Information Retrieval: A Study of Various Models and Methods

Gurpreet Kaur
Assistant Professor
Department of Computer Science
Guru Nanak Khalsa Girls College,
Baba Sang Dhesian Phagwara, Punjab

Abstract: Information Retrieval is become a research area in the field of Computer Science. Information Retrieval is the process of tracing and recovery of specific information from Database. It is a continuous process during which we consider, reconsider and refine the research problems by using different retrieval techniques. It is crucial to documentation and organization of knowledge. This paper provides a comprehensive study of major Information Retrieval Models and Methods and their Applications.

Keywords: Information Retrieval (IR), Indexing, IR model, Searching, Vector Space Model (VSM).

1. INTRODUCTION

Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers). It is the task of representing, storing, organizing, and offering access to information items. IR is different from data retrieval, which is about finding precise data in databases with a given structure. Information retrieval is generally considered as a subfield of computer science that deals with the representation, storage, and access of information [1]. Information retrieval is concerned with the organization and retrieval of information from large database collections [2]. Information Retrieval (IR) is the process by which a collection of data is represented, stored, and searched for the purpose of knowledge discovery as a response to a user request (query) [3]. Information retrieval is a problem-oriented discipline, concerned with the problem of the effective and efficient transfer of desired information between human generator and human user. In other words:

- The indexing and retrieval of textual documents.
- Concerned firstly with retrieving relevant documents to a query.
- Concerned secondly with retrieving from large sets of documents efficiently.

An Example:

- Alternative: Use a High Speed Computer to read entire document collection and extract the relevant documents.
- Goal = find documents relevant to an information need from a large document set

IR Process

IR process involves various stages initiate with representing data and ending with returning relevant information to the user. Intermediate stage includes filtering, searching, matching and ranking operations. The main goal of information retrieval system (IRS) is to “finding relevant information or a document that satisfies user information needs”. The Figure 2 sketches the outline of information process.

Here, the user issues a query q from the front-end application (accessible via, e.g., a Web browser); q is processed by a query interaction module that transforms it into a “machine-readable” query q' to be fed into the core of the system, a search and query analysis module. This is the part of the IR system having access to the content management module directly linked with the back-end information source (e.g., a database). Once a set of results r is made ready by the search module, it is returned to the user via the result interaction module; optionally, the result is modified (into r') or updated until the user is completely satisfied.
2. IR MODELS

An IR model specifies the details of the document representation, the query representation, and the retrieval functionality. The fundamental IR models can be classified into Boolean, vector, probabilistic, and inference network model. The rest of this section briefly describes these models.

2.1 Boolean Model

The Boolean model is the first model of information retrieval and probably also the most criticized model. The model can be explained by thinking of a query term as an unambiguous definition of a set of documents. For instance, the query term economic simply defines the set of all documents that are indexed with the term economic. Using the operators of George Boole's mathematical logic, query terms and their corresponding sets of documents can be combined to form new sets of documents. The Boolean model allows for the use of operators of Boolean algebra, AND, OR, and NOT, for query formulation, but has one major disadvantage: a Boolean system is not able to rank the returned list of documents. In Figure 2, the retrieved sets are visualized by the shaded areas.

2.2 Vector Space Model

Gerard Salton and his colleagues suggested a model based on Luhn's similarity criterion that has a stronger theoretical motivation (Salton and McGill 1983). They considered the index representations and the query as vectors embedded in a high-dimensional Euclidean space, where each term is assigned a separate dimension. The vector space model can best be characterized by its attempt to rank documents by the similarity between the query and each document. In the Vector Space Model (VSM), documents and query are represented as a vector and the angle between the two vectors is computed using the similarity cosine function. Similarity Cosine function can be defined as:

\[ \text{sim}(d_j, q) = \frac{d_j \cdot q}{\|d_j\| \|q\|} = \frac{\sum_{i=1}^{N} w_{i,j} w_{i,q}}{\sqrt{\sum_{i=1}^{N} w_{i,j}^2} \sqrt{\sum_{i=1}^{N} w_{i,q}^2}} \]

Documents and queries are represented as vectors.

\[ d_j = (w_{1,j}, w_{2,j}, \ldots, w_{k,j}) \]
\[ q = (w_{1,q}, w_{2,q}, \ldots, w_{k,q}) \]

The Vector Space Model has been introduce term weight scheme known as tf-idf weighting. These weights have a term frequency (tf) factor measuring the frequency of occurrence of the terms in the document or query texts and an inverse document frequency (idf) factor measuring the inverse of the number of documents that contain a query or document term.

