

A Brief Practical Introduction to Taxonomies

Dan Dabney
Senior Director for Research and Development
Thomson – West

Taxonomy is a subject that has fascinated scholars from many disciplines. This short introduction mentions some of these theoretical aspects, but concentrates instead on some of the practical considerations faced by someone setting up a taxonomic document retrieval system, particularly in a law firm setting.

Why Use Taxonomies?

Taxonomies serve two main purposes: (1) they provide authority control; and (2) they facilitate browsing.

Authority Control

Authority control is a fearsome name for the benign principal that, in an index, the terms used to describe the salient features of items in the collection should be standardized. Where there is more than one way to express an idea, one should not force the user to anticipate all possible ways to express it, but rather group all of the appropriate items under a single, standardized entry. Thus, a library catalog that uses authority control for the names of authors will not put some books under *Twain, Mark* and others under *Clemens, Samuel*.

For our purposes, we're chiefly interested in controlling authority with respect to the subject matter of documents. There is an almost unlimited number of ways that ideas can be expressed in natural language, so a document retrieval system that controls subject authority will define a special indexing language that has, as its principal characteristic, that there is only one way to express each idea that the users of the system are presumed care most about.

For example, if there are documents in the collection about *children*, the user should not have to think of all the ways that idea can be expressed in natural language: *children*, *young people*, *kids*, *minors*, and so forth. Whatever the words the documents themselves use to express the idea of children, the index should gather them together under a single heading.

The usual way to specify the standardized form of entry for subjects is simple to list all of the allowable subject headings. This list of headings is, in effect, the universe of thinkable thoughts for the system—the subject headings it contains are the ones that users can use, and if the users are interested in some other topic, they are out of luck (at least as far as the indexing system is concerned). For each of the chosen headings there is a single, standard way of representing that idea.

Not all document retrieval systems use authority control. Simple free text systems that work by requiring the user to guess the words that occur in the documents they are looking for don't control subject authority. This lack of authority control is one of the chief problems that motivate system designers to implement taxonomic systems.

Research has shown that users are not particularly good at guessing what words occur in the document they are looking for. So, while virtually all modern electronic information archives permit the user to search by guessing words, more fully realized systems also offer subject searching based on controlled authority. Providing a subject authority system is one way to help users find what they're looking for with more accuracy and less effort.

One of the most important reasons to control authority is that there are some ideas that are almost impossible to find in an uncontrolled free text system. Consider, for example, the concept of *chattel*. The word "chattel" is uncommon in modern legal writing—it is an old-fashioned word, and one that has some negative connotations. It is, however, a useful concept, denoting an item of personal property (as opposed to real property). In most legal writing, the word (or words) that carries with it the concept of chattel is simply the name of the item—the car, the dog, the computer, or whatever. The lawyer reading the document knows that the item is considered a chattel, but the concept would be essentially impossible to express comprehensively in a key word search because to do so one would need to list all of the names of objects that would be considered chattel.

In other instances, ideas are difficult to find in free text systems not because they do not correspond to particular words in the text, but because those words are too common. Consider, for example, the question of whether a lawyer who is suspected of a crime can suppress evidence obtained by questioning him before he was read his *Miranda* rights. It could be argued that it is futile and unnecessary to inform a lawyer of his right to consult a lawyer, but it can also be argued that lawyers ought to enjoy the same procedural rights as anyone else. Here the problem is that the distinctive part of the issue is represented by a word like "lawyer", which is not only exceedingly common in legal text, but is implicated (in another sense) by the very question that is being asked.

Thus, document retrieval systems that control subject authority not only enjoy some advantages of convenience and accuracy in searching, but they also make it possible for the user to find some important ideas that tend to drop out of uncontrolled systems.

Browsing Support

The second main reason to implement a taxonomic system is to support browsing. Searchers often like to proceed by looking at what types of documents are available rather than by making a formal search.

It is the rule rather than the exception that searchers approach a document retrieval system with an indistinct mental image of what they are looking for. They know some, but not all, of the salient characteristics of the document or documents they are seeking, and their notion of some of those characteristics may be fuzzy. They know what they're looking for, but they may be unable to describe it in abstract terms. If the retrieval system can work by recognition rather than description, it is likely to work better. This is especially likely in a subject index using authority control—the user will probably recognize an expression as representing the topic of interest, but may not anticipate the particular expression chosen by the index to be the canonical representation of the idea.

