

BASIC CONCEPTS FOR DATABASE SEARCHING

By Marvin Hunn

Here are some important terms and concepts related to database searching. Below you may see sample search statements typographically distinguished by angle brackets like this: << search statement here >>.

Search engines search for words, not concepts. Search engines look for the words and phrases you supply. They don't understand the words. And there is not a one-to-one match between words and concepts. For example, one word may have many meanings (polysemy), and two words may have approximately the same meaning (synonymy). To overcome problems like this, you can often improve a search statement. But be realistic; the search engine does not understand what you want; it can only do what you tell it to do. You are the one who must do the thinking.

Document. For our purposes, this means a recorded intellectual work. It can use any recording technology. For example, a document might be a print book, an ebook, an article in a journal, a sound file in MP3 format, a hand-written manuscript, a photograph, a clay tablet, or an oil painting.

Metadata. Bibliographic metadata is information that describes a document, such as author, title, date, subject, and abstract. So for bibliographic databases we have data (documents) and metadata (descriptive information about the documents). Metadata is often organized into records, one record per document, and the records organized into a database.

Record. A database record is a unit of information in a database. It consists of one or more fields. Each record is about one document, and each field is about one specific aspect of the document. For example, a record might be about one book, and might include fields for author, title, publisher, subject, etc. Here is a simplified library record with typical fields labeled.

Author:	Bibfeldt, Franz
Title:	John Knox and the British Reformations
Publication info:	Dallas : Nonesuch Press, c2001.
Series:	Studies in Reformation history
Subject:	Reformation—Great Britain
Subject:	Great Britain—Church history—16th century
Subject:	Knox, John, ca. 1514-1572

Controlled vocabulary database. This is a database which uses standardized terminology to describe documents. Standardization is meant to guarantee a name or concept is always expressed in a consistent way. Standardization lessens problems caused by variant spelling (e.g., Koran or Quran or Qur'an) and variant forms (e.g., J Smith vs John Smith) and synonyms (e.g., anger vs wrath). So controlled vocabulary supports consistency and cross references from non-standard terminology to standard terminology.

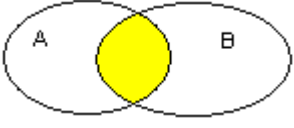
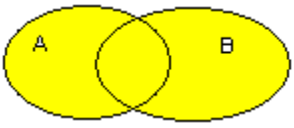
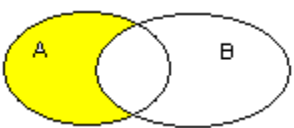
Subject headings. Descriptors and subject headings are terms (words and phrases) assigned to documents to indicate the subject/topic. Normally these terms are drawn from a Thesaurus which lists all the standardized headings.

Search engine. Search engines are computing systems that attempt to find items (records, documents) whose contents satisfy the conditions specified in the search statement. Google is an example.

Search operators. Many search engines require/allow you to specify not only search terms, but also information about how to relate those terms to each other. Operators are commands that tell the search software how to relate terms. For example, you might use an exact phrase operator to tell the software you want the word infant adjacent to the word baptism (no intervening words) and you want them in that exact word order. (So your search query would match the book title “History of Infant Baptism” but would not match the book title “Why we Baptize Infants.”)

Search statement. A search statement is a combination of terms, operators and options that constitute a search. For example << (law OR covenant) AND (romans OR galatians) >> is a search statement. This definition assumes the searcher is using a procedural command language with operators to explicitly tell the search engine what to do (in contrast to using a natural language interface).

Logical search operators. Most search engines support the logical operators AND, OR, NOT (also called Boolean operators after George Boole, the mathematician who popularized their use in set operations).

Operator or function	Common symbol	Example	Explanation	
intersection	AND	A AND B church AND state		'AND' retrieves records containing both terms. You will commonly AND concepts to narrow a search.
union	OR	A OR B clergy OR pastor		'OR' retrieves records containing either term. You will commonly OR synonyms to broaden a search.
exclusion	NOT	A NOT B spirit NOT holy		'NOT' excludes records containing the second term. It is easy to accidentally exclude desired material.

