Information Processing & Management Vol. 24, No. 3, pp. 243–248, 1988
Printed in Great Britain.

0306-4573/88 $3.00 + .00

GETTING BEYOND BOOLE

WILLIAM S. COOPER
School of Library and Information Studies, University of California, Berkeley, CA 94720, U.S.A.

Abstract—Although most computer-based information search systems in current use employ a Boolean search strategy, there is by no means a clear consensus throughout the information retrieval research community that the conventional Boolean approach is best. The well-known drawbacks of the Boolean design include an inhospitable request formalism, frequent null output and output overload, and lack of provision for differing emphasis on different facets of the search. Nontraditional design principles that overcome these problems are already known and available in the research literature. In this article several such alternative approaches are sketched and their advantages over the Boolean design indicated.

In the 1950s, the era in which serious thought was first given to the possibility of computerized information searching, it was proposed that search requests might advantageously be formulated as Boolean combinations of document descriptors. This suggestion seemed to meet with the immediate approval of most mathematicians, computer scientists, and technically oriented information professionals. At that time only Bar-Hillel, a mathematical logician, objected strenuously [1].

A decade later, when the first large-scale bibliographic retrieval services were set up, the Boolean approach was adopted as the underlying retrieval strategy. Since then it has become the more-or-less standard search mode for almost all the commercial search services and in most automated library catalogs. It is also used in the command languages of many database management systems, office information systems, personnel search systems, and various other information access programs for scholarly, institutional, or personal use. In fact, insofar as search systems in actual operation today are concerned, the Boolean request form is quite ubiquitous. Thus it may come as a surprise to some readers to learn that specialists in information retrieval are by no means unanimous in their praise of the Boolean approach, that the research literature is full of alternative proposals, and that knowledgeable information scientists who think the standard Boolean design could be significantly improved on probably constitute an overwhelming majority.

Admittedly, there is as yet no clear consensus among researchers as to which of the many available non-Boolean designs is best, and this lack of a single clear alternative candidate has doubtless been a factor tending to perpetuate the current monopoly of Boolean systems in the marketplace. Nevertheless, it is not difficult to point to well-researched retrieval strategies that are clearly superior to the Boolean in at least some important respects. This article sketches briefly a few of these possibilities for the benefit of system designers who might not otherwise be aware of them. A commercial implementation of any of them would be a practical advance and might help the information community to break through the Boolean barrier toward some of the more sophisticated designs that are already familiar to information retrieval researchers and experimenters.

Several proposals will be sketched in order of increasing sophistication and decreasing conformity with the conventional Boolean design. Although some of them may be novel in detail, the general principles behind these designs are all to be found in the research literature. The design ideas will be presented in the form of certain problems inherent in the Boolean search logic, and the proposed post-Boolean solutions to these problems.

PROBLEM 1: THE UNFRIENDLINESS OF BOOLEAN FORMULAS

Those who were initially enthusiastic about Boolean retrieval in the 1950s and 1960s were presumably computer people and other mathematically minded folk who already

knew some Boolean algebra or could easily learn it. It is doubtful that many of them were typical potential lay users of retrieval systems, or they would have had more doubts about the suitability of the Boolean request language.

As anyone who has had occasion to teach the Boolean request form knows, at first there is a marked tendency among learners to confuse the Boolean AND with the OR. This is an understandable mixup for English speakers because in ordinary conversation a noun phrase of the form "A AND B" usually refers to *more* entities than would "A" alone, whereas in the information retrieval usage it refers to *fewer* documents than would be retrieved by "A" alone. In training courses for future information professionals, the confusion usually subsides after some on-line search experience, but occasionally it resurfaces; and in any case one would ideally wish for a request language that could be used immediately by naïve users without long explanations or hours of practice.

The AND/OR difficulty is by no means the only tricky aspect of the Boolean language. Even after the learner has the correct meanings of the Boolean operators clearly in mind, he or she must still gain facility in combining them, and this takes practice. There are connective symbols to be memorized, parentheses to be matched up, scope problems to be dealt with, conventions about connective priorities to be grasped, and so forth. Computer professionals, librarians-in-training, and some others may willingly jump these minor hurdles in order to learn a retrieval language, but it is doubtful whether large numbers of less highly motivated individuals would spend the effort required to feel at home with the Boolean formalism if they were not forced to do so by a lack of available alternatives.

