of the sought documents. For example, consider a search for technical papers on the operation of speech recognition systems. A simple query such as "speech recognition" will collect product announcements, popular descriptions, and business projections, in addition to the desired technical papers. Within the constraints of keyword matching, how does the user indicate a preference for technical papers? The query language is just not expressive enough to allow this type of query. Therefore, the usual solution is to keep different kinds of documents in different databases, leaving the user with the difficult responsibility of choosing the appropriate database. However, there is a trend toward making larger sets of information available from a single query, eliminating the distinction between different databases (*e.g.*, the GlOSS server index [6]). This trend solves the problem of choosing an appropriate database, but also eliminates this important dimension of information need expression. Thus, as databases become larger and more diverse, it seems that the query language should be improved so that the desired documents can be found.

Therefore, we observe that keyword-based retrieval systems have limitations in their range of expressiveness. Users have the capability to describe their information need more completely than the search language allows. We present AIM as an approach for expanding the user's ability to express an information need beyond keywords.

AIM DESCRIPTORS

In AIM, both documents and queries are represented by sets of descriptors (see Figure 1). These descriptors extend the expressiveness of both users' information needs and documents' contents beyond the keyword model. In this section, we consider each of the AIM descriptors that we have developed. We describe the contents of the descriptors, the semantic meaning of the descriptors, and examples of their usefulness for describing users' information needs. These represent our first implementation of a set of AIM descriptors, not necessarily the absolute best set. We expect this set to grow and mature as we better understand users' information needs and the possibilities of automatic document analysis.

Language

An obvious characterization of a document is the language that it is written in. This aspect might be easily overlooked because the overwhelming majority of electronic information is written in English. However, for users with other preferred languages, this could be the most important factor in determining the usefulness of a document.

AIM determines the language of a document by examining character and word frequency statistics. Its language descriptor reports both a language class and specific language for the document. The language class includes 'human', 'computer', and 'machine-readable'. A human language is one that is normally used for human-to-human communication, such as English, French, or Spanish. A computer language is one that is normally used for human-to-computer communication, such as Lisp, C++, or FORTRAN. Machine-readable languages are ones that are only directly usable by computers, such as binary, hexadecimal, or uuencode. Each language is described by the statistics of its most common words (*i.e.,* stopwords) and by the usage statistics of each of the standard ASCII characters. More complex analyses could be used, as in the RUFUS classifier [11].

The language descriptor can be used by the other AIM analysis modules to guide the processing of the text. For example, the date descriptor (discussed later) can use the language descriptor to direct the parsing of date structures as they are usually written in the language of the main text. Similarly, the language descriptor can be used to gate certain language-specific analysis modules. The subject analyzer is currently English-only and is not executed when a document is in another language.

Subject

The subject descriptor is currently the most complex in the AIM system. The goal of this descriptor is to determine the main topic of the document. The AIM subject descriptor classifies a document within a pre-defined subject classification system (SCS). The calculation of a subject descriptor is similar to automatic indexing [9]. Of course, it is unrealistic to suppose that a single AIM SCS will be adequate for all fields, languages, and document types. Documents can easily be classified under multiple SCSs and then matched to user queries using one or more of them. Alternatively, AIM could be guided to analyze subject using a human-chosen SCS.

In our implementation of subject analysis, we assume that the subject can be determined by the vocabulary statistics of the document, and we assume that a certain pattern of word usage will characterize each subject. For example, documents about 'bowling' would include words such as 'strike', 'spare', 'pin', 'bowling', 'ball', and 'alley' with relatively high frequencies, while almost never using the word 'glacier'. AIM is trained with the vocabulary statistics for each subject in the SCS. Once AIM is taught the vocabulary statistics a document can be analyzed by comparing its vocabulary statistics with that of each subject. This comparison can be done using one of many similarity functions [9]. The result is a ranked list of subject matches with the top *N* ranked subjects defining the AIM subject descriptor for the document. Including more than just the one top-ranked subject in the descriptor has two advantages: (1) if a document does not fit neatly into one subject classification, the top several describe the real subject matter of the document more completely. And, (2) if the SCS contains several very similar subjects, slight variations in documents can cause the top-ranked subject to change. Therefore, the robustness of the AIM descriptor is improved if several top-ranked subjects are maintained so that the overall AIM descriptor is not strongly sensitive to small variations in documents.

