

NAVIGATE Here

Search

- :: [Next article](#)
- :: [Previous article](#)
- :: [Table of Contents](#)

RESOURCE Links

- :: [Similar in PUBMED](#)
- :: [Search Pubmed for](#)

- [Nadkarni P M](#)

:: [Search in Google Scholar](#)
for

- [Nadkarni P M](#)

::Related articles

- [Databases](#)
- [Bibliographic](#)
- [Information Storage and Retrieval](#)
- [Support](#)
- [U.S. Gov't](#)
- [P.H.S.](#)
-

:: [Article in PDF](#) (145 KB)

:: [Citation Manager](#)

:: [Access Statistics](#)

:: [Reader Comments](#)

:: [Email Alert](#) *

:: [Add to My List](#) *

* Registration required (free)

IN THIS Article

- :: [Introduction and...](#)
- :: [Document indexing](#)
- :: [Types of query m...](#)
- :: [Automated concep...](#)
- :: [Use of vocabular...](#)
- :: [Term recognition...](#)
- :: [Practical limita...](#)
- :: [Ir and tradition...](#)
- :: [Other areas of i...](#)
- :: [Conclusions](#)
- :: [Grant support](#)
- :: [References](#)

Article Access Statistics

Viewed	12406
Printed	347
Emailed	13
PDF Downloaded	210
Comments	[Add]
Cited by others	2

...s, PubMed and Science Citation Index Expanded



E-MEDICINE

Year : 2000 | Volume : 46 | Issue : 2 | Page : 116-22

Information retrieval in medicine: overview and applications.

[PM Nadkarni](#)

Centre for Medical Informatics, Yale University School of Medicine, New Haven, Connecticut 06520-8009, USA. , USA

Correspondence Address:

P M Nadkarni

Centre for Medical Informatics, Yale University School of Medicine, New Haven, Connecticut 06520-8009, USA.

USA

Login to access the email ID

Source of Support: None, Conflict of Interest: None



PMID: 0011013482



Keywords: Databases, Bibliographic, Information Storage and Retrieval, Support, U.S. Gov't, P.H.S.,

How to cite this article:

Nadkarni P M. Information retrieval in medicine: overview and applications. J Postgrad Med 2000;46:116

How to cite this URL:

Nadkarni P M. Information retrieval in medicine: overview and applications. J Postgrad Med [serial online] 2000 [cited 2020 Dec 17];46:116 from: <https://www.jpgmonline.com/text.asp?2000/46/2/116/298>

:: Introduction and history

Information Retrieval (IR) is a branch of computer science that is concerned with the processing of collections of documents containing "free" of hospital discharge summaries, radiology reports, or surgery notes, or (in a non-medical context) the full text of the complete works of Shakespeare or a database table, which is divided into rows and columns, such documents have no obvious structure: any structure that is imposed is of little use in retrieving information. Thus, while the Bible is divided into books, chapters and, occasionally, verses, a reader is more interested in finding a particular verse (e.g., the name of particular Biblical characters or a quote). Similarly, the "structure" of a Chest X-Ray report, in terms of the headings, is different from the structure of a Barium swallow or Intravenous Pyelogram report.

IR began as an offshoot of "information theory," a field defined in a classic paper by Claude Shannon of Bell Laboratories in 1949. (Shannon's paper was written for a general audience.) However, "information" was defined in a very broad sense. Some of the work in this field considered practical problems such as how to add redundant (extra) information so that data transmission or storage would be reliable despite the presence of errors in transmission, (e.g., Reed and Solomon[2]). The focus on textual information can be traced to several researchers, most notably the late Gerard Salton, the definitive textbook for IR[4]. Salton's group has performed research (using a system called SMART[5]) that has defined the methodology of IR. The textual component of the clinical patient record, IR techniques have found ready applicability in medicine: a good reference for medical IR is

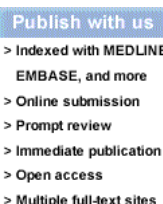
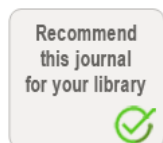
For several decades, IR was a relatively "orphan" technology researched by a relative handful of scientists, and commercial offerings were rare. However, due to the spread of the World Wide Web, IR is now mainstreaming because most of the information on the Web is textual. Web search engines like Google and AskJeeves are used by millions of users to locate information on Web pages across the world on any topic. The use of search engines has given access to the Internet, the World Wide Web has replaced the library as the reference tool of first choice. Search engines can also be purchased by hospitals and search facilities for their own Web sites. For example, Microsoft bundles a component called Microsoft Index Server with its Web server software. MEDLINE access as implemented through PubMed. (A subsequent article in this series will discuss MEDLINE article retrieval from the use of principles underlying IR technology.)

