UIC John Marshall Journal of Information Technology & Privacy Law

Volume 1
Issue 1 *Computer/Law Journal*

Article 9

1978

Legal Information Retrieval Systems: The Need For and the Design of Extremely Simple Retrieval Strategies, 1 Computer L.J. 379 (1978)

Jon Bing

LEGAL INFORMATION RETRIEVAL SYSTEMS: THE NEED FOR AND THE DESIGN OF EXTREMELY SIMPLE RETRIEVAL STRATEGIES

*by Jon Bing**

INTRODUCTION

Legal questions are the reason for all legal research. The lawyer today is offered a wide variety of tools to research those questions—from the conventional subject indexes and systematic tables to powerful, computerized retrieval systems like ITALGIURE.[1] To

* Candidate of law (Oslo) 1969; research assistant, Institute for Private Law, Oslo University, 1970; associate professor in computers and law, Norwegian Research Center for Computers and Law, Oslo University, 1972 to date. Dr. Bing's main fields of research are legal information systems, decision theory, privacy, deontic systems, and related computer and law subjects, although he has also published materials on copyright law. Most of his works have been published in Norwegian. In English, his major work is LEGAL DECISIONS AND INFORMATION SYSTEMS (Norwegian University Press 1977), co-authored with Trygve Harvold. The author wishes to thank his colleagues at the Norwegian Research Center for Computers and Law for comments on the ideas contained in this article, especially Mr. Trygve Harvold. Some of the ideas in this article were outlined earlier in a report to the Commission of the European Communities Technical Study in Legal Information Retrieval, entitled Users and Usage of Legal Information Systems (Jan. 1978). An earlier version of this article was presented at the II Conference on Judicial Informatics, Rome, June 1-3, 1978.

1. The ITALGIURE project originated within the Italian Corte di cassazione in 1963-1964. This court issues approximately 12,000 decisions each year which are considered important in the Italian legal system. Originally, the system was designed to facilitate the dissemination and retrieval of the court's decisions in the form of authoritative abstracts ("massimes"). It has now been extended, however, to include other types of legal documents, such as doctrine and national legislation, EEC legislation and jurisprudence. As of November 1977, the data bases contained approximately 450,000 documents. The system operates on UNIVAC computers and incorporates self-developed retrieval software (FIND), which supports interactive retrieval based on Boolean queries, and offers unique thesauri facilities for query expansion. Today, the ITALGIURE network covers twenty-six courts of appeal, 159 tribunals and many public authorities, totaling over 275 terminals. By 1979, the ITALGIURE data bases will be accessible through EURONET.

use a retrieval system, a lawyer must transform his question into a query, constructed according to the rules of the retrieval system being used. In conventional systems these rules are simple and well-known, *e.g.*, an index term is selected and located in the alphabetically sorted index, or a node in the systematic table is defined, based upon the lawyer's knowledge of the legal system. In computerized systems, however, these rules are more complex. Computerized systems have more possibilities than conventional systems and, consequently, a higher degree of complexity. This article, because of its brevity, cannot attempt to indicate the wide variety of choices available to the designer of a computerized, text retrieval system. It is sufficient to state that today, most systems use Boolean operators,[2] several have different ranking functions available,[3] and some have possibilities for query expansion through the use of thesauri, grammatical generators and other means.[4]

The basic retrieval strategy is not difficult to learn. This ease of learning has led many to think that research into the possibilities of even simpler strategies should be given low priority.[5] User studies, however, have indicated that, although the Boolean retrieval strategy is not difficult to learn, neither is it easy to use. For instance, DATEV in Germany[6] has simplified the user interface of STAIRS,[7]

---

2. Boolean algebra, named after the British mathematician George Boole (1816-1864) consists basically of the logical operators *and* and *or*. If A and B are two sets of elements, A *and* B represents the intersection of the two sets, while A *or* B represents the union of the two sets. Applied to text retrieval, the combination of two search words with the *and* operator represents the condition that requires both words to be present in a document for that document to be retrieved. Combining two words with the *or* operator represents the condition that permits retrieval of a document if either word is present. The basic, Boolean retrieval strategy requires a user to perform a search by combining search words with these operators.

