Information Space based on HTML Structure

Gregory B. Newby[*]
UNC Chapel Hill

Abstract

The main goal for the Information Space system for TREC9 was early precision. To facilitate this, an emphasis was placed on seeking matches from only the TITLE, H1, H2 and H3 tags in the Web (wt10G) and large Web (wt100) document collections. Ranking of documents was based on a combination of Boolean union sets, term weights, and principal components analysis (PCA). Very large sparse cooccurrence matrices were created for term weighting and PCA. The Information Space system is part of a larger general software package called IRTools.

Introduction

This year's TREC entry for the Information Space system builds on past years, with some specific goals. Due to 2000 being the first year with Web data for the main task for TREC (instead of newswire and other data, as in past years), it seemed desirable to make use of the structure of HTML. As casual observation of the popular Web search engines (Google, Lycos, etc.) reveals, these systems provide additional weight to terms occurring in the <TITLE> tags of documents, in addition to searching through the terms in each document.

The Information Space (IS) main Web task entry for this year focused only on tags in the <TITLE>, <H1>, <H2>, and <H3> tags in the datasets. This was intended to facilitate early precision, by matching the short TREC topic title or title plus description statements to terms in these tags. The submission for the main Web task was 6 days late, and therefore not judged (although it was counted by NIST as an "official" run). Post hoc analysis of some queries indicate that if results were judged, they probably would not have been substantially better than the non-judged results found in the conference proceedings.

IS also made an entry to the large Web task or VLC. The 100GB VLC (w100) was processed similarly to the main Web task, by focusing only on terms in the same set of tags (title, h1, h2 and h3). Because this run was also submitted late, by nearly 2 weeks, it was not judged. Due to the small number of official VLC submissions and small number of judged documents, no useful recall or precision statistics are available.

This paper will present an overview of the procedure used to index and retrieve from the wt10g and w100 datasets, followed by a brief discussion of the large co-occurrence matrices generated. Then, system-based and relevance-based performance outcomes are discussed. It is concluded that query expansion did not serve well to facilitate early high precision. Furthermore, a lack of sophisticated term weighting also hurt results.

[*] Contact data: gbnewby@ils.unc.edu, http://ils.unc.edu/gbnewby. CB 3360 Manning Hall, Chapel Hill, NC, 27599-3360 USA.

The IRTools Software

IS is part of a set of software tools for IR experimentation under development by the author and his colleagues. The software is called the "Information Retrieval Toolkit," or IRTools. The purpose of IRTools is twofold:

1. To provide an integrated collection of C++ classes designed to facilitate IR experimentation; and
2. To incorporate design for large-scale practical use.

Although modern information scientists have always relied on software for their experiments, relatively few have chosen to make their software freely available to others. For those who have shared, the software is often not suitable for re-use in other experimental settings – due to either lack of documentation, cross-platform instability, or non-modular design. IRTools is intended to help address the shortage of software for retrieval experimentation.

Another problem that has often hindered information scientists is the difficulty of demonstrating the scalability of their ideas. IRTools places and emphasis on high performance data structures, file structures and algorithms (Newby, 2000b). Real-world functionality will include the ability to update the document collection (e.g., by spidering the Web periodically). IRTools' goal is to index billions of documents, with hundreds of millions of unique terms, and over a terabyte of aggregated data.

IRTools is designed modularly, as a library of C++ classes. Currently, IRTools is over 25,000 lines of code including test programs. It makes extensive use of the standard template library (STL). The plan for IRTools is to incorporate the functionality of all major types of experimental IR: probabilistic retrieval, the vector space model, latent semantic indexing, simple Boolean retrieval, and others. IRTools will make it easier and faster for information scientists to perform experiments or expand software. The software development is supported in part by a grant from the NSF under their information technology and research (ITR) program. The project homepage is http://irtools.sourceforge.net.

Information Space Techniques for TREC9

Information Space, or IS, is an approach to information retrieval that is similar to latent semantic indexing (LSI). Over the past several years, IS has incorporated different specific techniques to achieve particular goals. IRTools will enable more of these goals to be integrated – for example, the TREC9 IS programs did not have good facilities for term weighting, even though the utility of term weighting using IS techniques was demonstrated in TREC8 (Newby, 2000a; Newby 1998).

The main distinction between LSI and IS is that LSI utilizes a singular value decomposition (SVD) on the term by document matrix, while IS utilizes principal components analysis (PCA) on the term by term matrix. In both LSI and IS, the distinguishing point from the vector space model (VSM) is that terms are not assumed to be mutually unrelated. The basic process is the same, however: document vectors are computed based on the vectors for terms they contain. A query vector is similarly computed, and the closest documents to the query are retrieved.