SPONSORED SEARCHES

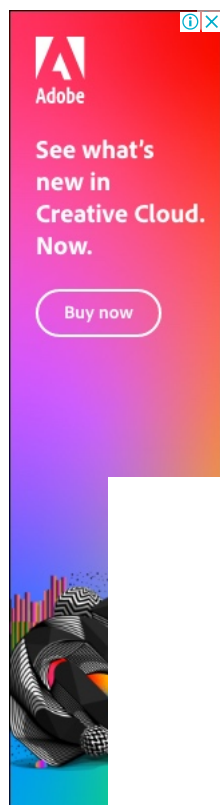
:: Document indexing

Online Document Retrieval

Software Suite



Click on image for details.



Search engines rely on pre-processing a document collection for subsequent fast retrieval based on keywords. This pre-processing step is termed by the words contained in them are described below. Some of the operations involved are also employed in concept-identification algorithms:

* Every document in a collection is scanned word-by-word, skipping words belonging to a stop-word list. *Stop-words* are very common words for searching (e.g., "the", "an", "of", etc). It is important to note that, in specific contexts, words uncommon in the language as a whole may be notes, "surgery" would be a stop word. Conversely, when processing medical text, certain common words should not be treated as stop word concepts. Thus, "greater" and "lesser" specify a femoral trochanter.

* One or more disk-based structures (*indexes*) are created. The *global word-frequency index* records, for each distinct word, how many times *word-frequency index* records how often a particular word occurs in each document. An optional *proximity index* records the position of individual sentence, or paragraph offsets. (A proximity index takes up lots of space, but is useful for searches where one can specify two or more words in a sentence, or next to each other, or within so many words of each other).

* The index stores words either as they occur in the text, or after transformation. One type of transformation is *normalization*, which involves variations in person or tense. Thus, "children" is normalized to "child," and "brought" is normalized to "bring." Another transformation is *stemming* suffix removal or letter substitutions to yield a "root" form. Stemming is more drastic than normalization; often yielding a root that is not a valid word. "hypothesize" and "hypothetical" yield the same root, "hypothes." Stemming uses simple linguistic rules (e.g., removing suffixes like "-ing" on synonyms (e.g., liver, hepatic) is not used in stemming. The most widely used technique for stemming is the Porter algorithm[7]. Both normalization and stemming, but at the cost of specificity. (In the context of IR, *sensitivity*, also called *recall*, is defined as the ratio of the number of relevant documents in a query to the number of relevant documents *actually present* in that collection. *Specificity*, also called *precision* in the context of IR, is defined as the ratio of documents retrieved by a query to the *total* number of documents retrieved: some retrieved documents may turn out to be irrelevant to the query. In tests in diagnostic medicine, the higher the sensitivity, the less the *false negative* rate: similarly, specificity is negatively correlated with the *false positive* rate. It is known to have problems in stemming words with Greco-Latin suffixes (-um, -ae, -ii, -us, etc.), which often occur in medicine. The C language described in [8] is driven by rules that are stored in tables, however, and such suffixes can be managed by adding more rules.

* For a large document collection, it is wasteful of disk space to store the words themselves in the index. Instead, a table is created to store each word and each word is assigned a numeric ID. This ID is used in the index.

* Every document in the collection is also given an ID. In the medical field, each document applies to a particular patient, and so one maintains a patient index. This information is used as a bridge between the free text and other non-free-text information (e.g., laboratory test values) on that patient, with the patient record system.

:: Types of query methods

There are two methods of searching documents using such indexes. The conceptually simpler (and older) *Boolean* method (named after the mathematician George Boole, who devised a mathematical approach to logic) allows the user to specify keywords combined with the operators AND, OR and NOT (the subsequent article on MEDLINE.) Boolean methods have been (falsely) criticized because Boolean logic is hard for end-users to understand. One criticism of a purely Boolean approach is that the returned documents are not ranked in any way: if numerous documents are retrieved, the user has to sort through them to find the most relevant.