Consider, for example, the concept of *chattel sales*. As discussed above, few lawyers would think to search a legal document archive with the term "chattel sales" because it's an uncommon expression. Most lawyers, however, would recognize the meaning of the

term chattel sales in a browsable subject index, and would know just what sort of items it contained.

Browsable systems also enjoy an advantage in that they can identify, in context, the ideas in a collection that are important. One of the vices of free text systems is that they include all words whatever (except sometimes a stop list of very common words), including both those that are important to the legal concept discussed and those that are not. Take, for example, the words “guard rail”. These words represent an idea that is sometimes legally significant, and sometimes not. The presence or absence of guard rails may be of considerable interest in the context of a lawsuit against a state alleging negligence in the design of a bridge. But in a suit against the state for failing to pay for the purchase of guard rails, the concept of guard rail is unimportant—the case would probably have the same legal import if it concerned any other unpaid item.

Arranging an otherwise unruly collection of individual documents according to a hierarchy of subject terms allows the collection of documents to explain itself. In the same way that you can sometimes tell best what a book is about by browsing the index, you can tell what kinds of documents are available in an information system by browsing a taxonomy. The taxonomy has organized the collection according to a selection of the ideas that users find most important.

Taxonomic systems address one of the chief difficulties that beset controlled vocabulary indexes. The form of entry chosen by the system as the canonical form may not be the one that occurs to the user. For example, if a lawyer wants documents about children, he may not find them if they are listed, as they are in some older legal indexes, under the word *infants*. In flat subject indexes, it is important to supply cross references from any common expression not used by the system to the corresponding one that is used. Accordingly, the system should, in addition to listing the documents themselves under *infants*, contain such references as *children, see infants*, and *minors, see infants*. Such references are sometimes referred to as the system’s entry vocabulary. But entry vocabulary is seldom important in a browsable taxonomy because the user can see what terms are used.

Building a New Taxonomy

The process of building a taxonomy is difficult and labor intensive. It is often out of the question to build a new taxonomy from scratch, but it is well to understand the problems that one encounters in the process of creating a brand new, hand-crafted taxonomy before turning attention to the ways that an existing taxonomy can be adapted and modified for use in a new application, or the ways a new taxonomy can be created automatically.

Coverage Issues

The classic statement of the proper coverage of a classification system is that it consists of classifications that are mutually exclusive and collectively exhaustive. Modern systems tend to relax at least one of these two requirements.

Mutually exclusive categories

There are many classification systems in which it is essential that the categories be mutually exclusive, so that if an item belongs in one category it doesn’t belong anywhere

else. The most obvious application for this rule is in systems that arrange a collection of physical objects, like the arrangement of books in a library. The principle is also generally observed, though not so strictly, in many paper indexes, where posting the same item under many different headings would cause the space occupied by the index (and consequently the cost of printing and distributing it) to increase.

Much of the complexity of old-fashioned indexing systems is based on the need to decide which subject controls the primary location of items that treat more than one subject. For example, in the Key Number System, West's indexers frequently must decide which of several logical topics takes headnotes of a particular kind. One such rule is the "that which is preempted" rule. By convention, West's classifiers double-post preemption headnotes to the topic States, which takes all preemption questions, and to one substantive area. But, in order to ensure that the paper digests do not grow too quickly, they do not post to both the topic that takes the law that is being preempted and the topic that does the preemption—they post only the former. Thus, a headnote that says ERISA preempts the community property law of a state is posted to the topic *Husband and Wife*, but not to the topic *Pensions*.

The principal of mutually exclusivity makes much less sense in an electronic environment. Where a document logically belongs to more than one category, it usually ought to be put into all the categories that it belongs to.

Collectively exhaustive categories

The second conventional requirement is that the categories should be collectively exhaustive, that is, that every object should belong somewhere. This requirement also is relaxed in some modern systems.

The requirement of collective exhaustivity gives discipline to the creation and maintenance of an indexing system. If every document has to belong somewhere, it is not possible to build an indexing language that overlooks any important part of the collection it serves. When the indexer encounters a document that simply cannot be described by any of the existing categories, it is necessary to create a new one.

This requirement is sometimes finessed by creating a residuary classification called something like *miscellaneous*. This is not condoned by standard indexing theory, but it is practiced to some extent by nearly all systems. While one seldom encounters free-standing *miscellaneous* categories in a well-constructed taxonomy, one frequently encounters, at lower levels in the outline, headings like *other cases* or *in general* that take items that belong to the general area but not to any of the particular subheadings supplied.