As mentioned, AIM must be trained with the vocabulary statistics for each subject in the SCS. If this were a manual process it would be terribly laborious. But AIM can be trained on any set of text which has already been classified. AIM simply gathers the vocabulary statistics for each document, notes the subject classification(s) which have been given to the text, and generates a set of statistics for the entire collection. Sources of subject-classified text are plentiful. We have tested it with two different SCSs, the Library of Congress scheme and the USENET scheme.

We trained AIM on the Library of Congress SCS, by giving it the titles of all of the books in our library along with their Library of Congress call number. We designated the first one or two letters of the call number as an AIM subject. Thus the 'QA' subject concerns mathematics and computer science and 'TK' concerns electronics technology. The vocabulary statistics from all of the books in a given subject classification then defined that AIM subject. We observed that this method was able to reasonably classify articles on a wide variety of topics. Nevertheless, we discovered limitations in the way we had implemented the Library of Congress AIM subject descriptor. The first two letters of the call number are often not fine-grained enough to make the system usable for specialists. As noted above, the 'QA' subject contains all of mathematics and computer science. If we used this to subject-classify the Internet, a very large fraction of the documents would by classified into this one subject.

So we turned to the USENET as another set of training data for AIM. We designated each of approximately 1500 USENET newsgroups as an AIM subject. We took the contents of our USENET news server as the training text and gathered the vocabulary statistics for all of the postings contained in each newsgroup. This compilation resulted in 1500 subject classifications which covered a broad range of popular technical subjects. Although the USENET newsgroups are arranged hierarchically, we flattened the structure so that each newsgroup was considered as a separate subject, regardless of its place in the USENET hierarchy.

The USENET SCS has proven to be highly effective at classifying documents from a wide variety of sources, including technical reports, newspaper articles, computer documentation, and corporate speech texts. We show two representative examples of the results of AIM's subject analysis in Table 1. AIM analyzed two different documents using the USENET SCS. The table shows the list of USENET subjects which were assigned to the text in order of decreasing similarity along with their matching score and the dominant words which contributed to the score. The absolute magnitudes of the scores are not significant, resulting as much from the breadth of vocabulary used in the subject as the closeness of match. The order of the subjects in the ranked list contain the significant information, not the values of the scores themselves.

In the first example, the 'document' consisted of a collection of 2 MB of journal abstracts on the physics of cold nuclear fusion. The list of subjects starts with sci.physics.fusion, the obvious best choice, followed by the subject of electrochemistry which is used in the cold fusion experiments. then by the subject of 'science used in science fiction' which also covers fusion power. Subjectively, these all seem to be good matches.

SUBJECT ANALYSIS

COLD FUSION TECHNICAL PAPERS

score	subject name	dominant words
3766	sci.physics.fusion	fusion deuterium palladium tritium neutron
906	sci.chem.electrochem	electrode ion cathode concentration nickel
890	rec.arts.sf.science	fusion fission ion nuclear plasma hydrogen
734	sci.research	fusion AIP energy atoms isotopes physics
672	sci.tech.spectroscopy	spectra abstract emission cathode

BATTERY DISCUSSION

score	subject name	dominant words
1963	rec.radio.amateur.equip	alkalines nicads batteries rechargable nicd
1146	sci.chem.electrochem	batteries nicads alkaline rechargeable
858	sci.energy.hydrogen	renewable recharge batteries charging
782	rec.radio.shortwave	rechargable batteries nicads alkalines
464	comp.sys.handhelds	batteries alkaline charger recharge poke

Table 1: Results of subject analysis on two collections

The second example is from the analysis of a collection of articles from an IBM discussion forum on battery technology. This test was more difficult for AIM because there is no USENET subject on batteries. Therefore, AIM matched it to various USENET subjects which discuss battery technology. AIM's subject classification for this document included different subject areas which use batteries, including amateur radio, handheld and palmtop computers, in addition to electrochemistry, which covers battery physics. These two examples are shown for illustration, we have analyzed many such documents using AIM with similarly reasonable results.