Modern IR search engines (e.g., Web-based engines), which may retrieve hundreds of matches for a user's query, therefore perform *automatic relevance ranking* to handle situations where no document matches every single keyword specified by the user. (In the case of Boolean search, zero documents would be returned.)

Relevance ranking is typically done through the *Document Vector approach*, originally devised by Salton's group. Several variants of this approach exist. The most common is the *cosine* metric, which is computed between the terms in a query, and a document that must be screened to see how closely it matches the query, is called the "*normalized cosine*" metric. The principle is that keywords in the query that are uncommon in the document collection are given more weight than common terms. Thus, if we take a query with the keywords "carotid endarterectomy complications" the first two words would be less common, and therefore given more weight. Similarly, documents that contained the words "carotid" and "endarterectomy" many times would be likely to have these words as a theme or topic more relevant to the query than documents that mentioned the phrase in passing only once.

The word "normalized" in the metric refers to the fact that the frequency with which a particular term occurs in a document is adjusted for the total number of words in the document. For example, the word "carotid" three times would be more relevant than a 10,000-word article that mentioned it four times. The *normalized cosine* metric is computed between the terms in a query, and a document that must be screened to see how closely it matches the query, is called the "*normalized cosine*" metric. The principle is that keywords in the query that are uncommon in the document collection are given more weight than common terms. Thus, if we take a query with the keywords "carotid endarterectomy complications" the first two words would be less common, and therefore given more weight. Similarly, documents that contained the words "carotid" and "endarterectomy" many times would be likely to have these words as a theme or topic more relevant to the query than documents that mentioned the phrase in passing only once.

Modern search engines use a combination of Boolean and vector methods, with the former being used as a filter to restrict the search. Most search engines use the *Boolean* method to mandate that all retrieved documents must contain that word. In other words, + implies the AND operator. - implies that the retrieved documents should NOT contain that word. (PubMed uses a slightly different syntax, as a subsequent article will discuss.) Boolean criteria are then ranked by relevance.

:: Automated concept indexing

Thus far, we have discussed electronic indexing of documents based on the *words* within them. For documents belonging to a single domain capitalize on domain knowledge. For example, synonymous phrases are not automatically recognized. Searches of medical free text that is manually specify synonymous forms, or else risk missing relevant documents.

Using concepts in a domain-specific vocabulary (thesaurus) can enhance retrieval: one can index the *concepts* identified in a document. Indexing by word. Text is processed to recognize potential terms, which are then looked up in a vocabulary. The ID of the concept in the vocabulary discussed in a subsequent article, the US National Library of Medicine has historically used manual concept recognition using trained humans before assigning concepts to it. Here, we discuss *automated* approaches that have attempted to use computer programs for the same task.

:: Use of vocabularies in automated concept indexing

In medicine, concept-indexing research has used a variety of vocabularies, depending on the material that must be indexed. The biggest thesaurus of Medicine's Unified Medical Language System (UMLS) Metathesaurus[11]. This incorporates numerous other vocabularies, such as SNO (used for pathological classification of disease), ICD-10 (Internal Classification of diseases, typically used for billing code), DSM-IV (Diagnostic Disorders, 4th Edition), as well as MeSH (Medical Subject Headings), used by human MEDLINE indexers.

The UMLS's contents centre on concepts, terms, strings, and words. A *concept*, such as hypertension, may be expressed in different ways: e.g., "disease" and "hypertensive vascular disease" are different synonymous forms, or *terms*, that refer to the same concept. (Every concept is given expressed in multiple *string* forms through variations such as transposition of words, punctuation, or differences in case, person and tense, etc.) is composed of one or more *words*. The UMLS contains numerous cross-reference tables that greatly ease the programmer's task. For example, containing a particular word. A concept can belong to one or more *semantic categories* (e.g., pharmacological substance, therapeutic procedure) the *ID of the source vocabulary* (e.g., ICD-10) from which it was taken. This information allows researchers to create UMLS data subsets for

:: Term recognition in automated concept indexing



Python Online Training

Develop must-have Python-based programming skills that top IT firms are looking out for



Book Doctor Appointment Online

Preferred Date & Time | Get Confirmed Appt. with Experienced Doctors & Surgeons. Book Now!

Yashoda Hospitals

We now discuss the methods used for automated concept indexing.