Although LSI and IS are comparable, and have a similar intellectual heritage in the mathematics of linear algebra, they actually operationalize a significantly different goal. With both LSI and IS, only *k* columns of the eigenvectors from the SVD or PCA process are used, rather than all *N* columns for each of the *N* terms. With LSI, all columns of the eigenvectors would in fact result in a vector space in which all terms are mutually orthogonal – in other words, the same fundamental model of the VSM. Thus, the *k*-dimensional vector space representing term relations in LSI is an approximation of an orthogonal term space. By reducing *k*, LSI attempts to account for assumed "errors" in the original term by document matrix.

With IS, all *N* columns of the eigenvectors would result in a vector space in which term relations are identically scaled to the numeric relations among terms in the original term by term input co-occurrence matrix. Thus, the *k*-dimensional vector space representing term relations in IS is an approximation of the relations among terms actually measured in the term by term matrix.

These differences are moderated by the other differences in how the techniques are actually applied. For most purposes, it is accurate to characterize IS as similar to LSI. The author has written a more extensive treatment of this subject which has been submitted elsewhere for publication.

The specific techniques used for both the main Web and VLC in TREC9 are as follows:

Phase 1: Indexing

1. Only terms in the <TITLE>, <H1>, <H2> and <H3> tags were processed. All terms in other tags were ignored, as was any document metadata for the wt10g or w100 collections. Documents without these tags were ignored.
2. All terms with fewer than 20 characters and consisting only of alphabetical characters A-Z (case insensitive) were indexed. No stemming was applied.
3. A term by term co-occurrence matrix was built for all the indexed terms for all the documents they occurred in. This resulted in a very large and very sparse matrix.

Phase 2: Retrieval

1. Only terms that had been indexed were used; others were stopped. In addition, the SMART stoplist was employed, along with a few additional stop words consisting of HTML tags.
2. Query terms were expanded (by 100 terms for wt10g, and 25 terms for w100). The top co-occurring terms for each query term were added to the query.

3. All documents with any of the expanded query terms were selected for further consideration; the rest of the documents were assumed to be non-relevant.
4. The full (sparse) co-occurrence matrix for all of the expanded query terms was used to calculate the full (dense) correlation matrix for the terms.
5. PCA was performed on this correlation matrix:
    a. The eigenvectors of the correlation matrix were computed
    b. Term vectors were computed as the dot product of that term's eigenvector and the terms standardized (*z*) scores from the original co-occurrence matrix.
6. Each document under consideration was located at the geometric center of the expanded query terms it contained (terms it contained that were not part of the expanded query were ignored).
7. The query was located at the geometric center of its terms.
8. The query and document locations were normalized to unit length.
9. Distances from each document to the query were ranked, and the closest retrieved.

Note that the choice of the geometric distance versus cosine is arbitrary for unit length vectors: the ranking is the same. But for non-uniform vector lengths, the geometric distance is more accurate than the cosine, as the cosine only considers the angle of incidence between vectors, not the difference.

Large Co-Occurrence Matrices

A difficulty of working with co-occurrence matrices with large numbers of terms is that the number of updates to the matrix during indexing can be daunting. Consider that for a document with 1000 terms, (1000 x 1000-1)/2 or 499500 term pairs exist, and must be considered for updating the term by term co-occurrence matrix. Even if term ordering or term counts are ignored, the number of possible term pairs per document can be large.

One approach to avoiding a very large number of term pairs for each document is to consider co-occurrence only within subdocuments (this is also conceptually appealing). A subdocument might be considered as a term plus its surrounding terms (a sliding window), terms within the same paragraph, or terms within the same sentence. Another obvious approach, employed by IS for TREC9, is to only consider terms within the same tag set. Here, the co-occurrence matrix was computed based only on terms that were found together within a title, h1, h2 or h3 tag.

This resulted in a manageable number of term pairs for most documents, as HTML titles and h1, h2 and h3 tags tend to contain fewer than a dozen terms. This also added to the sparsity of the matrix, which helps with storage. Were every cell in a term by term matrix to be filled, the storage size on disk would be N times N (for N terms) times the size of each datum stored. For the 1.2M unique terms identified in the w100 collection and 4 bytes per integer, this is well over 5 terabytes.

Using a variation on the Harwell-Boeing sparse matrix format, IS only stored the non-zero cells on disk. The storage required using the IS variation on the H-B format is:

$$S(3*N + 2(C) + 2(N))$$

where:

S is the number of bytes per integer
N is the number of terms (aka rows)
C is the total number of non-zero column entries

Using this format with the number of non-zero co-occurrence scores reported in Table 1, about 304Mbytes were required to store values for the co-occurrence matrix for the 1.2M unique terms from w100, a savings of well over 99%. In fact, this is nearly twice as much storage would be required to store only ½ of the matrix with no loss of information, as the matrix is symmetric. Both sides were used during the retrieval phase described above, so the symmetric matrix was converted to a full matrix after indexing was completed.