3. A Boolean query may be considered as placing the documents of the data base into one of two sets (binary ranking)—those retrieved and those not retrieved. When speaking of ranking functions, one commonly understands that function as sorting retrieved documents pursuant to some given criteria, in which the highest rank is assigned to that (or those) documents which, according to the criteria, have the highest probability of satisfying the user. Several examples of ranking functions (or algorithms) are discussed at § IV *infra*.

4. *See* § VI *infra*.

5. *See* BING, HARVOLD, KJØNSTAB & STABELL, KONTROLLERT FORSØK PÅ TRYGDERETTENS KJENNELSER—NORIS (8) I, at 76-77 (Norwegian Research Center for Computers & Law, Paper # 18, 1976).

6. DATEV (a welcome acronym for Datenverarbeitungsorganisation der Steuerbevollmächtigen für Anhörigen des steuerbearbeitenden Berufes in der Bundesrepublik Deutschland) is a non-profit organization of some 12,000 members in the tax-consulting professions. Since 1975, DATEV has offered a data base of tax law (Steuerrechtsdatenbank) to its members. As of July 1977, this data bank contained approximately 34,000 documents, with a monthly increase of 600-1,000 documents including jurisprudence, administrative decrees and literature. At the present time, the

as has CRIDON de Lyon[8] with MISTRAL.[9] Problems in maintaining competence have also been reported—though easy to learn, acquired skills can be maintained only through regular practice. The relatively low number of direct end-users in European systems may result from this difficulty. The difficulties encountered by casual users appear considerable. The problems of the novice can be reduced if a simple strategy is available.

This article does not propose that there are sufficient reasons to revise current priorities and to launch a massive effort to develop a simpler user interface. However, there is an obvious need for a very simple, retrieval strategy as a supplement to those currently in use. Such a strategy would have the advantage of demanding little skill from the user. Ideally, the user would not be required to adhere to any rules apart from those necessary to operate the terminal; a simple retrieval strategy would move control from the user to the system. One would expect an experienced user to employ a more sophisticated strategy to obtain more satisfactory response from the system. A simple retrieval strategy will expectedly reduce system performance and, indeed, one of the major problems will be to design a simple retrieval strategy so that the loss in performance will be an acceptable trade-off. Introducing a very simple, retrieval strategy as an alternative will, hopefully, replace the high entry threshhold of existing systems and permit rapid development of necessary user competence.

---

system is reported to have nearly one thousand users. It runs on an IBM computer, using the STAIRS retrieval software. *See* note 7 *infra*.

7. STAIRS (an acronym for Storage and Information Retrieval System) is a terminal-oriented, interactive, general purpose retrieval system developed by IBM. The system operates on either IBM System 360 or System 370 computers under the Customer Information Control System (CICS) or Information Management System (IMS). Retrieval is usually conducted on an IBM video display terminal with function keys—though DATEV also uses TTY terminals for this purpose. The search language of STAIRS is based on Boolean operators (*see* note 2 *supra*), but is extended by word frequency ranking functions. *See* § IV *infra*.

8. CRIDON de Lyon is a French organization established to supply the notaries of the de Lyon region (twenty-eight departments) with legal information services. Since the organization normally does not supply documents, but only advice, the computer system developed by CRIDON de Lyon primarily serves its own staff. Plans have begun to make the system available to the four other French CRIDONs and the Conseil supérieur du notariat through a network. The data base contains some 70,000 documents, which were selected to satisfy the specific needs of a notary. The MISTRAL retrieval program (*see* note 9 *infra*) has been redesigned to meet the changing requirements of the system as the system continues to develop.