We have also run quantitative tests on the AIM subject descriptor using the USENET SCS on a subset of the TREC document and query collection [7]. The TREC collection contains approximately 3 GB of text and 100 queries. For each query, a list of documents has been judged by the TREC committee to be relevant for the query. We let AIM analyze each of 11,840 documents from the TREC collection for subject. Then the 'concept' section of each of the 100 TREC queries were likewise analyzed for subject. We then matched the documents against the queries according to how well their AIM subjects overlapped. No direct comparison was done between queries and documents, only comparisons of their AIM-generated metadata. Figure 3 shows the results of this measurement. As shown, a large fraction of the documents judged relevant for a query were found simply by matching AIM subjects between documents and queries. For example, over 75% of the relevant documents are found by querying only the top 2% of the documents recommended by the AIM subjects descriptors. These results demonstrate that AIM can successfully determine the subject matter of both documents and queries so that they can be matched.

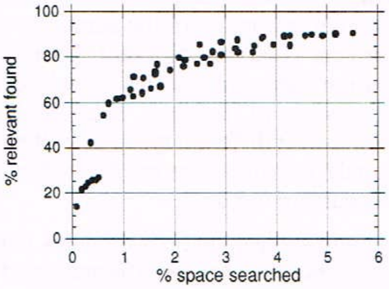

Figure 3: Quantitative measurement of subject analysis using USENET training text and the TREC evaluation collection.

While the TREC collection is designed to be a realistic measure of retrieval effectiveness, the AIM descriptors must still be tested in a real user community. AIM also demonstrated good robustness in the TREC test because the training data and testing data were so different. It was trained on the USENET text and then tested on articles from the *Associated Press*. A practical use for this type of analysis will be discussed in the Distributed/Agent Retrieval scenario below.

Writing style

The AIM writing style descriptor contains several sentence-oriented measures of the type of writing used in a document. These measures correlate with the format of the document, its purpose, and its target audience. These measures are all gathered together into a single descriptor because of their close relationship. To make these measures, AIM parses the document into sentences. Sentence boundaries are determined by a word which terminates in a period, question mark, or exclamation point, followed by white space. A valid sentence is defined as a sentence which contains at least three words and does not contain any words which have punctuation within them besides a hyphen, slash, or apostrophe.

The first measure AIM reports is the fraction of words in the document which are contained in valid sentences. This measure describes the format of the document. A number near 100% means that the document is almost pure text, like a letter, a speech, or a book chapter. A number closer to 50% means that the document is text + description, like e-mail messages (body with a header that is not in sentence form) or a journal abstract (body with formatted fields such as author and publication date). A number closer to 0% means that the document is not natural language at all, possibly in tabular form or ASCII text images. This measure allows a user to specify an information need which requires an almost fully textual document, such as a newspaper article.

Next, the AIM writing style descriptor reports the fraction of valid sentences which end in a question mark. This measure can differentiate between expository documents and discussion documents. Expository documents have almost no sentences phrased as questions. Discussion documents such as newsgroups have a consistent 6% to 9% of their sentences as questions. As an extreme case, a questionnaire may have nearly 100% questions.

In many cases, information searchers know the format of the document they want in response to a query. For example, if we are looking for a way to work around a limitation in a popular word processor, we are most likely to find useful information in a discussion group. By specifying that documents should only be reported from sources which have at least 3% questions, we can greatly increase the likelihood of getting discussion-oriented documents.

The final writing style descriptor that AIM returns is the reading level of the text. AIM uses the Kincaid score to calculate the reading level in units of the American school system. The formula is straightforward, depending only on the average number of syllables per word and the average number of words per sentence in the valid sentences of the text. It is given by:

$$\begin{aligned}Kincaid = 11.8 \times (syllables\_per\_word) + \\ 0.39 \times (words\_per\_sentence) - 15.59\end{aligned}$$

The Kincaid score is useful for differentiating the purpose and target audience for a document. Scholarly articles tend to be written at a Kincaid level of 11-14. Professionally-written news is in the range 9-11. Informal writing such as e-mail and discussion groups are typically in the range 6-9. This measure can then be used to gather only documents which are written for a particular purpose. For example, newsgroups and journal articles may both contain information about the same subject and use the same keywords but they will have different Kincaid scores. Because information seekers generally know whether the information they seek will be written in an informal or scholarly forum, this writing style descriptor can be used to express that information need more precisely.