Grouping and nesting. Many retrieval systems allow the searcher to use parentheses to group terms and specify the order in which search operators are to be executed. For example, consider this search:

<< brown AND dog OR cat >> It is pretty clear that the dogs must be brown. But what about the cats? Most systems would retrieve cats of any color. To specify both cats and dogs must be brown we would use parentheses like this: << brown AND (dog OR cat) >> This specifies the OR operator is to be executed first, creating a set that holds the results of the dog OR cat. Then that intermediate result is to be AND-ed

with brown to yield a final result. Nesting refers to embedding one set of parentheses within another set like this: A OR (B AND (C OR D)).

Truncation operator. This operator allows partial word matches by truncating (cutting off) part of the word, usually the final part. Truncation is usually indicated by * (asterisk). For example, in some systems bapt* matches any word that starts with the four letters "bapt"(such as baptism and baptist). On some systems you must use * to match final possessive s (e.g., moses* to match Moses's). It is easy to accidentally include undesired material when you truncate. Some systems have "wildcard" or character masking operators that can be used in the middle of a word (e.g., wom#n to match woman or women).

Proximity search operators. Many search engines support proximity operators. Proximity operators specify how far apart matching words can be. They may also specify word order. Details vary considerably system to system. Proximity operators tend to be more precise than logical operators, and more appropriate for full-text searching. Examples follow.

Operator or function	Common symbol	Example	Explanation
proximity	NEAR + number N + number	spirit NEAR4 filled	'NEAR' specifies maximum distance between words in the same field, in any word order. The example specifies a maximum distance of four words (NEAR4). It matches "Spirit filled" as well as "filled with the Spirit". Systems count distance differently. Some say the distance between spirit and filled in the exact phrase "spirit filled" is zero , but others say the distance from spirit to filled is one word.
Exact phrase	". . ."	"spirit filled"	Most systems will interpret quotation marks as an exact phrase operator. It specifies distance between words plus word order. Use straight double quotes ("spirit filled"), not curly quotes ("spirit filled"), not single quotes ('spirit filled').
Word order specific proximity	WITH + number W + number PRE/ + number	spirit W4 filled	This operator is similar to NEAR: it specifies maximum distance between words in the same field, but it also specifies word order. So you can use it to match an exact phrase like 'infant baptism'. Use it to allow intervening modifiers. For example, <<big W4 dog >> matches 'big brown dog' as well as 'big dog.'

Default operator. The search engine will use the default operator if the search statement does not explicitly indicate how to relate multiple search terms. For example, if you type a two word search

statement like << word1 word2 >>, the software might AND the terms or look for an exact phrase or do something else. The default operator for EBSCO is n5 (near five).

Field codes. Field codes specify which fields are to be searched. For example, << au:calvin AND ti:institutes >> might mean search the author field for the word 'calvin,' and the title field for the word 'institutes.' The codes are usually two letter abbreviations with distinct punctuation. Most interfaces provide search forms with some means of selecting the fields you wish to search, so you don't have to memorize the codes. But the search forms don't accommodate complex search statements. So you may want to use the codes.

Keyword. A keyword is any arbitrary searchable word (in contrast to an entire field). It could be any word anywhere in a record or document. It does not mean "important word."

Stopwords or noise words. These are specific words that a system does not search for. To save time and storage space, some retrieval systems ignore some very common words that carry little meaning such as conjunctions, prepositions, and articles. The theory is that you do not need to search for "the" or "a" or similar words. But you do. (For example, "a" is necessary in a search for Vitamin A). Some systems will search for a stopword if you put it in quotes.