Collectively exhaustive categories are almost always used when the indexing process itself is manual—that is, where each new item entering the system is individually considered by a human indexer. Many automatic indexing systems, however, are implemented in such a way that many items in the collection are not assigned to any category whatever.

Where the subject index is the only way to retrieve items, not assigning a document to any heading is tantamount to not adding it to the system at all. Where, however, there are alternate ways to retrieve documents (such as by free text queries), it may make sense to eliminate the collective exhaustivity requirement.

But allowing a system to contain documents that don't belong to any category in the taxonomy removes an important source of discipline on the indexing language, and can

mislead the user. As one of the virtues of a taxonomic system is that it allows the user to see what kinds of documents the system contains, omitting any subject from the classification language that actually occurs in the collection is likely to cause some users to conclude incorrectly that the system does not contain any helpful items when it does.

Literary Warrant and Use Warrant

The next consideration in building a taxonomy is selecting the terms to be included. There are two general principles that govern the choice of terms: literary warrant and use warrant.

Literary warrant is the principle that you shouldn't add subject categories to a taxonomy unless there are documents in the collection that fit into the category. In other words, you do not add terms based on speculation about what documents might exist or ought to exist, but based on what documents you actually have. It is the principle of literary warrant that ensures that the taxonomy is shaped by the collection, and thus it is literary warrant that causes a taxonomy to be a useful guide to the collection itself.

Use warrant is a complementary principle to literary warrant, according to which you don't add categories to the taxonomy unless you anticipate that searchers will use them. Together, literary warrant and use warrant set sensible limits on the growth of the indexing language. Literary warrant confines the language to the documents actually present in the collection, and use warrant limits it to subjects actually of interest to users. The two principles are sometimes seen as being in conflict, but they are better considered to be complementary.

A somewhat related issue, however, does bring the document and the user into conflict. Once you have decided to add a concept to an indexing language, you sometimes need to decide whether to follow the documents or the usage on the way the idea is to be represented. In theory, you should prefer to follow the usage, but most indexing systems tend to follow the language of the documents instead for the practical reason that it is easier to see what the document said than to try to imagine what future users will be thinking when they use the system.

Since terminology tends to be set when the first documents on a subject are added to the collection, the potential difference between the way an idea was originally represented in documents and the way users think about now it tends to increase over time. Indexing languages that are more than a few years old tend to be perceived as using old-fashioned language. For example, the law of employment relations is currently covered in the West Key Number System, which dates from 1907, by the heading *Master and Servant*.

The Structure of a Taxonomy

Creating a taxonomy is a tedious business. Parts of the process seem to have a good deal of appeal, and you might find that there are more people who want to be involved in creating the system than are strictly required or even desirable. But in most cases, the issues that appeal to the taxonomic urges of most of the system's users are a few high-level structural issues, and interest is likely to flag as the design reaches down to the detail of choosing individual terms.

The broad, structural issues that seem to attract attention are these: (1) how many lines should the classification have; (2) how deep or flat should the classification be; and (3) what should be the classifications at the top level or two.

The number of lines in a taxonomy

Perhaps the most apparent structural aspect of a taxonomy is its depth, which is generally measured by its number of lines. The more lines the taxonomy has, the finer distinctions it makes among the documents it contains, and the more elaborate the representation of the subject matter.

There is a temptation to think that in taxonomies more is better, but this is a temptation that should be resisted. One of the purposes of the taxonomy is to bring similar things together, but if the classification system is too finely articulated, it will end up having the opposite effect. If you have 10,000 documents rattling around in a classification system that has 100,000 lines, the tendency is for the system to separate documents from others that have similar subject matter rather than to bring them together.

A good rule of thumb is not to create a subject heading for any topic unless you expect it to have at least 10 documents in it. When a classification acquires more than 30 documents, it is reasonable to start thinking about whether there is a sensible way to subdivide it. The general goal is to have each classification yield a list of documents that is short enough that users are willing to examine it, but not so short that documents are isolated from the other documents with which they have most in common. Thus, as a starting point, consider creating a classification system that contains about one-twentieth as many lines as you expect to have documents to put in it.

This rule of thumb is subject to many conditions and exceptions, and is often best honored in the breach. Here are some considerations that militate in favor of making exceptions.

First, the ten-item rule should be violated whenever you identify a subject that has nothing usefully in common with other subjects. Your goal is to bring similar documents together, so if you have a document so different from others that it's not useful to see it in the company of even the documents most similar to it, there is no reason to lump it in with others just to get the count up to 10. On the other hand, if you have a category with hundreds of documents in it, you should subdivide the category only if you can think of a useful distinction to make among the documents in the group. In almost every document retrieval system, there are some topics that are so common that they attract hundreds of postings, and yet the documents in those categories are so similar that it makes no sense to separate them.