2.3 Probabilistic Model

Whereas Maron and Kuhns introduced ranking by the probability of relevance, it was Stephen Robertson who turned the idea into a principle. He formulated the probability ranking principle, which he attributed to William Cooper, as follows (Robertson 1977). The most important characteristic of the probabilistic model is its attempt to rank documents by their probability of relevance given a query. Documents and queries are represented by binary vectors \( \sim \) and \(-\) each vector element indicating whether a document attribute or term occurs in the document or query, or not. Instead of probabilities, the probabilistic model uses odds \( O(R) \), where \( O(R) = \frac{P(R)}{1 \cdot P(R)} \), \( R \) means "document is relevant" and \(-R\) means "document is not relevant".

2.4 Inference Network Model

In this model, document retrieval is modeled as an inference process in an inference network. Most techniques used by IR systems can be implemented under this model. In the simplest implementation of this model, a document instantiates a term with a certain strength, and the credit from multiple terms is accumulated given a query to compute the equivalent of a numeric score for the document. From an operational perspective, the strength of instantiation of a term for a document can be considered as the weight of the term in the document, and document ranking in the simplest form of this model becomes similar to ranking in the vector space model and the probabilistic models described above. The strength of instantiation of a term for a document is not defined by the model, and any formulation can be used.

Fig: 2. A high-level view of the IR process

Fig: 3 Boolean combinations of sets visualised as Venn diagrams
3 INDEXING TECHNIQUES
There are several popular information retrieval indexing techniques, including inverted indices and signature files.

3.1 Signature File
In signature file method each document yields abit string ("signature") using hashing on its words and superimposed coding. The resulting documents signatures are stored sequentially in a separate file called signature file, which is much smaller than the original file, and can be searched much faster.

3.2 Inversion Indices
Each document can be represented by a list of keywords which describe the contents of the document for retrieval purposes [6]. Fast retrieval can be achieved if we invert on those keywords. The keywords are stored, eg alphabetically; in the index file for each keyword we maintain a list of pointers to the qualifying documents in the postings file. This method is followed by almost all the commercial systems.

4 SEARCHING TECHNIQUES
There are various searching algorithms, including linear search, binary search, brute force search etc.

some general searching algorithms are described below:
1) In linear search algorithm is a method of finding a particular element or keyword from list or array that checks every element in list, one at a time and in sequence. Linear search is a simplest search algorithm. One of the most important drawbacks of linear search is slow searching speed in ordered list. This search is also known as sequential search.
2) Brute force search is a very general problem solving technique that consists of systematically enumerating all possible candidates for the solution and checking whether each candidate satisfies the problem's statement. Brute force algorithm is simple to implement and it will always find a solution if it exists.
3) Binary search algorithm, finds specified position of the element by using the key value with in a sorted array. In each step, the algorithm compares the search key value with the key value of the middle element of the array. If the keys match, then a matching element has been found and its index, or position, is returned. Otherwise, if the search key is less than the middle element's key, then the algorithm repeats its action on the subarray to the left of the middle element; if the search key is greater, on the subarray to the right.
If the remaining array to be searched is empty, then the key cannot be found in the array and special "not found" indication is returned.

5. TYPES OF IR SYSTEMS
There are three types of IR systems, which are explained below.

5.1 Textual Retrieval
Textual operations translate the user’s need into a logical query and create a logical view of documents. The most widespread applications of IR are the ones dealing with textual data. As textual IR deals with document sources and questions, both expressed in natural language, a number of textual operations take place on top of the classic retrieval steps. Figure 4 sketches the processing of textual queries typically performed by an IR engine:
1. The user need is specified via the user interface, in the form of a textual query qU (typically made of keywords).
2. The query qU is parsed and transformed by a set of textual operations: the same operations have been previously applied to the contents indexed by the IR system; this step yields a refined query q'U.
3. Query operations further transform the preprocessed query into a system-level representation, qS. Textual IR exploits a sequence of text operations that translate the user’s need and the original content of textual documents into a logical representation more amenable to indexing and querying.

5.2 Document Retrieval
In Document Retrieval, some processes take place dynamically when the user inputs their query, while other processes take place off-line in advance and in batch mode and do not involve individual users. These static processes are run on the documents that will be made available in the retrieval system. These will be explained first. Then, the two dynamic processes, Query Processing and Matching, will be presented. Figure 5 provides a simple, but clear view of the relationship between these three processes.

Fig: 4 Architecture of a textual IR system.

Fig: 5 Basic Components of a Document Retrieval System
Document Processing: The first two steps in the processing of documents are somewhat mundane, but necessary, and can be considered as batch pre-processing. These are:

1. Normalize document stream to a predefined format, whereby multiple external formats (e.g. newsfeeds, web pages, word processed documents) are standardized into a single consistent format. This is an essential step (much akin to data clean-up in data mining) as all downstream processes rely on receiving a common format they can recognize and process. Preprocessing is particularly vital for systems with more complex processing than simple 'characters between white spaces' indexing.