Table 2 shows the AIM writing style descriptors for the same two documents used to generate Table 1. We see that the journal articles in the cold fusion collection contain a moderate fraction of words in valid sentences (indicating header+abstract form), has almost no questions (expository text), and is written at a high reading level (scholarly). Likewise, the battery discussion group also has a moderate fraction of its words in sentences (header+body form), has a few questions (indicating a discussion group), and is at a low reading level (informal text). These simple descriptors reveal information complementary to the simple list of words normally used for retrieval.

WRITING STYLE ANALYSIS

measure	COLD FUSION TECHNICAL PAPERS	BATTERY DISCUSSION
words in sentences	68.7 %	72.5 %
questions	0.6 %	7.8 %
exclamations	0.0 %	1.3 %
reading level	14.0 grade	7.1 grade

Table 2: Results of writing style analysis on two collections

Date

The AIM date descriptor is a set of statistics derived from any date structures that can be found in the document. AIM understands several common date structures such as Aug 12, 1995 and 8/12/95. These structures can be ambiguous so AIM does make mistakes but the statistics are largely correct. The AIM date descriptor contains the average, standard deviation, minimum, and maximum dates from the list it gathers from a document. It also reports a histogram showing the frequency of occurrence of dates in different ranges.

DATE ANALYSIS

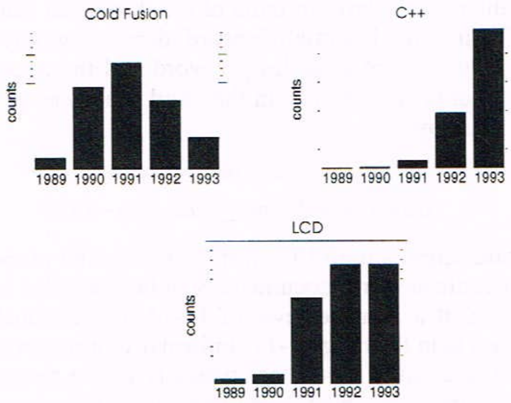

Figure 4: Frequencies of date occurrence in three different subject areas

These date measures are most useful for collections of documents. For example, in a collection of bibliographic documents or discussion group postings, the dates indicate the timeliness of the articles. Discussion group archives are usually divided into several different files by date. AIM analysis discovers which date ranges are covered by which files. In addition, the distribution of dates in the collection can reveal trends of popularity for the collection. Figure 4 shows the histograms for several different subject areas within the INSPEC database of journal abstracts. As shown, the cold fusion subject had a boom in 1991 which quickly disappeared after its initial excitement. In contrast, articles on C++ have been exponentially increasing in frequency over this same range. Articles on liquid crystal display technology increased rapidly over 1990-1991 and has now saturated as the research has matured.

Because information searchers often know the timeliness of the information they seek, this descriptor helps describe their information need more fully. In addition, the shape of the date histogram for a collection can be used to identify exciting new areas which may be brought to a user's attention. This sort of analysis broadens the range of description that an information retrieval system can offer to its users.

Profession

The last AIM descriptor is 'profession'. This descriptor characterizes the common vocabulary in a document, which typifies the writer of that document. For example, an article about microprocessors may be written from the business profession point-of-view or the technical profession point-of-view. Although articles from these two professions may contain many of the same keywords, their information content is largely different. Users would have trouble generating keyword queries which can separate these different types of articles. The AIM profession descriptor assumes that different types of writers have their own core vocabulary. Professions are defined by collections of frequently-occurring words.

The profession and subject analysis modules both examine vocabulary usage, but they target different word frequency ranges. AIM subject analysis focuses on rare words, profession analysis on higher frequency words. The subject analysis emphasizes distinctive words which indicate one particular specialty or another. Profession analysis emphasizes the frequently-occurring words, which are used in many different subjects, but which occur more commonly in one profession than in another. Table 3 shows the partial results for two different collections. The newspaper profession shows some distinctive 'news reporter' profession words such as 'said', 'police', and 'people'. The electrical engineering profession shows distinctive core vocabulary words such as 'system', 'processing', and 'data'.