Table 1: Term co-occurrence matrix properties

Dataset	Term count	Non-zero co-occurrence scores	Sparsity
wt10g	310050	27233214	0.00028329
w100	1207560	34982212	0.00002399

Indexing and Retrieval System-Based Performance Measures

For TREC8, IS was able to index w100 in 5 hours, and process all 10K VLC queries in about 52 seconds. The TREC9 implementation did not strive for such high system performance measures: term co-occurrence added significantly to the indexing overhead, as did identification of tag sets within documents. Indexing time for the w100 was about 120 hours; the wt10g took about 20 hours.

As for TREC8, all indexing and retrieval was completed on UNC's Sun Enterprise Server 10000, a high-end server that was shared with many other processes. The ES10000 had 36 processors and 20GB of memory, but IS utilized only one processor at a time and operated in less than 2GB of memory. A high-speed disk subsystem with a tape-to-disk robot enabled virtually unlimited storage with latency of less than a minute for staging the files to be indexed.

Retrieval for the wt10g took well under .1 seconds per query. Query processing involved minimal disk access: the key to the inverted index was read into memory, as was the term hash and full co-occurrence matrix. Disk access was needed to get inverted index entries (that is, the list of documents containing each expanded term) and to map document ID numbers to TREC document strings.

For the w100, retrieval time depended on what sort of query expansion was used. When simple query expansion by 25 terms was used, as described above, queries were completed in an average of .21 seconds across the 10K topic statements. A more sophisticated query expansion model was attempted, in which several iterations and permutations on the co-occurrence matrix were made. The retrieval performance for this variation is not known, because the w100 runs were not judged, but the system performance of over 9 seconds per query is not favorable.

Table 2: System performance for indexing and retrieval

Build index	wt10g	20 hours	
	w100	120 hours	
Index size	wt10g	.58GB	
	w100	2.7GB	
Retrieval time	wt10g	.1 sec/query	
	w100	.21 sec/query	method 1: simple expansion
	w100	9.7 sec/query	method 2: complicated expansion

Retrieval Performance

Because the results for wt10g were not judged, there is some risk of bias in interpretation of the TREC performance measures. However, an informal evaluation of non-judged documents for a set of 6 topics gave the author some confidence that the retrieval performance measures are reasonably indicative of IS' performance in TREC9.

Because there are essentially no judgments for the VLC that are useful for evaluating the w100 submission discussed above, no retrieval performance measures can be discussed here.

For the main Web task, recall from above that the main goal for this year's work was to have high early precision by utilizing the structure of HTML documents. The reasoning was that terms in the title, h1, h2 and h3 tags were most indicative of a document's content. Thus, indexing and retrieval focused on terms in those tags.

In hindsight, it was poor judgment to apply query expansion. In reading through highly-ranked documents, many documents had expanded terms but no query terms. More effective term weighting would have helped avoid this problem, although computation of term weights was hindered by the particular file structures employed (because counts of the frequency of term occurrences were not kept at a document level, only a tag level).

A better approach would have been to bypass the use of the co-occurrence matrix entirely in order to develop baseline retrieval performance. In other words, to perform simple ranked Boolean retrieval based only on terms occurring in the targeted tag sets. Although this would have resulted in several TREC9 topics with no results, a far larger dataset (either w100 or, more interestingly, the Web as a whole), presumably would have produced results for all 50 topic statements.

A challenge in seeking strong retrieval performance combined with strong system-based performance measures is the conflict in the number of documents that can be evaluated. Conceptually, IS (like LSI) would like to evaluate the relationship between every single document in the collection and a query. This is because the IS technique (like LSI) enables matching based on concepts even when terms do not match. However, for practical purposes this is not feasible: evaluating all 18M w100 documents would take too long.

There may be a solution to managing the size of the problem for computing all possible document relations, as discussed in the author's submission to the TREC8 proceedings. But in the meantime, the time-tested approach for IR is to only consider the subset of documents that contain terms of interest – either the query terms themselves, or the query terms plus expanded terms.

Based on the previous paragraphs, the IS system was implemented to evaluated a larger subset of documents than would be evaluated based on a simple Boolean matching of query terms, but far smaller than the complete document set. This is a goal consistent with traditional goals of the IS approach, but (again, in hindsight) probably not a good match for efforts at high early precision based on a limited number of HTML tags.

