Integration of Taxonomy and Keyword Searches: A Comparison of Two Implementations

by Ronald P. Millett

Subject taxonomies have a long history of usefulness in indexing libraries of documents or records. One of the weaknesses of this approach is the difficulty of traversing the taxonomy hierarchy to find the subject term. Full text indexing of keywords enables automatic discovery of records containing terms ordered by relevance including phrase, title text, word proximity, emphasized text, linguistic and other criteria. Major weaknesses of the keyword approach are the lack of the ability to use controlled subject terms and result hits that are false-positive. This paper compares two implementations that combine taxonomy and keyword approaches in an attempt to address these weaknesses for efficiency, ease of use, recall, precision and hit relevance.

NICEM Database Search Implementations

These two search implementations combine taxonomy and keyword searches:

1. A system based on Access Innovation’s Search Harmony product, using their subject taxonomy and user interface with the Perfect Search search engine.
2. A system that is an earlier Access Innovation’s implementation using the Lucene open-source search engine with the same subject taxonomy.

The comparison search data set for both systems uses about 500,000 records from summaries of multimedia videos or presentations in the National Information Center for Educational Media (NICEM) database stored as separate XML files. The subject taxonomy terms were specified with a XML subject category tag. Other important tags include title, abstract and publication date. Table 1 summarizes the sample database and taxonomy information.

The database was indexed by both systems on similar 64 bit Windows servers, with from 3 to 16 GB of RAM. The indexing process was single threaded even though there were several cores available. Table 2 summarizes indexing and retrieval information for each system. The Lucene implementation used the Java version.

<table>
<thead>
<tr>
<th>TABLE 1. NICEM test database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Records (in separate XML files)</td>
</tr>
<tr>
<td>Average record size</td>
</tr>
<tr>
<td>Database size</td>
</tr>
<tr>
<td>Number of subject taxonomy terms</td>
</tr>
<tr>
<td>Average subject terms per record</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE 2. Lucene and Perfect Search Indexing and Retrieval Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lucene</td>
</tr>
<tr>
<td>Indexing Time</td>
</tr>
<tr>
<td>Index Size</td>
</tr>
<tr>
<td>Threading</td>
</tr>
<tr>
<td>Typical Engine Retrieval Time</td>
</tr>
</tbody>
</table>
Retrieval Speed vs. Indexing Cost

One of the apparent anomalies of the Perfect Search system is that the typical inverse relationship between retrieval speed and indexing cost is missing. Faster retrieval speed usually implies that a slow and expensive indexing process needs to be executed. However, Perfect Search’s approach uses an indexing pipeline that creates temporary files and utilizes indexing pattern speedups at every point of the indexing process to improve the indexing functionality. Note that even though the Perfect Search index and accelerators used in retrieval are much bigger than Lucene’s index, these files were generated much faster than the Lucene index. This indexing and retrieval architecture results in both impressive indexing speeds and retrieval speeds for the same system.

Once indexing speeds are a small multiple of file copying speeds, as Perfect Search’s data for this test illustrates, the whole character of creating and using indexes can change. Formerly expensive re-index operations are no longer a problem, if needed. Creation of incremental index files can become almost instantaneous and merging together many small index files can also be an easy part of making a very responsive indexing process to support the search of dynamic data.

Document caching and acceleration files are also generated and included in the numbers in the table for the Perfect Search-based system. The 192 MB acceleration files were generated for the test system in under a minute. On a large system such as the installation for WorldVitalRecords.com, the 149 GB of indexes require acceleration files of only 13 GB (9% of index). 440 MB is not considered a large index for a Perfect Search-based system and the acceleration files are a much bigger percentage of the total index size (192 MB, 43%) than they would be for a larger search system.

These acceleration files play a key role in enabling very fast retrieval speeds for the system. Even when a single query can actually trigger as many as 20 sub-searches for different relevance rankings, a typical retrieval speed of 1,200 to 7,000 queries per second is truly impressive for each complete search. These speeds are for a system where both accelerators and index files have been cached.