2. Break document stream into desired retrievable units, whether this is web page, chapter, full document, paragraph, etc. The pointers stored in the inverted file are to whatever unit size has been pre-determined. Therefore, document retrieval could in fact be paragraph retrieval, if the indexable unit was determined at this stage to be the paragraph.

3. Identify potential indexable elements in documents. This is a key decision point that dramatically affects the nature and quality of the retrieval performance. First, the important definition needs to be made as to what is a term. Is it any string of alphanumeric characters between blank spaces or punctuation? If so, are non-compositional phrases or multi-word proper names, or inter-word symbols such as hyphens or apostrophes treated differently (e.g. are “small business men” and “small businessmen” the same)? At this stage, the system requires a set of rules to be executed which control what actions are taken by the ‘tokenizer’ – the algorithm which recognizes ‘indexable terms’. IR systems vary as to which of these processes they perform, but the most frequently used processes are:

   a. Delete stop words via an algorithm that filters the document’s potential indexable elements against a Stop Word list to eliminate terms that are deemed to be insignificant in determining a document’s relevance to a user’s request. The original objective in using stop words was to save system resources by eliminating those terms that have little value for retrieval performance.

   b. Stem terms by removing suffixes. In this morphological step, some IR systems do just inflectional (‘weak’) stemming which only changes the subclass within a part-of-speech category, i.e. past tense to present tense, while others also do derivational (‘strong’) stemming which removes suffixes, sometimes recursively, that may actually change the part of speech of a word. Use of stemming will result in fewer entries in an index, each of which is likely to have higher frequency counts than if all morphological variants and their counts are used.

   c. Bracket noun phrases, usually by means of regular expressions which define the part-of-speech patterns which comprise a noun phrase (e.g. <ADJ NN> or <NN NN>). This is a step that can negatively affect recall of retrieval results by either excluding documents when the phrasal expression in the query is not exactly the same as the index entry of a document, or positively affect precision by retrieving only documents that include the terms in the desired phrasal expression.

   4. Produce an inverted file containing a sorted array of all indexable terms (with terms defined as referring to either a word or a phrase), along with the unique identification number of each document in the collection in which the term occurs, a link to each of these documents, weights for each term as determined by the IR model being implemented in the system (and optionally, the within-document location of the term). More sophisticated systems may include further information in the inverted file, such as named entity category for proper names (i.e. PERSON, ORGANIZATION, GEO-LOCATION, etc) but the most common features are simply term, document ID, and weight.

Query Processing: The system’s internal representation of the user’s question / searcher terms is typically referred to as the query. Most of the same processes that are run on the documents are also run to produce the query, but there are some unique processes as well. As distinct from document processing, all of the query processing is done in real time, while the user awaits their documents. These are:

   1. Recognize query terms vs. special operators, such as “I need information about...” which do not convey the topic of the user’s information need and will not be included in the query representation.

   2. Tokenize query terms, a process that requires similar decisions as were described on the document processing side – that is stop word deletion, stemming, and phrasal recognition.

   3. Create query representation, which typically follows stop word removal and stemming, and which may also include insertion of logical operators between / amongst terms requiring co-occurrence or simple presence of only one of the arguments.

   4. Expand query terms to include variant terms that refer to or relate to the same concept. Query expansion relies on the user of needing to generate all conceptual variants of their search terms and is likely to improve recall, but may reduce precision when erroneous senses of the newly introduced terms retrieve irrelevant documents. The longer query is, the less likelihood that erroneous senses of expanded terms will have an adverse impact, but also the less likely that expansion will contribute much to retrieval results.

   5. Compute query term weights. This step is less commonly included in Document Retrieval systems, mainly because it is difficult both for users to know how to assign weights to query terms in a way that improves retrieval results, or for automatic weighting, since queries are frequently so short that they reduce the importance of query terms as most terms only occur once in a single query. The process description below may be easier to follow if you conceive of both the query and the documents as vectors of terms, with each term being a vector in the vector space.

   a. Search inverted file for documents that contain terms in the query. This is typically done using a standard binary search. Each document that contains any of the query terms becomes a candidate for retrieval.

   b. Compute similarity score between query and each candidate document using the algorithm prescribed by one of the four Document Retrieval models being used. This score is referred to as the Similarity Coefficient.

   c. Rank the documents in decreasing order based on the scores assigned them by the scoring algorithm.

   d. Provide list of perceived relevant documents to user ranked by similarity score between query and document. Systems that utilize other sources of evidence of value of a document to the query, such as number of links from the page/document to
or from other pages/documents, would integrate this information and produce a potentially different ranked list.