Phrase-Based Methods use natural language processing (NLP) methods to scan the text and identify words/phrases of interest, which are then matched against the vocabulary. These methods have utilized phrases representing nouns (e.g., "obstructive lung disease"), as in the work of Elkin et al[12]. Aronson and Rindfleisch at the National Cancer Institute (NCI) used a similar approach, as described in several published papers[13],[14],[15]. If the entire phrase does not have a match in the vocabulary, one generates subsets of the phrase and matches them against the vocabulary. Thus, "digitalis-induced atrial fibrillation" does not match any concept in the UMLS, but the subsets "digitalis-induced" and "atrial fibrillation" do.

A problem with using noun phrases alone is that, in medical text, many concepts can only be correctly identified through other parts of speech. For example, "blood pressure was greatly elevated", which implies hypertension, as opposed to blood pressure alone. Verb phrases such as "surgically resected" are excluded. Further, the same concept may be divided across two noun phrases, e.g., "hypertension is secondary to renal disease," which indicates hypertension.

In an attempt to overcome the limitations of single noun phrases, alternative approaches have attempted to use larger units of text, such as sentences. Algorithms devised by Hersch[16],[17],[18]. Using larger text units yields more sensitivity, but at the cost of specificity; that is, there is an increased number of false matches. A text segment "spleen rupture and normal stomach" in an emergency surgery note will match the concept of "stomach rupture" which, while incorrect, is a match. In addition, using an entire sentence is no guarantee of success. In the admittedly artificial example, "Blood pressure was recorded in the emergency room and was greatly elevated," the concept of hypertension is split across two sentences. Finally, the time required to generate subsets of the input phrase (and search the vocabulary) is prohibitive for large sentences. Production systems therefore tend to favour phrase-based approaches.

:: Practical limitations of automatic concept detection in medical text

For medical text, detection of a concept in a document does not, per se, make that document relevant for that concept. The concept may refer to a condition ruled out, or that occurred in the remote past. The recording of "significant negatives" is important in medicine, and robust recognition of negatives is a challenge.

Concept recognition addresses the problem of synonyms (different phrases for the same concept) but fails in the presence of *homonyms* (words with multiple meanings). For example, the term "anaesthesia" by itself can indicate either a procedure for pain relief, or a clinical finding of loss of sensation. Disambiguation (selection of the correct concept from several possible matches) can be performed based on neighbouring words. Thus, the phrase "endotracheal anaesthesia" can be uniquely identified as a procedure. Disambiguation is present further away, e.g. in the previous or succeeding sentence, the problem is much more difficult. Elaborate rules have been devised to handle such cases, but the approach of manual rule creation does not scale up: the 1999 edition of the UMLS lists 13,688 ambiguous terms that manually would be a Herculean task. There may be other methods, however, that are less labour intensive than manually devising rules (e.g., using machine learning).

The presence of elisions, neologisms, or abbreviations in medical text complicates automated concept recognition. An *elision* is a phrase with

disambiguate the phrase based on the phrase's context. For example, in the phrase "the white count was 1800", "white count" is an elided form of a newly coined word that would not be expected to be in a dictionary. An example is "coumadinize" for the act of administering coumadin (as it is attained). Abbreviations that are also words in their own right can confuse a computer program, e.g., "RAT" can mean "recurrent acute tonsillitis". There are numerous entries for elisions and abbreviations, but it is by no means complete: it is therefore updated continually, with new data being released as they become available.

Some of the practical problems faced in automated concept indexing are discussed in [19].

:: IR and traditional databases

While medical text is part of the patient record, it must coexist with more structured parts of the record that are stored in tables of a traditional relational database: demographic information or laboratory results. Historically, however, such integration has been lacking, and hospital-based software developers have not integrated IR technology into their systems. Over the last two years, the situation has improved considerably as relational database system vendors have integrated IR technology into their products. The version 7 incorporates a full-text indexing engine, as does the "ConText option" of Oracle versions 7 and greater. Such integration enables the integration of text and data management.

Feature-wise, however, these offerings currently lag greatly behind those of companies that specialize in offering IR technology alone. For example, the PDR provides the search engine for the well-known Physician's Desk Reference (PDR) database, goes beyond Microsoft and Oracle in several ways: it provides a thesaurus, as well as allowing proximity searching (which has been discussed earlier in the section on document indexing). While it is possible to use a programming language if programming expertise is available (several engines, such as INQUERY [20], are freely available for researchers), it is best to go with a commercial engine for most purposes. The commercial engines have various extra features that would be highly tedious to program by oneself. For example, they understand HTML, MS-Word, and PDF. Other packages, such as IBM's Intelligent Miner for Text [21], constitute a full-fledged development environment for building a "Web crawler" that will search the Web on a periodic basis for keywords of interest, and a relational database engine where the results that are found are stored.