Some systems are designed so that only part of the search logic is intended to be represented by the classification system. If your system allows the user to combine free text search logic with the controlled vocabulary represented by the index, it is sensible to allow individual categories to contain hundreds and perhaps even thousands of documents. For this kind of system, the important thing is to ensure that the classification system makes all of the distinctions that are difficult to make with free text searches. It is over-fine if it subdivides categories along lines that are readily obtainable by adding a word or two of free text to the query. Since such hybrid systems are increasingly common, it is more and more usual to find classification systems with many fewer than one-twentieth of the number of items they describe.

Small, medium, and large indexing languages

There are certain numbers that seem to limit the size of certain kinds of taxonomy. Regardless of how many items there are to describe, or how distinct the items are from each other, usage patterns tend to draw indexing languages into three general categories, which may be thought of as small, medium, and large.

Small classification systems are those that are intended to be browsable, but not to have any hierarchy. What you want is generally to have the maximum number of categories that can be presented to the user as a single choice. That number seems to be something in the neighborhood of 30 to 40. If the number is much smaller, the system tends to make groups that are so abstract that the user may have a hard time understanding what each contains. If it is much more than this, the list of choices is too long to show on a single screen, or sometimes even on several screens, and the user misses the most helpful documents because they cannot hold all of the relevant choices in mind at once.

Medium classification systems gravitate toward indexing languages on the order of 1000 items. Here the limitation is more a matter of the behavior of the indexer than that of the searcher. The experience of many indexing agencies seems to be that 1000 is about the maximum number of distinct subject categories that can be understood and consistently applied by a single indexer. If there are more than 1000 categories, either the quality of the indexing begins to deteriorate or the indexing effort needs to be distributed among specialized subject experts, which greatly increases the organizational burden of running the system.

Because this medium level seems to be derived from the behavior of indexers rather than that of searchers, it is less relevant in systems that do not depend on human indexing, but those systems have their own limitations, as will be discussed below under the heading *automatic indexing*.

Large systems seem to encounter a limit to growth at about 100,000 items. There are relatively few indexing systems that have grown so large as to encounter these effects, but most of the largest systems in use seem to grow more slowly once they approach six figures. This limit seems to affect both systems with a hierarchical structure like the West Key Number System and flatter systems like the Library of Congress Subject Headings. Perhaps something on the order of 100,000 is simply the limit of comprehension of a system even when it is used by dedicated specialists.

How many levels should a taxonomy have?

The feature that distinguishes a browsable taxonomy from a flat indexing language is that it is hierarchical. The subject headings are divided into tiers, with certain categories being embedded as sub-categories of more general ideas.

A rough rule of thumb for the number of levels an indexing language ought to have is that it ought to have about the common logarithm of the number of lines it contains. To put the rule in more familiar terms, this means that a language of 10 lines needs only 1 level, a language of 100 lines needs 2, 1000 lines need 3 levels, and so forth.

Many database systems require the user to set a strict limit on the number of levels in the hierarchy of a taxonomy when the system is first configured, and it is generally wise to specify one or two levels more than you intend to use.

The factors that inform the proper depth of a taxonomy are related to the user. There should be enough levels so that at any particular decision point the user is confronted with a manageably small number of choices. The number of choices at lower levels of the system should be smaller than the 30 to 40 allowed at the first level of the taxonomy. It's best to try to limit the number of choices to about ten, though that is not always possible.

The maximum number of levels should be the minimum necessary to show all of the lines with a reasonable number of choices at each level. Each additional level in the scheme is one more decision and one more click between the user and the information she is looking for, so it is best to keep these to a minimum. Different parts of the taxonomy are likely to have different depth because they cover more or less material—it is not a good idea to force the entire system to the maximum number of levels required for the most finely articulated part of the taxonomy.

What should the top levels be?

The third question is the one that most engages the most attention from users. The answer depends very much on the particularities of the collection. Creating a list of the thinkable thoughts in a collection sometimes takes on the aspect of a parlor game. It allows knowledge workers who spend most of their time deeply immersed in the particularities of their own specialties to step back from the welter of detail and try to describe the big picture.