9. 1 TECHNICAL STUDY IN LEGAL INFORMATION RETRIEVAL: MAIN REPORT 60 (1977) [hereinafter TECHNICAL STUDY]. MISTRAL is a retrieval program supplied by the CII-HB; the version at use by the CRIDON de Lyon is MISTRAL-V2 (1977). To simplify the search language, MISTRAL has been combined with the retrieval system SUMER from the service bureau EURINFOR.

II. Alternate Principles of Design

There are several basic principles available for designing a simple user interface. Three will be discussed.

A. A Question-Answer System

In a question-answer system, the system poses questions to the user, and the user responds to them. The system leads the user through a decision-tree,[10] which eventually concludes with the system presenting the information identified by the dialogue as corresponding to the user's needs.

Such a decision-tree can be constructed on several different principles.[11] A systematic table[12] easily lends itself to such an adaptation. Such a table leads the user through a hierarchical, systematic index, starting with general legal terms and concluding with one or more specific categories. The documents in the data base, classified as belonging to those categories, are then displayed to the user.[13] This approach results in a system not significantly different from one based upon intellectual indexing;[14] the possibilities and re-

10. A decision tree may be viewed as a question-answer strategy, in which the user is presented with a rather general question, and answers "yes" or "no." The tree then forks into a yes-limb and a no-limb, each leading to a new, more specific question. This question is also to be answered by "yes" or "no," which causes a new bifork, and so on, until the user reaches the terminal points—the roots of the matter (or rather the leaves, since this is an upside-down tree). If the questions are adequately designed, the dialogue will permit the user to describe his problem or situation, specify his requirements, make a decision, and obtain the sought-after result.

11. Two examples are given below. In one, the decision-tree is based upon an analysis of the relationship between legal issues (a systematic table); in the other, it is based upon the relationship between the possibly legally relevant facts which depict the particular user's situation. *See* notes 12-17 *infra* and accompanying text.

12. The term "systematic table" is used herein as a general term for any scheme which describes the relation between legal issues organized from general issues to specific issues, and often including cross-references. A prime example of such a scheme is the Key Number System developed by West Publishing Company.

13. An example of a system where some of these possibilities are realized is the French DARIUS system. This system is an interesting hybrid of computer and microfilm technology. The computer is used for phrasing the query—a systematic table is projected on the terminal screen, and the user identifies (and combines) indexing terms with a light-pen. When the system reaches the end nodes of the systematic table, the identified documents are accessed in a "mémoire de masse optique" (M.M.O.), and the microform images are projected onto the same screen. *See* Buffelan, Introduction a l'informatique juridique 181-92 (1975).

14. CREDOC (Centre de documentation juridique), which began operation in 1966, was created to assist Belgian notaries. It is the oldest of the European systems still operational. CREDOC has adopted an indexing philosophy which results in the assignment of a four-digit indexing term to each document from a list of some 7,000 descriptors, with additional ante- and post-descriptors. The thesauri list the terms in both French and Dutch. In this way, CREDOC has solved the problems of retrieval of

strictions of such a system are also similar.[15]

A different approach is to construct a decision-tree which allows the user to characterize his *problem*, rather than the location of his problem in the legal system. The user describes the facts of the case by employing the alternatives presented to him by the system. The system then locates those documents determined to be pertinent to those facts. This requires that the system have the ability to link facts with legal sources, which is by no means impossible; systems for individual consultancy in divorce[16] and tax[17] law have already been developed. Such a system would require a large amount of resources in designing the decision-tree to be used in the dialogue. It is doubtful, however, that a professional environment would be a suitable context for such a system—it would appear more promising as an interface to the lay user.

B. A Citation-Based System

Citations have not been exploited to their full potential in legal systems. A reference to a case or a statute is very often an adequate and obvious way of formulating a query. Users place great importance on citations[18] and, in cases where the reference to a relevant document is known, it is a simple query to formulate.