PROFESSION ANALYSIS

NEWSPAPER	ELEC. ENGINEER
said	processing
he	data
police	signal
people	based
they	model
president	method

Table 3: Examples of frequently-occurring words from two different data collections

SCENARIOS

We have described the AIM descriptors that we have developed to date. These descriptors can be used to automatically characterize documents along several dimensions. They can also be used by information searchers to express their information needs in a much richer fashion than can be accomplished by keywords alone. AIM provides a more complete bridge between the user's information need and the data within a computer. In this section, we examine different user scenarios which illustrate the ability of information retrieval systems to improve their usefulness by incorporating AIM. In particular, we look at three different cases: *ad hoc* retrieval, distributed agent-based retrieval, and selective dissemination of information. In *ad hoc* retrieval, a user has an information need and relies on an information retrieval system to address that need. In distributed agent-based retrieval, the user's information need is dispensed to a system of 'agents' which act on the user's behalf to cull the information from a distributed set of information resources. In selective dissemination applications, users describe continuing information needs in profiles which are then run periodically against dynamic data sources to generate spontaneous information alerts when something is found which matches a profile. Each of these cases has unique features which can exploit an information metadata system such as AIM.

Ad Hoc Retrieval

In *ad hoc* retrieval situations, a user wishes to quickly describe an information need, get results, browse through them, and find an answer. One problem in ad hoc retrieval is low precision (*i.e.,* too many useless results), especially with untrained users. For example, in a report on the World Wide Web Worm [8], a retrieval tool for finding WWW pages by content, the developers suggest that a major problem with the system is imprecise search results. The main culprit is the brevity of user queries, which average only 1.5 search terms. Apparently users are unwilling, unable, or unaware of the need to describe their information needs in detail. By providing a more comprehensive description language, AIM provides the searcher the ability to interact more naturally with the available resources.

We describe two ways AIM can be used to improve *ad hoc* retrieval. First, AIM can improve the user's ability to describe the information need. AIM's subject descriptor can provide a "two-step query" model. In the first step, the user describes the general information goal by a free-text description of the subject matter. This subject description is converted into a set of AIM subjects and then only information resources which fit those subjects need to be reported back to the user. This two-step model decreases the number of useless results. It is like choosing among libraries on a college campus, or choosing which catalog to browse when looking for a product. Keyword-based systems can often be like having your kitchenware catalog combined with your electronic test equipment catalog along with every other catalog in the world. Normally, people are accustomed to describing their needs in more general terms before diving into the details. AIM affords users this natural approach to information retrieval.

As a second example, AIM can be used to improve the format of the search results. A problem with standard information retrieval systems is that queries result in long, complex, text-only lists. Any means to organize or order this list is helpful. Today's systems take users from the root of a tree (the database) to the leaves (the documents) with no structure in between. AIM enables a user interface to organize the result list so that higher-level information is available rather than just the titles of all matching documents. For example, documents can be arranged according to AIM subject, so that similar articles appear together. Likewise, results can be organized by writing style, or by any other AIM dimension to help orient the user in the morass of returned information.

Distributed/Agent Retrieval

A number of distributed agent-based search architectures have been proposed [2]. In these systems autonomous agents gather information and make it available to users and other agents. Information brokers do not deal with information directly, but know how to route information seekers to information possessors by keeping track of 'who knows what'. One of the missing links in such systems is the language that is used to describe information. In order to keep track of 'who knows what', we must know how to describe the information that that agent knows. Possible solutions are to describe the information in terms of lists of keywords [5] or manual subject classifications [4]. AIM provides a clear method of describing information in terms of automatically derivable descriptors.

Our tests of the AIM subject descriptor with the TREC collection demonstrate that we can effectively organize documents and queries by subject. It is straightforward to apply AIM to a community of agents. For example, each agent could specialize in documents which are concerned with a single subject. Agents could then route documents and queries to the appropriate agents to match them together. Such an agent-based system could provide order to a disorganized collection of information to allow the efficient retrieval of useful documents. In essence, AIM provides the language by which agents communicate with one another as they go about their task of collecting information for their users.