The reason the task is so engaging is that it not only requires the language designer to identify the thoughts of interest, but also to decide which ones are more basic than others. Nearly any legal idea has several aspects to it—the legal theory involved, the jurisdiction, the nature of the parties, the procedural posture, and so forth. In a browsable taxonomy the ideas are ordered in such a way that the most basic or general ideas are the principles of classification for the higher levels, and the more particular ideas determine classification for the lower levels. Setting up the high levels of a taxonomy is an exercise in deciding what is important.

This is so satisfying to many people that the system designer is likely to have all the help she needs (and then some) with this task. And, since the main subdivisions need to be established before it is sensible to spend a lot of time on the subsidiary detail, it may be necessary to resolve these questions well before all of the interested parties have had a chance to resolve their different theories.

In the case of archives of legal information, two main ways to selecting the principal subject heads suggest themselves. One is to choose topics that tend to correspond to law school courses—the thought being that lawyers using the system can almost always recognize which of their law school courses the matter at hand would be treated in. The other main approach is to make the main heads the same as the different practice areas in the firm.

Filling Out the Outline

Once the basic structure of the taxonomy is set and the main subject headings selected, the hard work of filling out the detail begins. This is best approached through an iterative process. You begin by making your best guess as to what topics should be included, and then you classify some documents to see how the outline works. You adjust the classification system to take into account the problems encountered in your initial

trial run, and then you do some more trial classification, which yields other, often more subtle, problems. After each iteration, the outline becomes a better reflection of the collection it serves, but in almost all cases you run out of time to do the exploratory trial classification well before you have learned everything you would like to know about how the collection could best be described.

As a general rule, it is better, when you are in doubt about the need to subdivide a topic, to begin with a more detailed taxonomy. If after trial classification you find that the heading does not receive enough postings to justify the distinction, it is easy to collapse it into a broader category.

Adapting an Existing Taxonomy

Creating a new taxonomy from scratch is such a difficult and labor-intensive process that many system designers borrow other taxonomies instead. This is a sensible step for many, but it is subject to a number of concerns that become apparent when you consider them in relation the process of creating a new taxonomy.

You should expect to pay something for the use of an elaborate and well constructed taxonomy. Taxonomies are the intellectual property of their creators. Some organizations will not license the use of their taxonomies, and those that are willing to license their taxonomies often will do so only as part of a complete document organization system.

Bearing in mind the principal of literary warrant, it is important to start with a taxonomy that serves a collection as similar as possible to the collection that you need to serve. Consider, for example, building a collection that organizes the briefs and memoranda of a law firm. If you choose, as a model, a taxonomy that was built to serve a collection of published cases, you are likely to find that it serves the documents generated by the litigation department reasonably well, though it is likely to be much better developed with respect to the kind of issues that are frequently decided on appeal than those that tend to be confined to trial courts. But it is also likely that the system will not serve the needs of the firm's transactional attorneys well. Those lawyers are busy creating document intended to avoid litigation.

An existing taxonomy will probably need some pruning before it is applied to your collection. If your firm does essentially no domestic relations work, there is no point in bringing an elaborate analysis of family law over into your system.

Even after the obvious topics have been eliminated, you are likely to find that the borrowed taxonomy contains many categories that have no documents in them. It makes the use of a borrowed taxonomy much less frustrating to the user if it is set up so that the number of document in each category is displayed on the screens on which the user makes choices.

Finally, you are very likely to want to add categories in certain areas. Your collection is likely to be unusually rich in certain subject areas, and you want to be able to add a more finely articulated analysis to those parts of the outline.

Indexing, Manual and Automatic

Indexing is difficult and labor intensive process. It takes time and money to do it well, and so it is tempting to avoid it as much as possible. But again, before considering the options for automatic indexing, it is well to think about the quality of indexing in general and about indexing as a manual process.

Indexing Performance

Document retrieval systems aspire to provide the user with *all* and *only* the documents they will find relevant to the task at hand. The *all* and *only* specifications are generally in conflict with each other—a system that is good at retrieving everything relevant will almost always retrieve much that is not relevant, and a system that is selective enough to avoid presenting the user with irrelevant documents usually overlooks a large proportion of the relevant documents as well.

Two performance statistics, *precision* and *recall*, are generally used to describe the quality of document retrieval systems. Precision is the proportion of documents actually returned by the system that are relevant. If, in response to a query, the system returns six relevant document and six irrelevant documents, precision is 50%.

Recall is the proportion of the total number of documents that the user would find relevant that are actually retrieved by the system. If a system contains 24 relevant documents and actually returns six, recall is 25%.