For many queries of medical data, it makes sense to combine a query of data present in traditional relational tables with that present in free text. For example, to find patients of a particular age, sex, and diagnosis who presented with particular clinical findings. Age, sex, and diagnosis are stored in traditional relational tables, while the clinical findings may be stored in narrative. Because querying relational tables for laboratory test values or diagnosis codes is much more efficient than querying free text, it is best to use a query of traditional and textual data [22]. MS SQL Server and Oracle allow integrated query through proprietary extensions to the database system.

When a medical institution is computerizing its electronic records, one must have a means of accessing old paper-based data that may go back many years. Information entered into a computer by hand is extremely labour-intensive, and *document management systems* (DMSs) allow one to scan the paper and reference it against essential demographic data on the patient such as medical record number, date of admission/discharge/death, primary diagnosis, etc. (If the information on paper has been typed or printed rather than hand-written, one can often use *optical character recognition* (OCR) images. OCR generates text files from the images, and one can then use IR technology, which is bundled into several DMSs, to index these files.)

:: Other areas of IR applicability

We have already discussed the document vector approach for relevance ranking. It is similarly possible, using the technique discussed earlier, to use a document vector to find documents similar to a query vector. When one gathers a series of documents (passages) written by a particular author, the set of documents, to some extent collectively acts as a literary fingerprint for the author, because particular authors tend to use some words more often than others. The proportion of verbs with a passive voice, can also be used for fingerprinting.) This knowledge has been used in various ways.

If the field of literature, if one collects several passages from different authors, document-vector methods can be used to *cluster the documents*. *With enough passages per author, one typically observes documents from a particular author falling into their own special cluster. This is suitably long passage by an author whose identity was uncertain (but about whom it was known that s/he was one of the profiled authors) do a reasonable guess as to the author's identity. This technique has been used in detecting literary forgeries (as in the "Hitler diaries", where the diaries claimed to be those of Hitler in his autobiography, "Mein Kampf"). Similarly, a comparison of passages of William Shakespeare with those of Francis Bacon indicates that the diaries were not those of Shakespeare (as has been claimed by some). During the Cold War, "Kremlin Watchers" in the CIA were able to classify a particular communication as being from one of a small handful of anonymous writers based upon which cluster the communiqué matched most closely. (These writers' identities were never revealed.)*

The Pubmed database, maintained by US National Centre for Biotechnology Information (part of the National Library of Medicine), provides a service that allows one to find all other references in the database that have been determined to be similar, by clicking a button on the Web form. It uses an algorithm devised by NCBI's John Wilbur to compute similarity. For every article in medicine, supercomputer-class machine MEDLINE are similar to it (in terms of passing a statistical significance threshold). This information is stored in the PubMed database so that one can instantly find out all other references in the database that have been determined to be similar, by clicking a button on the Web form. The full text of the article where available electronically; otherwise, only the abstract is used. The similarity algorithm is quite powerful, and it has been used to identify similarities despite the fact that it was working with translations in the latter case. The journal Science reported that a researcher using PubMed discovered a case of serial plagiarism [23]. Several papers published by a particular Polish scientist had in fact been translated into Polish without Acknowledgment, and passed off as the scientist's own work to identify similarities despite the fact that it was working with translations in the latter case.

:: Conclusions














Information retrieval technology is the hidden workhorse that underlies the powerful Web-searching facilities that all of us take for granted in our daily lives. It is an armamentarium of the developer of computer software. Just as relational database packages have become progressively easier to use and as microcomputer database packages will include IR components.

:: Grant support

: NIH Grants R01 LM06843-01 from the US National Library of Medicine and U01 CA78266-03 from the US National Cancer Institute.

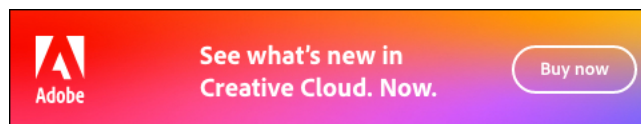
:: References

1. Shannon C, Weaver W. *Mathematical Theory of Communication*. University of Illinois Press; Chicago, IL: 1963. ■
2. Ziv J, Lempel A. A Universal Algorithm for Sequential Data Compression. *IEEE Transactions on Information Theory* 1977; IT-23: 337-343.