5. Allow for query modification by the user if user-based relevance feedback is provided by the system. If so, typically, the user marks the documents they find relevant, either based on just the title and brief description shown them in the initial list or by actually reviewing the full document, which they can link to from the results page.

6. Perform relevance feedback based on user’s input. The algorithm for user-based relevance feedback is typically the same as that for automatic relevance feedback as described in Step 3 above. The system then re-runs the search with the revised, and hopefully improved, query and produces a revised ranked list of documents. Therelevance feedback loop is iterative and can be performed as many times as the user wants.

5.3 Image Retrieval Systems

Content-based image retrieval, uses the visual contents of an image such as color, shape, texture, and spatial layout to represent and index the image.

![Diagram](image)

**Fig: 6 Diagram for content-based image retrieval system**

In typical content-based image retrieval systems (Figure 6), the visual contents of the images in the database are extracted and described by multi-dimensional feature vectors. The feature vectors of the images in the database form a feature database. To retrieve images, users provide the retrieval system with example images or sketches.

The system then changes these examples into its internal representation of feature vectors. The similarities/distances between the feature vectors of the query example or sketch and those of the images in the database are then calculated and retrieval is performed with the aid of an indexing scheme. The indexing scheme provides an efficient way to search for the image database. Recent retrieval systems have incorporated users' relevance feedback to modify the retrieval process in order to generate perceptually and semantically more meaningful retrieval results.

6. APPLICATIONS

Information retrieval is used today in many applications. It is used to search for documents, content thereof, document metadata within traditional relational databases or internet documents more conveniently and decrease work to access information. Retrieved documents should be relevant to a user’s information need. Obvious examples include search engines as Google, Yahoo or Microsoft Live Search.

6.1 Digital Library

A digital library is a library in which collections are stored in digital formats (as opposed to print, microform, or other media) and accessible by computers. The digital content may be stored locally, or accessed remotely via computer networks. A digital library is a type of information retrieval system. Many academic libraries are actively involved in building institutional repositories of the institution's books, papers, theses, and other works which can be digitized or were 'born digital'. Many of these repositories are made available to the general public with few restrictions, in accordance with the goals of open access, in contrast to the publication of research in commercial journals, where the publishers often limit access rights. Institutional, truly free, and corporate repositories are sometimes referred to as digital libraries.

6.2 Recommender systems

Recommender systems or recommendation engines form or work from a specific type of information filtering system technique that attempts to recommend information items (films, television, video on demand, music, books, news, images, web pages, etc.) that are likely to be of interest to the user. Typically, a recommender system compares a user profile to some reference characteristics, and seeks to predict the 'rating' that a user would give to an item they had not yet considered. These characteristics may be from the information item (the content-based approach) or the user's social environment (the collaborative filtering approach).

Collaborative filtering is concerned with making recommendation about information items (movies, music, books, news, web pages) to users.

6.3 Search Engines

A search engine is one of the most the practical application of information retrieval techniques to large scale text collections. Web search engines are best-known examples, but many others searches exist, like: Desktop search, Enterprise search, Federated search, Mobile search, Social search.

A web search engine is designed to search for information on the World Wide Web. The search results are usually presented in a list of results and are commonly called hits. The information may consist of web pages, images, information and other types of files. Some search engines also mine data available in databases or open directories. Unlike Web directories, which are maintained by human editors, search engines operate algorithmically or are a mixture of algorithmic and human input. Reliability of information is a pre-requisite to get most from research information found onto the web. A frequently encountered issue is that search terms are ambiguous and thus documents from a different non-relevant context are retrieved or you may not know which terms describe your problem properly, especially if you are a non-expert user in this particular domain. The novel idea of relevance feedback allows users to rate retrieved documents as relevant or less relevant and thus help other users to find
documents more quickly. These ideas were adopted from image retrieval. Images are hard to describe using words.

6.4 Media search
An image retrieval system is a computer system for browsing, searching and retrieving images from a large database of digital images. Most traditional and common methods of image retrieval utilize some method of adding metadata such as captioning, keywords, or descriptions to the images so that retrieval can be performed over the annotation words. Manual image annotation is time-consuming, laborious and expensive; to address this, there has been a large amount of research done on automatic image annotation. Additionally, the increase in social web applications and the semantic.

7. CONCLUSION
Information retrieval is a process of searching and retrieving the knowledge based information from collection of documents. This study dealt with the basics of the information retrieval. In this paper we have also discussed about the different indexing techniques and searching techniques. This paper also includes the area of IR applications.

REFERENCES
[5] Introduction to Information Retrieval – Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze
[6] Data Mining: Concepts and Techniques - Jiawei Han & Micheline Kamber