Precision and recall are useful concepts in discussing the implementation of document retrieval systems, but claims about the absolute performance of a system in terms of precision and recall (particularly recall) should be taken with a grain of salt. Valid document retrieval experiments are notoriously difficult to conduct, and when you encounter numerical claims by the creators of systems, you can be fairly sure that all uncertainties have been resolved in the direction of making the system look good. Claims of recall exceeding 65% at any reasonable level of precision are particularly suspect.

Of the two statistics, precision is much easier to measure than recall. One can look at any retrieval set and make a judgment about which document belong in the set and which do not, thus estimating precision for that single use. But it is almost never possible to be confident that you know, in any realistically large collection, what relevant documents were not retrieved by a query. The other relevant documents you are aware of are almost invariably a subset of all documents that you would find relevant, so there is a strong tendency to overestimate recall.

Because users have a better notion of the performance of a system with respect to precision than with respect to recall, their perception of the quality of the system is influenced more by the system's precision than by its recall, even for tasks in which recall is more important than precision.

Manual Indexing

In a knowledge management context, it is often assumed that documents should be indexed by their authors. As part of the document-creation process, authors are expected to provide certain meta-data for the document, often including key words or subject categories.

Experience has shown, however, that attempting to depend on authors to index their own work is futile. Indexing is a skill that not everyone has, and while individual authors sometimes do outstanding jobs of indexing, authors collectively cannot be made to supply consistent and accurate subject headings for their work.

If a system uses manual indexing, it is important to have people on staff who are experts in the task. Many, but not all, librarians have formal training in indexing, and even those who did not take indexing courses in library school tend to make good indexers. In general, it seems to be easier for a good indexer to acquire enough

knowledge of the subject matter of the collection than it is for a subject matter expert to become a good indexer.

Automatic Indexing

Most knowledge management systems sold today use some means of automatic indexing to relieve the system administrator from the burden of indexing. Much of the appeal of the system is often vested in the automatic indexing technique applied, and extravagant claims about the performance of the automatic system are common.

The quality of automatic indexing systems is sensitive to the size of the taxonomy it is attempting to classify to. The more elaborate the indexing system in use, the finer the distinctions the system is asked to make, and the more opportunity there is for erroneous classification. It is only in the past few years that fully-automatic systems have been thought capable of handling taxonomies of more than 1000 categories, and some experts do not agree that even the best systems can do an adequate job now.

Nearly all automatic classification systems use proprietary algorithms that are described to potential licensees only in very general terms. While the secrecy surrounding the classification algorithms themselves makes it risky to generalize, there is good reason to believe that most systems use one or more of a few common techniques.

Boolean queries

Some systems work by storing, for each line in the classification system, a fairly straightforward Boolean query that retrieves documents that are likely to belong to that category. These systems are among the least proficient at actually getting the right documents into the right categories, but they do have some signal advantages.

When the system lets the user inspect and, if desired, modify the stored query, the classification process becomes more transparent to the user. The user does not need to take the judgment of the system on faith, but can tinker with the queries to make it more likely to retrieve the needed information.

Boolean systems, however, are essentially free text systems repackaged in taxonomic form, and so one should not expect the system to perform well in finding subjects that are inherently difficult for free text searches.

Category learning systems

A more sophisticated approach to automatic classification is to have a system learn from a group of documents that are known to belong to a category what features of those documents tend to distinguish them from documents that do not belong.

To use a system of this sort, it is necessary to train the system upon a collection of representative documents. A very simple system, for example, might operate by identifying words that occur much more frequently among the documents that belong to the category than in the collection as a whole. Each category in the system is then represented by a list of words characteristic of that category, probably with some numerical weight that reflects how strong each individual word is. Each document is considered with respect to each category, and those that have many of the words that are found in the category's word list are assigned to the classification.

It is often possible to tune the performance of the system by adjusting the threshold used to decide whether to post a particular document to a particular category. If the

threshold is set low, more documents will be assigned, recall will rise, and precision will fall. If the threshold is raised, the opposite occurs.

More sophisticated systems achieve better performance by using more elaborate techniques. For example, the system can consider phrases rather than single words, greatly increasing its ability to correctly classify documents when the operative language cannot be captured by individual words. Other systems apply some simple natural language understanding techniques such as tagging parts of speech and preferring the use of certain word groups.

Still more sophisticated algorithms use several different classification techniques, and make final category assignments based on a weighted average of the judgment of different systems. The most elaborate of these learn which classifiers are most reliable for finding documents for particular categories, and adjust the weights accorded to individual classifiers accordingly.