Studies of citation structures in legal sources are rather scarce.[19] The nature and interpretation of citations in legal sources have still not been satisfactorily researched, and experiments with systems representing networks of citations should be undertaken. One might expect rather distinct differences between legal systems. For instance, while cases in the Anglo-American system properly

---

documents in the bilingual Belgian legal system. The retrieval software was originally developed specially for CREDOC. The system was batch-oriented, and the CREDOC consultants used additional tools in advising clients. Today, CREDOC is supported by the IBM STAIRS system (*see* note 7 *supra*) on an interactive basis.

15. The practical benefits to be derived from using a systematic table may, of course, be significant. However, the principal constraints on retrieval efficiency are the same, including the problem of constructing an adequate indexing language for describing the content of the documents and the problem of correct, exhaustive and consistent indexing.

16. McCoy & Chatterton, *Computer-Assisted Legal Services*, 11 LAW & COMPUTER TECH. 2-7 (1968).

17. Bellord, *Office Management and Tax Modelling*, in THE SOCIETY FOR COMPUTERS AND LAW, LTD., PRACTICAL BENEFITS OF THE COMPUTER FOR LAWYERS 6-11 (1976).

18. TECHNICAL STUDY, *supra* note 9, at 55-56.

19. *See, e.g.*, BERGER, DIE ERSCHLIESSUNG VON VERWEISUNGEN BEI DER GESETZDOKUMENTATION (1971); BING & HARVOLD, LEGAL SOURCES AND INFORMATION SYSTEMS—NORIS (3) & (4), at 53-74 (Norwegian Research Center for Computers & Law, Paper #4B, 1973); Tapper, *Citation Patterns in Legal Information Retrieval*, 5 DATENVERARBEITUNG IM RECHT 249-75 (1976).

can be represented by their case citations, those in a Continental system are perhaps better represented by their statutory citations.

C. A Natural Language Retrieval Strategy

A retrieval strategy based on natural language queries has several attractive features. It is simple, as there are no restrictions on query formulation. One may also cite extracts from documents, articles and statutes as queries. The rules for framing such queries do not derive from the query language, but rather from the procedural rules of the retrieval system, such as those necessary for terminal handling and log-on protocols. The problem with natural language queries is to make them efficient. The remainder of this article addresses different possibilities for increasing the performance of natural language retrieval strategies, and their applicability to legal information retrieval systems.

III. CONTROLLED EXPERIMENTS

In an effort to compare the performance of different retrieval strategies, a method for controlled experiments in text retrieval has been developed at the Norwegian Research Center for Computers and Law.[20] Independent of the retrieval strategy used in a particular experiment, sets of "relevant documents" are developed for each of the questions in the experiment. For each particular question, the experimenters formulate alternative queries corresponding to each retrieval strategy being tested. The results of the retrieval are compared to the "relevant documents," and the performance is expressed in terms of recall and precision.

If the total number of relevant documents in the data base is T, the number of retrieved relevant documents is R, and the total number of retrieved documents is D, recall may be defined as R/T and precision as R/D. The best possible result is represented by 100% recall and 100% precision.[21] For each retrieval strategy in the experiment, as many recall-precision graphs are produced as there are questions in the experiment. These are reduced to average

20. The method was first proposed in BING & HARVOLD, KONTROLLERT FORSØK I TEKSTØKING PÅ AVGØRELSER AV SVENSKE FORVALTNINGSDOMSTOLER—NORIS (8) II, at 6-39 (Norwegian Research Center for Computers & Law, Paper #9, 1974); and has been employed in two later experiments reported in BING, HARVOLD, KJØNSTAD & STABELL. *supra* note 5, at 23-36, and FJELDVIG, KONTROLLERT FØRSOK I TEKSTSØKING PÅ UTTALELSER FRA SKATTEDIREKTØREN 19-40 (Norwegian Research Center for Computers & Law, Paper #16).