- [3.](#) Cipra B. *The Ubiquitous Reed-Solomon Codes*. SIAM (Society for Industrial and Applied Mathematics) News 1993; 26. 
- [4.](#) Salton G. *Automatic Text Processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley; Reading, MA: 1988.
- [5.](#) Salton G. *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice Hall; Englewood Cliffs, NJ: 1990.
- [6.](#) Hersh WR. *Information Retrieval: A Health Care Perspective*. Springer-Verlag; New York, NY: 1996. 
- [7.](#) Porter MF. An algorithm for suffix stripping. *Program* 1980; 14:130-137. 
- [8.](#) Baeza-Yates R, Frakes WB. *Information Retrieval: Data Structures and Algorithms*. Prentice-Hall; Englewood Cliffs, NJ: 1993. 
- [9.](#) Salton G, Wu H, Yu CT. Measurement of Term Importance in Automatic Indexing. *J Am Soc Inf Sci* 1981; 32:175-186. 
- [10.](#) Wilbur WJ, Yang Y. An analysis of statistical term strength and its use in the indexing and retrieval of molecular biology texts. *Com*
- [11.](#) Lindberg DAB, Humphreys BL, McCray AT. The Unified Medical Language System. *Methods of Information in Medicine* 1993; 32.
- [12.](#) Elkin PL, Cimino JJ, Lowe HJ, Aronow DB, Payne TH, Pincett PS, et al. Mapping to MESH: the art of trapping MESH equivalenc the Annual Symposium on Computer Applications in Medical Care; 1988; pp185-190. 
- [13.](#) Aronson A, Rindflesch T, Browne A. Exploiting a large thesaurus for information retrieval. In: *Proceedings of the RIAO*; 1994; pp1
- [14.](#) Aronson AR, Rindflesch TC. Query expansion using the UMLS Metathesaurus. In: *Proceedings of the AMIA Annual Fall Symposiu*
- [15.](#) Rindflesch TC, Aronson AR. Ambiguity resolution while mapping free text to the UMLS Metathesaurus. In: *Proceedings - the Annu Medical Care.*; 1994; pp240-244. 
- [16.](#) Hersh W, Greenes R. SAPHIRE--an information retrieval system featuring concept matching, automatic indexing, probabilistic ret Biomed Res 1990; 23:410-425. 
- [17.](#) Hersh W, Hickam D, Leone T. Words, concepts, or both: optimal indexing units for automated information retrieval. In: *Proceeding Applications in Medical Care*; 1992; pp644-648. 
- [18.](#) Hersh W, Hickam D. A comparison of retrieval effectiveness for three methods of indexing medical literature. *Am J Med Sci* 1992; .
- [19.](#) Nadkarni PM, Chen RS, Brandt CA. UMLS Concept Indexing for Production Databases: A Feasibility Study. *J Am Med Inform Ass*
- [20.](#) Callan J, Croft W, Harding S. The INQUERY Retrieval System. In: *Proceedings of the International Conference on Database and E* 356. 
- [21.](#) IBM Corporation. *Intelligent Miner for Text*. 2.3ed; 1999. 
- [22.](#) Vasanthakumar SR, Callan JP, Croft WB. Integrating INQUERY with an RDBMS to support text retrieval. *Bulletin of the IEEE Tec*. 19; 24-33. 
- [23.](#) Marshall E. Medline searches turn up cases of suspected plagiarism (News). *Science* 1998; 279:473-474. 

This article has been cited by

- 1 **Negation recognition in medical narrative reports**
Rokach L, Romano R, Maimon O
INFORMATION RETRIEVAL. 2008; 11(6): 499-538
[\[PubMed\]](#).
- 2 **Matching clinical and biological needs with emerging imaging technologies**
Vannier MW, Staab EV, Clarke LP
PROCEEDINGS OF THE IEEE. 2003; 91 (10): 1562-1573
[\[PubMed\]](#).



[Previous arti](#)

[Site Map](#) | [Home](#) | [Contact Us](#) | [Feedback](#) | [Copyright and Disclaimer](#)

Online since 12th February '04

© 2004 - 2020 Journal of Postgraduate Medicine

Official Publication of the Staff Society of the Seth GS Medical College and KEM Hospital, Mumbai, India

Published by Wolters Kluwer - [Medknow](#)