Mis-coordination is an undesired semantic relation between matched words in a multi-word search. One often mentioned example is the guy who wanted to buy car polish (a compound used to preserve and shine car paint). He searched for car AND polish. He found information about a Polish car (car manufactured in Poland). Mis-coordination is very common when logical AND is used to combine words. It is less common when exact phrases and proximity operators are used to combine words. In this example we could use an exact phrase operator to improve our search results. Even when words are syntactically related the way we want, they may still fail to be semantically related the way we want. In this example we have two different words spelled the same way (polish and Polish). This is a good example of why searching requires more than matching character strings; it requires matching meaning/concepts. (Mis-coordination is not a standard term; I am using it to cover what the IR people call "false coordination" and "incorrect term relation" and some similar retrieval problems.)

Relevance of search results. The concept of "search relevance" is slippery, context dependent, and changes with time and personal perspective. Most often we search for items "about" a certain topic. So relevance is aboutness. But this is not always correct. In another sense a retrieved document is relevant (pertinent, germane, apposite) if it provides information useful to the searcher. So relevance is usefulness. Documents may be considered relevant because they are helpful to the problem at hand even though they are not about the originally specified topic. For example, suppose a student is studying how a specific OT law is interpreted in the NT. He stumbles on an article about US constitutional interpretation which makes a point about modern legal hermeneutics. This makes him ask new questions about how the NT is interpreting the OT law. The student calls that useful article "relevant." But it never mentions the bible. It is not about biblical hermeneutics.

Search precision and recall. Don't expect a perfect search result. Search success is often assessed in terms of precision and recall. Precision refers to accuracy, and recall refers to completeness or thoroughness.

Both are percentages. Precision and recall are usually defined as follows. If d = number of documents retrieved in a search, and R = number of relevant documents in the database, and r = number of relevant documents retrieved by the search, then precision of that search = r/d and recall = r/R . For example, suppose you search for information about women in the gospel of John. You retrieve 50 documents but only 25 are relevant. So search precision is $25/50 = 50\%$ (good). The database actually contains 250 documents truly on topic, so the recall is $25/250 = 10\%$. Precision and recall are inversely related; a high precision search is usually a low recall search. Often you can design a search to be high precision or high recall, but it is usually difficult to execute a perfect search (high precision and high recall).

Full-text vs full-image database. Many databases include both metadata and links to complete documents (so you can read the books or articles online). These documents may be available as text or as images (or both). A full-text database provides online access to the text (all the words) of some documents. A full-image database provides online access to scanned images. Often Optical Character Recognition software is used to convert image to equivalent text, and the text is stored "under" the image. In such cases it is possible to copy and paste text, but expect the OCR to make mistakes.

Very often "full-text" is used to mean "complete document" so you might see a database of MP3 sound files (e.g., sermons) called a full-text database even though it is sound not text. Sound odd? Meanings evolve. For example, to "break bread together" [meaning to eat together] probably originated when bread was a major part of almost every meal, and a loaf of bread was literally broken. But now "break bread together" means eat together even if there is no bread at all in the meal. So now we have full-text databases with no text.

Auto-suggest. This software feature automatically displays a list of terms (phrases) that are similar to the terms the searcher is typing. For example, start to type
<< charitable income tax deduction >> and before you finish typing the system suggests the following.
charitable income tax deduction limits
charitable income tax deduction carry forward
charitable income tax deduction chart
charitable income tax deduction calculator

Browsing vs Searching. Some databases support two primary ways of finding material: 1) searching for words or 2) browsing a list of headings. Searching retrieves records by matching combinations of words anywhere in a record. Looking for all records with the phrase "Jesus Christ" is a search. Searching is what you normally do. Browsing is fundamentally different. It is a two-step process. First, you begin by supplying a word or phrase you expect to match the start of a field. The database then displays a sorted list of headings that start with those words. Second, you pick one or more specific headings from the list, and matching records are displayed. For example, start with Jesus Christ. The system responds with a long list that begins like this:

Jesus Christ -- Appearances
Jesus Christ -- Ascension
Jesus Christ -- Authority
Jesus Christ -- Baptism

Jesus Christ -- Betrayal

Jesus Christ -- Burial

You then pick an entry from this browse display. Auto-suggest is similar to browsing. Both involve a two-step process and both display an intermediate list to help you select terms. But intermediate lists are not created in the same way. Auto-suggest is usually based on search statements supplied by other people. Browsing is based on fields that exist in records in the database. It assumes structured records with fields and controlled vocabulary. So it is not possible to browse unstructured web pages.