Solution: Symbol-free faceted requests

In training sessions on how to use Boolean retrieval systems, it is often suggested that the learner start out each search by writing down a separate list of search terms for each aspect or "facet" of his or her information need [2]. The student is then taught to combine these lists of quasi-synonymous words or phrases, or concept clusters, into the form of a Boolean request by ORing within the lists and ANDing between them. For instance, an information need describable by the three concept clusters (1) A, B, (2) C, and (3) D, E, F, G would get transformed into the faceted request (A OR B) AND C AND (D OR E OR F OR G). Negated facets can also be included if need be, as in the request form (A OR B OR C) AND NOT (D OR E). Whether consciously or unconsciously, most experienced searchers lean heavily on this approach, and in practice the vast majority of Boolean search requests turn out to be special cases of the faceted request form. One suspects that those who do eventually learn to cope comfortably with the Boolean formalism manage the trick by using faceted requests almost exclusively.

But if the faceted request form is the only Boolean form that is ordinarily used or needed, why should the average searcher be forced to confront Boolean algebra at all? The user need only be given the idea, perhaps with the help of a homely example or two, of how to describe his or her information need by constructing the lists of quasi-synonymous terms. There need be no mention of Boolean connectives or search logic. Once they are constructed and entered into the computer, the user's concept lists can be transformed automatically into the faceted Boolean form needed for the search in a process that is invisible to the user and with which he or she need never be concerned. Any programmer experienced in the design of "friendly" interfaces should be able to provide convenient facilities for entering, editing, modifying, and rearranging such lists of terms. Although still in essence a (restricted) Boolean input language, this nonthreatening protocol could probably be grasped by anyone in a matter of minutes.

PROBLEM 2: NULL OUTPUT AND OUTPUT OVERLOAD

A well-known problem with Boolean systems is that a Boolean search request often results in null retrieval as first formulated. In fact, in conventional bibliographic search systems an empty or too tiny output is typical for requests that AND together more than three or four facets. The user is then forced to reconstruct his or her request, and following the line of least resistance, he or she usually does so by removing one or more of the orig-

inal facets. After sufficient amputation of this sort has been performed on the request a non-null output is generally obtained, but only at a certain cost in time and frustration. Worse, the original sense of the request has been degraded by this process of excision.

On the other side of the coin, there is sometimes far too much output, leaving the user at a loss as to where to start looking through it. In such cases one could wish for some hints from the system as to what parts of the retrieved material are likeliest to be relevant, but in a pure Boolean design no such hints are forthcoming.

Solution: Ranking by coordination level

If the Boolean request has been entered into the system in faceted form, as suggested, the output can be ordered by the number of facets (disjunctive expressions) that are satisfied. Any unit of stored information satisfying all of the request's facets will be in the top rank of the output, any which satisfies all but one will be in the second rank, and so forth until some arbitrary limit on the length of the output has been reached. Under normal circumstances such an output will never be null; and by scanning through it from the top down the user will be led to examine what is probably the most hopeful material first, stopping either when his or her information need is met or when the density of relevant material becomes too low to warrant continuing. This kind of ranking will be in agreement with what is intuitively desirable provided all the request's facets are of approximately equal importance in representing the user's needs.

This general solution to the null-output and output-overload problem has been variously referred to as "coordination-level matching," "overlap ranking," and "vector product" retrieval [2–4]. It represents a distinct departure from pure Boolean logic but is thought by many specialists to produce better results than the traditional Boolean approach.

PROBLEM 3: UNDIFFERENTIATED FACETS

It is often the case that a searcher will feel that some aspects of his or her information need are more important or essential to the search than others. But in conventional Boolean designs there is no way in which the user can communicate this to the system, nor any way for the system to exploit such information to improve the retrieval results.