The better category learning classifiers perform much better than Boolean query systems at correctly classifying documents. They can correctly classify documents even if those documents are missing key words that would be required by the stored Boolean search, and they can often detect, by the absence of supporting evidence, situations in which a word ordinarily very descriptive of a topic happens to be used in some other context.

Still, category learning systems tend to perform poorly in any context in which the important content cannot be reliably associated with a list of words or other document features. They tend to fail, though less dramatically, on the same kinds of tasks that challenge any free text system.

Latent Semantic Indexing

Latent Semantic Indexing (LSI) is a technique that deserves special mention because it automates not only the process of indexing, but also (to some extent) the process of creating the taxonomy. LSI is very popular just now, and lies behind many of the automatic classification systems now on the market, though not all of the systems that use the general technique will call it LSI.

LSI systems begin by clustering documents in the collection. Rather than, as category learning systems do, proceeding from known categories and documents known to belong to those categories, they deduce their own categories on the basis of similarities between documents. The system administrator usually needs to specify in advance how many categories the LSI system is to identify, but after that the system will present the user with clumps of documents, each of which is presumed to represent a significant category.

LSI systems have some substantial advantages, but they also have some characteristic disadvantages. The greatest advantage is that they generate their own subject categories. The categories they create are derived from the collection itself, and so the principle of literary warrant is observed. Finally, the LSI systems tend to perform well in terms of precision and recall because the categories they create are, by necessity, ones that correspond to document differences of the kind that the system is capable of perceiving, so there are fewer categories with conspicuously poor performance.

The disadvantages begin with the introduction of a new step in the system design process. The categories inferred by the system need to be examined and named. The system knows only that the documents in a certain cluster are like each other. Better LSI

systems help the system implementer with the process of naming the clusters by providing lists of the words or expressions that are most characteristic of the cluster, but it is still up to the user to give the cluster a name that will allow browsing searchers to appreciate what kinds of documents may be found there.

Some of the clusters inferred by an LSI system correspond fairly directly to categories that would be created by a human indexer, but a fairly large proportion of the clusters examined by the system designer have problems that need to be resolved. Often, the system will lump together two or more kinds of documents that few users would see as being related. Often also, the system will arbitrarily separate categories that human indexers would want to keep together. Sometimes, a cluster appears to the human editor as little more than an ersatz miscellany, defeating all attempts to give it a reasonable name.

Better LSI systems allow the system administrator to override the categories inferred by the system—separating some and combining others.

Another disadvantage of LSI systems is that they simply won't create categories of the sort that free text systems have difficulty with. It is unlikely, for example, that an LSI system would infer category boundaries for concepts like *chattel*. Thus, while indexing performance benefits from the absence of unclassifiable concepts, the resulting outline, even after a conscientious effort to give descriptive names to the categories and arrange them in a sensible hierarchy, often strikes users as odd.

Finally, LSI doesn't scale very well to systems containing millions of documents and thousands of lines.

As a result, just as manually indexed documents usually appear more appropriate to their categories than automatically assigned documents, manually created taxonomies usually appear to organize the collection better than taxonomies based on groups of documents inferred by LSI.

Semi-automatic or hybrid systems

Document retrieval systems in general reward at least some of the effort used to create them with better performance. A system with a hand-crafted taxonomy populated by documents individually assigned to those categories by thoughtful human indexers will invariably outperform any fully automatic system. Yet few institutions have the money or the leisure to build hand-crafted systems, and those that attempt to do so often achieve the worst possible result, which is to have no working system at all.

Perhaps the best possible system is one in which the system administrator has a suite of fully automatic tools that permit the system to operated with as little human intervention as possible, but also allow the administrator to do as much manual adjustment of the taxonomy as desired, and allows human indexers to do as much manual classification of individual, high value documents as time and resources permit.

Taxonomies in Application

Before reviewing the choices involved in setting up a taxonomic search system, it is well to give some thought to the expectations that users will bring so the system.

User Expectations

As with so much else in the world, the benefit that one can get from a taxonomic system is related to the amount of labor that goes into its creation. What most users would like is a system that, in addition to offering free text retrieval, offers the advantages of a well-constructed manual index. The categories would be thoughtfully chosen to reflect the best and most natural way of thinking about the topics in the collection, and the individual documents would be assigned to those topics by thoughtful subject experts.