The specifics of retrieval performance are as follows. For wt10g, four variations on the steps described above were submitted:

1. iswt: title-only
2. iswtd: title + description
3. iswtdn: title + description + narrative
4. isnnwt: title + description + narrative, but with "not" or "non-relevant" phrases automatically removed

Retrieval performance for all four sets was not outstanding. Table 3 shows that the overall number of relevant documents retrieved @ 1000 is fairly low, with under 10% of relevant documents identified by any set. Intuitively, this would be the retrieval performance statistic most likely to be hurt by non-judged sets.

Table 3: Relevant retrieved @ 1000

	iswt	iswtd	iswtdn	isnnwt
Best	1	2	2	1
>= Median	3	3	4	1
Worst	12	11	13	24
Total relevant retrieved	242	236	172	126
% total relevant retrieved	9.25%	9.02%	6.57%	4.81%

Retrieval performance based on average precision tells approximately the same tale. IS tended to have scores above the median when the median scores were relatively low, without ever achieving average precision over 0.33.

Table 4: Average precision

	iswt	iswtd	iswtdn	isnnwt
Best	1	2	2	1
>= Median	8	7	7	5
Worst	13	12	12	24

What of early precision? Precision at 10 docs (P@10) across the 4 sets was not as high as hoped. None of the sets achieved perfect precision at 5 or 10 documents. Fewer than ½ of the queries for all sets resulted in any relevant documents at all in the top 10, which is disappointing. However, as shown in Table 5, these were numerous queries with numbers of relevant documents in the top 5 or 10 documents presented.

Table 5: Precision at 5 and 10 documents

P@5 score	iswt	iswtd	iswtdn	Isnnwt
0.8	0	1	0	0
0.6	1	2	3	0
0.4	4	7	3	6
0.2	12	13	16	11

P@10 score	iswt	iswtd	iswtdn	isnnwt
0.5	0	1	0	0
0.4	2	1	2	0
0.3	4	10	4	2
0.2	5	1	4	4
0.1	11	16	14	13

The main trends evident from examining the TREC9 topics and IS retrieval performance are variability in the HTML document use of tags, and failure of query expansion. Variability is, as mentioned above, perhaps less of a problem in a larger dataset (w100 or the whole Web). Exact matches of title or title + description terms were fairly rare. Furthermore, more effective retrieval would necessitate additional examination of the terms within the documents, not only the four tags used here.

From this result, we tentatively conclude that better retrieval from HTML documents would involve multiple phases or ranking schemes. At one level, documents with matching <TITLE> or other key HTML tags should be given high consideration. At another level, more typical IR techniques should be employed in order to identify potentially useful documents that do not have the query terms in the <TITLE> or other targeted tags. Then, ranking schemes need to be developed to assess which documents from these two sets of candidates are best for retrieval.

For query expansion, as mentioned above, the danger is in retrieving documents on unrelated topics due to the variability in human language. There is little reason to doubt

the general utility of query expansion based on the results here, and in fact prior IS entries to TREC have discussed the utility of the term correlation matrix for identifying synonyms.

For query expansion, we suggest that relatively inexpensive approaches, such as the co-occurrence matrix applied here, must be used with caution. More expensive approaches would, presumably, result in fewer ambiguous terms being used – such approaches might be applied at the indexing phase, the query phase, or the document ranking phase. Approaches could include dictionary lookups of term meanings and relations, more detailed statistical analysis (including LSI), and part of speech tagging. In fact, all three approaches and other variations have been used by IS in the past, and will be incorporated for further experimentation in IRTools.

Conclusion

Early precision was not achieved to the extent hoped for. The main problems were query expansion, which added some inappropriate terms to some topic statements, and reliance on only the <TITLE>, <H1>, <H2> and <H3> tags. For future work, terms from other tags will be included in the index, and query expansion will be employed more selectively.

Continued development of IRTools and the IS techniques it contains is anticipated to make it easier to incorporate multiple techniques without a large investment in programming time. A comparison of the relative contributions of the effects of such factors as stemming, PCA and LSI techniques, query expansion, term weighting and other approaches is needed to assess the situations in which each technique is most important for high precision or other goals.

References

Newby, Gregory B. 2000a. "Moving More Quickly Towards Full Term Relations in Information Space." Text REtrieval Conference (TREC-8) Proceedings. Gaithersburg, MD: National Institute of Science and Technology. November 16-19, 1999.

Newby, Gregory B. 2000b. "The Science of Large-Scale Information Retrieval." Internet Archive 2000 Colloquium. San Francisco, March 8-9.

Newby, Gregory B. 1999. "Information Space Gets Normal." Text REtrieval Conference (TREC-7) Proceedings, pp. 567-571. Gaithersburg, MD: National Institute of Science and Technology. November 9-11, 1998.