Solution: Weighted request terms

If (as already recommended) each search request is entered into the system in the form of one or more term lists (facets), the opportunity can easily be provided to any user who wishes to do so to enter a numeric weight along with each list. Larger numeric weights would indicate aspects of the search that have greater subjective significance in the user's mind. Output is ranked by taking into account not only the number of facets satisfied but also their numeric weights. The most straightforward formula is a simple sum-of-weights ranking criterion. (Example: Suppose in the two-facet request (A OR B) AND C the user has given facet A OR B the weight 3 and facet C the weight 5. Then stored records bearing the descriptor C along with either A or B would stand at the top of the output ranking with weight 8; next would come those without A or B but with C with a weight of 5; and finally those with either A or B but not C with a weight of 3.) In some schemes negative facets—that is, facets that in traditional Boolean formulation of the request would have been prefaced by the NOT connective—may be given negative weights.

Many researchers regard weighted-request retrieval as highly promising and a number of such systems have been set up on an experimental basis. However, there has as yet been little experience with the use of weighted systems by large populations of typical users. Consequently, not much is known about how willing or able the average user might be to provide the subjective quantitative judgements demanded by such schemes. Until more experience has been gained, conservative user-interface designs should certainly make the assignment of facet weights optional. There is no problem in doing so, for when a user declines to assign any weights, the system need only assign equal weights to all facets as default values.

An interesting compromise possibility would avoid asking the searcher to assign actual numbers. Instead, the user would be instructed merely to order the facets (term lists) down or across the screen in approximate descending order of importance in the information need. In a system with good editing facilities, including a facility for quickly and conveniently rearranging the lists into any desired order (e.g., with a mouse), users who would otherwise be loath to give any judgments about relative importance of facets might be coaxed into doing so in effect by way of the ordering. The system would then assign weights by an arbitrary scheme that gives larger weights to facets ranked higher by the user.

Considerable research has been carried out on weighted-request systems, including the use of both user-supplied subjective weights as just described, and computer-derived weights of various sorts. Weighted indexing (as distinguished from weighted requesting) is also possible (e.g., an index term can be assigned to a document with a weight equal to the number of times it occurs as a word in that document). Various mathematical formulas have been proposed for exploiting request term weights and indexing weights for retrieval purposes, including some which have interesting vector-space interpretations (see e.g., [3,4]). Although these formulas do not always have as firm a theoretical basis as one could wish, it seems likely (and experiments would appear to verify) that any of them would tend to work better than a system design that does not allow the use of weights at all.

PROBLEM 4: INTERPRETING THE WEIGHTS

The thorniest problem connected with any weighted request scheme is the question of what the weights assigned to the request terms or facets are supposed to mean. For many users it will seem an inadequate explanation merely to say that the weights are supposed to express the relative "importance" of the facet to the information need. What does "importance" mean, after all, and how should one go about quantifying it in one's mind? Moreover, even for those users who are willing to hazard a quantitative estimate of importance, it is far from clear how the system should manipulate the resulting numbers to achieve optimal retrieval. Although the various formulas just alluded to for exploiting the weights are all fairly plausible, they are, in the last analysis, somewhat arbitrary.

Solution: The probabilistic interpretation

One of the things information retrieval researchers have accomplished in the past decade has been to put retrieval theory on a firmer statistical basis (for surveys see [5–7]). The starting point of information retrieval theory is the recognition that the items in the system output produced in response to a search query should in general be ranked in descending order of probability of usefulness to the searcher. This is the so-called Probability Ranking Principle [8–11]. From this principle it follows that clues supplied to a retrieval system should, whenever feasible, be provided in a form that will make it easy to estimate the required probabilities of usefulness. In particular, when term request facet weights are permitted they should if possible be given a probabilistic interpretation.