21. These measurements of retrieval performance are in common use, but are based on the rather controversial concept of *relevance*. *See* BING & HARVOLD, LEGAL DECISIONS AND INFORMATION SYSTEMS 153-57 (1977).

graphs. For the purposes of this article, only average graphs are presented.

To illustrate a few of the suggestions expounded in this article, a small, controlled experiment was conducted on a data base consisting of 376 responses from the Norwegian Central Tax Authority. The experiment was "piggy-backed" on an earlier experiment—NORIS (8) III.[22] Nine of the questions in the original experiment were selected, all questions being fairly complex[23] and all having an answer set greater than or equal to three documents. The experiment was conducted using NOVA*STATUS, the Norwegian version of the British STATUS I text retrieval system, which is an advanced text retrieval system with a very flexible user interface.[24] At the Norwegian Research Center for Computers and Law, NOVA*STATUS has been extended with new facilities named VEXT. VEXT supports controlled experiments and vector-based retrieval,[25] and calculates recall-precision graphs and average recall-precision graphs as part of the dialogue.[26]

IV. FREQUENCY-RANKED RETRIEVAL STRATEGIES

A natural language retrieval strategy should allow the user to formulate his query just as he would if he wished to pass it on to a colleague. For instance, if the user's client were demanding damages for injuries suffered in an automobile accident, the user would perhaps formulate the following query:

> Do you know any cases in which a motor accident has resulted in personal injuries?

To handle this query, the system would reconstruct the query as a conventional Boolean query. This could be done by first excluding common words according to a pre-defined stop list, and then combining the remaining words with the Boolean operator *or*. In the example, this would result in the following query;

> Do you know any cases in which a motor accident has resulted in personal injuries?
> Stop words:

---

22. FJELDVIG, *supra* note 20.

23. *See* § VII *infra*.

24. An English description is given in HARVOLD & ORE, THE NOVA*STATUS TEXT RETRIEVAL SYSTEM, COUNCIL OF EUROPE CI-LJ(77) 7 (1977).

25. *See* § V *infra*.

26. The text retrieval laboratory's NOVA*STATUS with the extension VEXT is based upon experiences with the NORIS research program in legal informatics, which has been running at the Norwegian Research Center for Computers and Law since 1971. The system became operational in 1978. *See* FJELDVIG, VEXT: ET SYSTEM FOR VEKTØRSOKING OG RESULTATANALYSE—NORIS (28) (Norwegian Research Center for Computers & Law, Paper #28, 1978).

DO YOU ANY IN WHICH A HAS IN

System-constructed Boolean query:

KNOW *or* CASES *or* MOTOR *or* ACCIDENT *or* RESULTED *or* PERSONAL *or* INJURIES

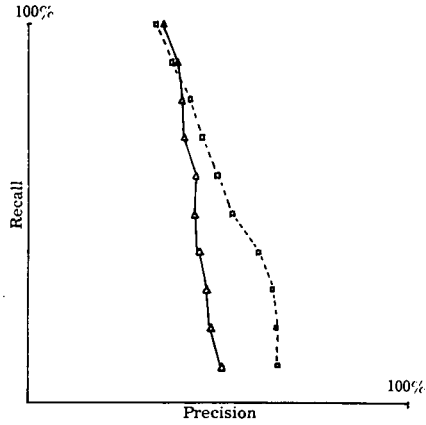
It is obvious that this query is less satisfactory than a query constructed directly by an experienced user employing a Boolean strategy. For instance, the query as formulated will retrieve cases in which only the word "know" occurs. The probability of such a case being relevant to the problem at hand is rather slight.

This weakness can be remedied to some extent by ranking the documents retrieved by this very broad query. The simplest ranking algorithm is to count the number of words from the query that occur in the document. In this way, a document where only five occurrences is scored will rank below a document in which twelve appear. It is safe to assume that documents in which many words from the query occur have, on the average, a higher probability of relevance.