Stemming and lemmatization. Linguistic stemming attempts to identify the root or stem of a word. For example, search for the word "baptize," and the system determines the stem is "bapt-" so it matches baptism, baptized, baptist etc. It may also match Anabaptist and rebaptize, depending on how stemming is performed in the particular system. Since English words are mostly inflected with suffixes, stemming is often equivalent to truncating the end of a word. (But this is not so in Hebrew, for example, which regularly inflects with prefixes and suffixes and infixes). Lemmatization attempts to identify the base or dictionary form of a word, which is known as the lemma. For example, "be" is the lemma for all of the following: are, am, is, was, were. Stemming and lemmatization are features of some special purpose software, but not standard database search engines. For example, Accordance and Logos support lemma searches of the Greek and Hebrew texts of the Bible.

Tagged text. A document is "tagged" if controlled vocabulary metadata identifiers have been inserted into the text to provide information about portions of that text. For example, every word might be grammatically analyzed (e.g. this word is a verb, first person singular, and the lemma is xzy) or geographically analyzed (this word refers to a city in Israel and here is a pointer to its location on a map). Syntactical and literary structure might be tagged (e.g. this is a verb phrase; this is an independent clause; this is a paragraph; this is a pericope within a longer work). Logos has many searchable tagged texts. You may also see the word "**treebank**" in linguistic contexts. A treebank is a corpus (collection) of tagged documents. The tagging typically includes information about phrase and clause structure, and is displayed as a tree structure of linked elements. So the "tree" part of treebank refers to the display of single documents, and the "bank" part of treebank refers to the collection of tagged documents. There are treebanks of ancient Greek text available for linguistic study, and there are some standards that govern how the texts are tagged. One example is the Diorisis Ancient Greek Corpus.

Natural language interface. A NL system allows the searcher to use plain English (or other natural language) to indicate to the system what the searcher wants. This is similar to using Siri on Mac OS or Cortana on Windows or Alexa or Google Assistant. Traditional search engines make the searcher use special syntax and operators to express a search statement the software understands (for example: << pray* near4 group* >>). Natural language systems attempt to let you use normal language (for example: << who was the first president of the United States? >>) Often the searcher has little control over a natural language system and therefore has little opportunity to help the system when it performs poorly.

Rank order. Some systems rank documents in an attempt to show the best documents first. Rank order depends on a combination of factors like relevance, popularity, etc. Sorting a retrieved set by a field value like date is not called ranking, but both ranking and sorting assign a display order.

Traditional vs Progressive information retrieval systems. Traditional information retrieval systems use explicit operators to search short highly-structured metadata records populated by controlled vocabulary headings. Modern progressive information retrieval systems support natural language interface with no explicit operators, and they rank documents. They search complete full-text documents like journal articles or books or web pages. The data store contains little structure and little metadata. Hybrid systems have aspects of both.

Traditional	Modern progressive
documents probably offline (print)	documents online
metadata; structured records with fields	little/no metadata; little or no useful structure in the documents
controlled vocabulary	free-text/natural language
search only metadata	search full documents
search engine uses operators and syntax	search engine allows brief quasi natural language input; no operators
search or browse	search and auto-suggest
cross references	no cross refs, but spelling correction and “do you mean?” suggestions
record sorting by metadata fields (e.g. date, author, call number)	document ranking by relevance or popularity

For more definitions, see <http://library.dts.edu/Pages/RM/Helps/glossary.shtml>