One such interpretation is the following: When a searcher includes in his request a term (or facet) T with weight W, the weight W is to be regarded as the searcher's subjective estimate of the probability that a stored record having the term T among its index terms would be relevant to his information need. For example, a surefire term whose presence on a document is almost a guarantee in the user's mind of the document's pertinence would be assigned a probability of close to one in the request, whereas terms that would have been negated in a traditional Boolean request would be assigned a probability close to zero. Equivalently, but less formally, the user might be instructed to try to imagine the set of all documents in the collection that bear the descriptor T and to guess at the proportion of those documents that might be useful. For instance, when documents are stored in full-text form and all content words contained in a document are regarded as descriptors of it, the weight assigned to a request term such as TRANSISTOR would be the searcher's guess as to the fraction of documents containing occurrences of the term TRANSISTOR that would be useful.

With the request term weights so interpreted, it is possible to program the computer

to estimate a probability of usefulness for each document in the collection. The simplest known formula for doing this is one given by Robertson and Sparck Jones, whose paper may be consulted for the mathematics [12], cf. [13]. The system output given in response to the request then consists of all documents for which this estimated probability is greater than some threshold probability, arranged in descending order of the estimated probability. The scheme has been shown to be computationally feasible in experimental setups (e.g. [14]), and though the probability-of-usefulness estimates so obtained are very crude, they have at least some foundation in statistical theory and the output rankings they produce are probably superior to those of ad hoc schemes.

It is not yet known how many retrieval system users would be willing to make probabilistic guesses of the kind called for by this design. However, it is not implausible that some might find probability estimation no harder than trying to quantify an ill-defined concept such as "importance." It can be taken for granted that the users' probabilistic guesses would usually be very rough, but again this must be weighed against the alternative of an arbitrary weighting scheme under which the output ranking rule is theoretically unmotivated. In any case there would seem to be no harm in making the probabilistic interpretation available to those users who wish to learn it. Even for users who refuse to make the effort to think probabilistically, preferring to assign weights merely by subjective "importance," there is no evidence that the probabilistic retrieval algorithm would perform any worse than its ad hoc competitors.

For users willing to attempt the kind of scientific guesswork the probabilistic approach calls for, various aids might prove useful. Consider again the situation of a searcher who has decided to include TRANSISTOR among his or her request terms and is in the process of assigning a weight to it. At a minimum, the number of records in the collection containing occurrences of TRANSISTOR should be displayed as background information to aid his or her decision. In addition, it would be helpful to display beside it another number, derived from prior experimental data, representing the proportion of records typically found useful in a set of this size defined by a request term. If the user does nothing, this second number would be taken as the default-value probability weighting for the term. If on the other hand the user is willing to provide his own subjective probability estimate for the term, he or she could do so simply by modifying the displayed number.

There is experimental evidence that even in the absence of any user-supplied request term weightings, retrieval effectiveness can generally be improved by having the system arbitrarily assign somewhat larger weights to request terms that are more specific (in the sense of indexing fewer documents) and smaller weights to terms that are broader. The user interface just described would automatically confer this benefit as a byproduct.

Probabilistic ranking within the output obtained from a conventional Boolean query is also a possibility (see e.g. [15] and the article "Probabilistic Methods for Ranking Output Documents in Conventional Boolean Retrieval Systems" in this issue). Hybrid systems of this sort would seem an attractive option to offer to those users who happen to be accustomed already to the Boolean request format and for that reason prefer it.

PROBLEM 5: TERM DEPENDENCIES

The probabilistic retrieval formula proposed by Robertson and Sparck Jones was derived with the help of a strong simplifying assumption concerning the statistical independence of index terms in the document collection. It is an assumption that is only approximately true, at best. The performance of probabilistic retrieval systems could presumably be improved if this assumption could be removed or replaced by a weaker assumption that would allow data concerning term dependencies to be used as part of the procedure for estimating the usefulness probabilities of the output documents.

Solution: Advanced statistical techniques

Various ways of using term dependency data to improve probabilistic retrieval computations have been explored. One possibility, proposed by van Rijsbergen, is based on the notion of a "maximum spanning tree" [16,17]. A related approach makes use of the so-

called Maximum Entropy Formalism [18–20]. The latter is especially flexible in that it eliminates the need for statistical independence assumptions while making use of whatever probabilistic data or constraints might happen to be available. However, we mention these schemes only as possible future solutions to the dependency problem, not as immediate practical proposals, because it is not yet clear which (if any) of them will prove to be computationally feasible. This is an area of ongoing research.