Autocompletion of Subject Taxonomy Terms

Using precategorized subject terms, searches can avoid irrelevant hits and find records or documents that directly pertain to the subject category selected. An important weakness of taxonomies is the difficulty of finding the descriptor that the user has in mind if it has to be traversed from the top of a hierarchy. Both test systems use the Access Innovations autocompletion module that uses a “key word in context” (KWIC) list and term matching to allow the user to find any term that he has in mind that might be part of a subject description. Figure 1 shows a sample screen shot of this powerful taxonomy autocompletion window.

Document caching and acceleration files are also generated and included in the numbers in the table for the Perfect Search-based system. The 192 MB acceleration files were generated for the test system in under a minute. On a large system such as the installation for WorldVitalRecords.com, the 149 GB of indexes require acceleration files of only 13 GB (9% of index). 440 MB is not considered a large index for a Perfect Search-based system and the acceleration files are a much bigger percentage of the total index size (192 MB, 43%) than they would be for a larger search system.

These acceleration files play a key role in enabling very fast retrieval speeds for the system. Even when a single query can actually trigger as many as 20 sub-searches for different relevance rankings, a typical retrieval speed of 1,200 to 7,000 queries per second is truly impressive for each complete search. These speeds are for a system where both accelerators and index files have been cached.

Autocompletion of Subject Taxonomy Terms

Using precategorized subject terms, searches can avoid irrelevant hits and find records or documents that directly pertain to the subject category selected. An important weakness of taxonomies is the difficulty of finding the descriptor that the user has in mind if it has to be traversed from the top of a hierarchy. Both test systems use the Access Innovations autocompletion module that uses a “key word in context” (KWIC) list and term matching to allow the user to find any term that he has in mind that might be part of a subject description. Figure 1 shows a sample screen shot of this powerful taxonomy autocompletion window.
the taxonomy structure. Figure 2 shows a screen shot of this taxonomy directory available on the Search Harmony system.

The combination of both top down and bottom up approaches to locating taxonomy terms provides the user with unprecedented flexibility and power to find that carefully crafted subject category for his search.

Precision and Recall Comparisons

Except for one characteristic of the Lucene-based system that lowered precision to 99%, the two systems both receive 100% precision and recall values for the taxonomy subject descriptors. When the Perfect Search stemming module and its effects are added, the recall of the Lucene-based system went down. The Perfect Search algorithmic stemmer was too broad in some cases, which reduced precision for a few specific terms.

Table 3 compares the Lucene and Perfect Search hits for the text “communications” in descriptor and keyword modes. This search was selected because it shows some interesting precision and recall differences.

Note that because the Lucene implementation can get a hit for “communications” as a sub-term of subject taxonomy descriptors such as “communications industry” and “satellite communications,” Lucene has a higher retrieval count on hits for this descriptor entry than Perfect Search. Because the actual descriptor “communications” was not in some hits, the precision value for this search for the Lucene implementation is 99%.

Note also that the Perfect Search system uses a stemming subsystem to include terms like “communication,” “communicate” and “communicating” as lower relevance hits for a keyword search for “communications.” The current implementation uses an algorithmic stemming module that also included hits for “community,” “communities,” “communist” and “communism” as possible word forms for “communications.” This algorithmic stemmer created false positive lower relevance hits. A dictionary-based stemming module being implemented for the next version of Perfect Search will avoid this problem and not include the extraneous forms.

When correctly stemmed terms are included for this keyword search, Lucene then has its recall percentage for “communications” go down to 22% while Perfect Search maintains a 100% recall percentage. When the false positive stems are counted, Perfect Search has its precision percentage go down to 70%. Lucene does have a similar algorithmic stemming module available, but it was not used in this implementation [4, pp. 282-284].
Relevance Algorithms

The descriptors receive a relevance score of 100 for the Search Harmony Perfect Search-based system and are ordered according to the order they were indexed. Keyword searches can have some false positive low relevance hits due to the stemming algorithm currently in use as previously mentioned. Because of the ambiguity inherent in all languages, the use of a word in a search may not match the query’s intended meaning. We did not evaluate that factor in our search precision numbers. Having exact phrase and near proximity in compact fields like the title helps to have more relevant hits and avoid false positive hits.