Selective Dissemination

In selective dissemination applications, users provide information profiles which are then run periodically against dynamic databases to gather new documents which match the profiles. Experience has shown that these systems often do not meet users' expectations. Either the information has too many results judged useless by the user to be worth sifting through, the type of the material is wrong, or the profiles are out-of-date. Many users simply delete the alert messages that come from their selective dissemination service each day, not wishing to figure out how to turn the service off.

AIM provides the possibility of greatly improving such systems by allowing users to describe their information needs with the increased expressiveness of AIM. Beyond the initial need description, AIM enables users to maintain their profiles more effectively. Users can indicate whether certain retrieved articles are useful or not so that the system can update the profiles automatically. Current feedback systems can only observe patterns in word usage to generate refined profiles. But with AIM analysis, all of the AIM descriptors can be used to observe patterns in user judgments.

In summary, the expanded expressiveness of AIM descriptors affords the possibility of improving many aspects of information retrieval systems. Users can more effectively communicate their needs and converse with information repositories through the more complete bridge that AIM builds between human and computer.

CONCLUSIONS

AIM is a first-step toward establishing a multidimensional conversation as the basis for information retrieval. Because keyword-matching alone limits the ability of users to express their information needs, we expand the range of communication to include the aspects of language, subject, writing style, date, and profession information. To enable automated information retrieval based on these descriptors we present mechanisms for automatically analyzing text to derive them. In this way, both documents and queries are represented by the same set of AIM descriptors. AIM can then match together the descriptors for the queries and for the documents. This approach of expanded expressiveness offers the possibility for supporting users in a more complete conversation as computers work to address their information needs.

ACKNOWLEDGMENTS

The authors would like to thank Thomas Kirsche for the TREC measurements, and both John Bennett and Paul Maglio for helpful discussions.

REFERENCES

1. Belkin, N.J. and Croft, W.B. Retrieval Techniques, in Williams, M. (Ed.), Annual Review of Inf. Science and Tech., Elsevier, New York, 1987, pp. 109-45.

2. Bowman, M.C. Danzig, P.B., Manber, U. Schwartz, M.F. Scalable Internet Resource Discovery, Comm. ACM, 37(8), 1994, pp. 98-107.

3. Cool, C. Belkin, N. Kantor, P. Characteristics of Texts Affecting Relevance Judgments, in Proc. of the Fourteenth Natl Online Mtg (New York, May 4-6, 1993), Learned Inf., pp. 77-84.

4. Danzig, P.B. Li, S.-H., Obraczka, K. Distributed Indexing of Autonomous Internet Services, Computing Systems, 5(4), 1992.

5. Duda, A. Sheldon, M.A. Content Routing in a Network of WAIS Servers, in Proc. 14th Int'l Conf on Distributed Comp Sys, 1994, pp. 124-32.

6. Gravano, L. Garcia-Molina, H. Tomasic, A. The Effectiveness of GlOSS for the Text-Database Discovery Problem, in Proc. 1994 ACM SIGMOD Conference (May 1994).

7. Harman, D.K. (Ed.) The Third Text Retrieval Conference (TREC-3) NIST Gaithersburg MD (1995).

8. McBryan, O.A., GENVL and WWW: Tools for Taming the Web, in Proc. 1st Int'l World Wide Web Conf, (Geneva, May 1994).

9. Salton, G. and McGill, M.J. Introduction to Modern Information Retrieval, McGraw-Hill, New York, 1983.

10. Salton, G. and Buckley, C. Term-Weighting Approaches in Automatic Text Retrieval, in Information Processing and Mgmt, 24(5), 1988, pp. 513-23.

11. Tresch, M. Palmer, N. Luniewski, A. Type Classification of Semi-Structured Documents, in Proc. 21st Int'l Conf on Data Engineering (Zurich, Sept 11-15, 1995).

12. Wartik, S. Boolean Operations, in Frakes, W.B. (Ed.), Information Retrieval: Data Structures and Algorithms, Prentice Hall, New Jersey, 1992, pp. 264-292.