SUMMARY

We have tried to suggest, by describing a series of gentle steps away from the standard Boolean design, that it should be possible to improve considerably upon the fundamental design features of most present-day retrieval systems and that this can be done simply by exploiting ideas that are already available in the research literature. The prevalence of conventional Boolean systems today does not reflect their inherent virtue so much as a historic head start.

With the advent of the mini- and especially the microcomputer, new opportunities have been and will continue to present themselves for introducing superior designs, either as intelligent interfaces for making better use of established Boolean services, or as independent search systems of various kinds. It is to be hoped that these opportunities will not be lost simply through a lack of awareness of the available alternatives.

REFERENCES

1. Bar-Hillel, Y. A logician's reaction to theorizing on information search systems. American Documentation. 3:103–113; 1957. Reprinted in: Bar-Hillel, Y., Language and information. Reading, MA: Addison-Wesley; 1964.
2. Lancaster, F.W. Information retrieval systems: Characteristics, testing and evaluation (2nd ed.). New York: Wiley; 1979.
3. Salton, G. Automatic information organization and retrieval. New York: McGraw-Hill; 1968.
4. Salton, G.; McGill, M.J. Introduction to modern information retrieval. New York: McGraw-Hill; 1983.
5. Robertson, S.E. Theories and models in information retrieval. Journal of Documentation, 33(2): 126–148; 1977.
6. Maron, M.E. Probabilistic retrieval models. In: Dervin, B.; M. Voigt, editors. Progress in communication sciences vol. V. Norwood, NJ: ABLEX; 1984; 145–176.
7. Bookstein, A. Probability and fuzzy-set applications to information retrieval. In: Williams, M.E. editor. Annual review of information science and technology, vol. 20. Washington, DC: ASIS; 1985; 117–142.
8. Cooper, W.S. The suboptimality of retrieval rankings based on probability of usefulness. Technical Report, School of Library and Information Studies, University of California, Berkeley, CA 94720; 1976.
9. Robertson, S.E. The probability ranking principle in I.R. Journal of Documentation, 33(4): 292–304; 1977.
10. Cooper, W.S.; Maron, M.E. Foundations of probabilistic and utility-theoretic indexing. Journal of the Association for Computing Machinery, 25(1): 67–80; 1978.
11. Robertson, S.E.; Maron, M.E.; Cooper, W.S. Probability of relevance: A unification of two competing models for document retrieval. Information Technology: Research and Development, 1(1): 1–21; 1982.
12. Robertson, S.E.; Sparck Jones, K. Relevance weighting of search terms. Journal of the American Society for Information Science, 27: 129–146; 1976.
13. van Rijsbergen, C.J.; Robertson, S.E.; Porter, M.F. New models in probabilistic information retrieval. British Library R & D Report No. 5587, Computer Laboratory, University of Cambridge, Cambridge, England; 1980.
14. Robertson, S.E.; Bovey J.D. A front end for IR experiments: Final report to the British Library Research and Development Department on project Number SI/G/569. Report No. 5807, Department of Information Science, The City University, London, England; 1983.
15. Radecki, T. A probabilistic approach to information retrieval in systems with Boolean search request formulations. Journal of the American Society for Information Science, 33(6): 365–370; 1982.
16. van Rijsbergen, C.J. A theoretical basis for the use of co-occurrence data in information retrieval. Journal of Documentation, 33(2): 106–119; 1977.
17. van Rijsbergen, C.J. Information Retrieval (2nd ed.). London: Butterworths; 1979; chapt. 6.
18. Cooper, W.S.; Huizinga, P. The maximum entropy principle and its application to the design of probabilistic retrieval systems. Information Technology: Research and Development, 1(2): 99–112; 1982.
19. Cooper, W.S. Exploiting the maximum entropy principle to increase retrieval effectiveness. Journal of the American Society for Information Science, 34(1): 31–39; 1983.
20. Kantor, P. Maximum entropy and the optimal design of automated information retrieval systems. Information Technology: Research and Development, 3: 88–94; 1984.