The challenge for the system designer is to create the illusion that this has taken place without spending a prohibitive amount of time and money. This is a perilous undertaking. Many users, especially those who are middle-aged and older, learned to find documents by using meticulously hand-crafted manual indexing systems provided by traditional publishing houses. The more a system succeeds at seeming to have been hand-crafted for the support of particular users and tasks, the greater the disappointment the user will feel when she encounters the compromises that are built into the system.

One can make a case for building a system that lowers user expectations by making its compromises and limitations such central features of the user experience that unrealistic expectations are limited. This is either cynical or honest, depending on your point of view. But it is undeniable that it is often more economical to bring performance and expectations into line by lowering expectations rather than by improving performance.

The chief virtue of a system like one in which the indexing is based on stored Boolean queries is that the user can see exactly what it is doing and adjust her expectations and search behavior accordingly. But in most cases, perhaps under the influence of the raft of internet systems that have made this choice, system implementers today usually prefer to implement systems that are opaque to the user.

Summary of the issues

There are two basic decisions to be made in setting up a taxonomic system: where the taxonomy is going to come from, and how documents are going to get assigned to the individual categories.

Choosing a taxonomy

The taxonomy can be created manually, created automatically, or borrowed from someone else. Creating a taxonomy manually is, at least potentially, much the best option, but it is such a labor-intensive activity that, as a practical matter, one runs the risk that the resulting taxonomy will be much too shallow to organize the collection usefully.

Thus, most system designers would like to borrow all or most of their taxonomy from someone else. This is a sensible compromise for many, but it is important to borrow wisely. Because they are created in accord with the principle of literary warrant, they reflect the collections they were initially created for. Thus, in deciding which taxonomy to borrow, the system designer needs to look not only at the craftsmanship with which it was originally created, but at the similarity between the new collection and they original collection it was designed to serve.

Automatically created taxonomies derived from LSI processes are likely to be the least satisfactory, but perhaps the easiest solutions. They derive much of their appeal from the fact that the LSI process also solves the indexing problem. But setting up an LSI system from scratch, including naming the categories and arranging them into a hierarchy, is quite a lot of work, and LSI taxonomies borrowed from other installations sacrifice the benefits of rigorous literary warrant without the concomitant benefit of producing a good scheme.

Indexing

The labor involved in creating a taxonomy from scratch is daunting, but the labor needed to manually classify the documents in the collection is even greater.

One common response has been to try to get authors to index their own work. This makes sense at a certain level—the labor needed to properly classify a document seems small when distributed among the system’s users, and presumably if anyone knows enough about the document to make sure that the right headings are applied, it is the author. But long and painful experience has shown that authors simply cannot be made to do accurate indexing of their own work consistently enough to populate the entire system. There are individual exceptions, and it is often a good idea to make author indexing possible for those who wish to use it, but the system as a whole will need basic indexing from somewhere else.

Manual indexing by a librarian or a professional indexer requires resources proportional to the size of the collection. In many organizations, the cost of manual indexing is reasonable on a go-forward basis, but the difficulty of indexing a huge backfile deters its consideration. Systems that allow selective manual indexing as an adjunct to some automatic system make sense for many applications, especially where the system designer can identify some subset of particularly important documents that merit this special treatment.

Automatic indexing systems need to be tuned to the requirements of the taxonomy in use, and are thus usually sold as part of a package that includes a taxonomy.

Automatic indexing by stored Boolean query is not particularly effective at getting the right documents into the right categories, but it is reasonably easy to set up and maintain, and it is the only one of the automatic indexing systems that ordinary users are likely to understand. This kind of automatic indexing would be appropriate for a system designer with a home-grown taxonomy, or one who wanted to make extensive alterations in someone else’s taxonomy. Its poor performance is less of an objection in a system in which the most important documents are classified manually.

Automatic indexing by category learning works better, but is less transparent to the user. It also requires a substantial amount of work to extend the technique to user-created categories because each category classifier needs to be trained on a substantial number of documents known to belong to the category. Typically, at least a few dozen exemplar documents are needed to train the classifier.

LSI systems typically classify as well or better than category learning systems, but that is because they don’t create categories they can’t classify. A system based on LSI, even with a substantial amount of editorial review and adjustment, is likely to look less polished to the user.

Conclusion

Reaping the benefits of a taxonomic search system is not easy unless one is willing to accept a system that performs, in terms of the categories it uses and the documents assigned to those categories, much less well than the manual systems that many users are familiar with. Building a better system requires careful choices in selecting systems that rewards an increase in the amount of time and money spent in deploying and maintaining a system with the largest and most visible benefit to the user.