Combinations of the title, abstract and other fields plus term proximity (exact phrase, near content word proximity) and term linguistic stemming provide relevance scoring of hits from scores of 100 to 50. These orderings and values can be customized according to the users’ preferences using XML configuration file information.

The Lucene-based system also has calculated relevance for keyword searches. Boosts are available for low frequency terms, special fields and normalized term weights [4, pp.78-81] [5].

Mixing and Matching Taxonomy and Keyword Terms

The Lucene-based search system has two modes of searching: preferred terms (taxonomy subject categories) and keywords. The two modes cannot be mixed in the same query. The preferred terms are matched using the autocompletion typing window.

The Perfect Search-based search system allows the user to mix and match preferred taxonomy descriptors and keywords and use a similar syntax to Lucene. Each subject taxonomy descriptor is indexed as a single complex term that is not divisible (e.g., SC:“children’s literature”). New searches and refinement of existing searches can select from the taxonomy terms or the keyword terms.

Figures 3 and 4 illustrate multiple search terms using taxonomy descriptors and keywords in the same search using the Search Harmony Perfect Search-based system. Tables 4 and 5 show the hit frequency for each search term and each stage of the search intersection process.

If, for example, a single term retrieves 1000 hits, the user cannot and will not in normal practice look through the hits to find the right hit. Typically, unless the hit is in the first few pages of results, it might as well be invisible as far as the user is concerned. Precision and recall numbers for large sets might as well not exist when the set cannot reasonably be examined. The power of having both carefully defined descriptors and full text terms in the same query results in a smaller search result set that can easily be viewed and examined.

In Table 4 and Figure 3 we see a search where, even after two subject-descriptor terms were entered, there were still over 1,000 hits in the results list. The keywords “lion” and “witch” also occurred in many database entries, not divisible (e.g., SC:“children’s literature”). New searches and refinement of existing searches can select from the taxonomy terms or the keyword terms.

Figures 3 and 4 illustrate multiple search terms using taxonomy descriptors and keywords in the same search using the Search Harmony Perfect Search-based system. Tables 4 and 5 show the hit frequency for each search term and each stage of the search intersection process.

If, for example, a single term retrieves 1000 hits, the user cannot and will not in normal practice look through the hits to find the right hit. Typically, unless the hit is in the first few pages of results, it might as well be invisible as far as the user is concerned. Precision and recall numbers for large sets might as well not exist when the set cannot reasonably be examined. The power of having both carefully defined descriptors and full text terms in the same query results in a smaller search result set that can easily be viewed and examined.

In Table 4 and Figure 3 we see a search where, even after two subject-descriptor terms were entered, there were still over 1,000 hits in the results list. The keywords “lion” and “witch” also occurred in many database entries.
but after the descriptors filtered the set down to 1,078, the two terms resulted in a short list that contained two copies of an audio book of the C. S. Lewis’ work *The Lion, The Witch and the Wardrobe*.

Table 5 and Figure 4 illustrate another search where a single category descriptor “business planning” was combined with keywords “computers” and “forecasting.”

### More Relevant Results through Tapping Multiple Dimensions of Search

The key to success in a search system is to help users quickly find the data they are looking for. The ambiguity of language, syntactic and UI details and the amount of data that a search system has to winnow out to find key results for the users are formidable obstacles for any system. The advantages of including taxonomy-based subject descriptor searches and keyword searches in the same system were emphasized by searches on both test systems. The lack of being able to mix and match descriptor and keyword searches in the Lucene implementation made it much more difficult to retrieve a result list that can easily be examined by the user.

The Search Harmony system combines several search methodologies in one unified system, including mixing and matching descriptor and keyword search terms. These methodologies include navigational trees, subject descriptor autocompletion, search within results and relevance-based full text searches including fuzzy search features such as stemming. Combining all of these features in a single interface, each as it were in a different search dimension, provides an important laboratory for improving search results and finding that “Voila!” best result.

### Resources Mentioned in the Article