Through such a ranking scheme, results are somewhat improved. In one of the experiments in the NORIS program, natural language queries of this type were compared to the strategy recommended by the experimenters. The results are shown in Figure 1[27]:

Figure 1

Experimental results NORIS (8) I



In this experiment, the natural language queries were simulated with the help of STAIRS, which has several ranking algorithms. Each query was originally formulated in natural language, but the search words were then truncated on the basis of manual judge-

27. *See* BING, HARVOLD, KJØNSTAD & STABELL, *supra* note 5, at 54.

ment. The "recommended strategy" was conceptor-based.[28]

The Canadian system QL (formerly QUIC/LAW)[29] began as a system which permitted users to formulate natural language queries. The system displayed the words left after the exclusion of common words, then queried the user as to whether he wished to proceed on this basis. Cued with a "yes" from the user, the system then searched for and ranked the retrieved documents.

There are many variations of word frequency ranking which have been discussed at length elsewhere[30] and will not be reviewed in this article. It should be noted, however, that the algorithms in use are more sophisticated than the one outlined above, though constructed on the same principle. Most of these algorithms are based upon the principle of frequency, where a word that appears in a great many documents is assigned a lower ranking value then one that appears only once or a few times. Also, the length of the document is taken into account, as the high frequency of a word in a long document may not indicate a higher probability of relevance, but merely that words are more likely to appear repeatedly in longer documents.

STAIRS, which was used in the experiment mentioned above, offers five different ranking algorithms.[31] The default algorithm A was found of lower performance than the optional algorithm E.[32] Algorithm E is characterized by allowing a "rare" word to contribute relatively more to the weight assigned to a document in ranking.[33] It is of interest to note that in another NORIS experiment, ranking documents on the frequency of *different* words from the query gave relatively higher performance than ranking on the *total* number of words from the query.[34] Also, an adjustment for document length improved the results. In the experiment just mentioned, the greatest improvement was obtained by adjusting the ranking of the algorithm by the following formula:

$$\text{rank value} = F_S/L^{0.5}$$

Where F_s is the frequency of search words, and L is the length of the document (in number of words).

---

28. *See* § VII *infra*.

29. The QL system is the basis for the United States WESTLAW system offered by West Publishing Company, though WESTLAW also has Boolean strategies available in more recent versions. *See* BING & HARVOLD, *supra* note 21, at 122-25.

30. For a comprehensive discussion, *see* Sager, *Vergleich von Rangfolge-Algorithmen*, MAJUS 7 (1976).

31. These algorithms are discussed in detail in BING, HARVOLD, KJØNSTAD & STABELL, *supra* note 5, at 94-101.

32. This result confirms the conclusion in Sager, *supra* note 30, at 26.

33. *See, e.g.*, STAIRS/VS (Storage and Information Retrieval Systems/Virtual Storage), Program Reference Manual, SH 12-5400-0 (1974).

34. FJELDVIG, *supra* note 20, at 78. *See also* § VII *infra*.

V. VECTOR-BASED RETRIEVAL STRATEGIES

Apart from the Canadian QL-system, only one other commercial system has offered users a natural language query ability—the Swiss CONTEXT system.[35] To the user, this system was similar to the QL-system, while, in fact, it employed a radically different method of retrieval known as the "vector principle." CONTEXT is the only commercial system which has used this principle for text retrieval.[36]

A vector may be pictured as an arrow of a certain length. The length of the arrow is determined by the number of elements composing its shaft. The direction in which the arrow points is determined by the value assigned to these elements. In a vector-based system, each document is represented by a vector, consisting of as many elements as there are different words in the data base. The value of each element is usually determined by the number of occurrences of the word in the document. As an example, a brief segment of the vectors of two cases discussing motor vehicle accidents could be displayed:

Example 1

Segments of document vectors

	car	careful	caretaker	cars	carve
Case A	4	3	0	6	0
Case B	8	1	0	12	1

In the same way, document queries may also be represented as vectors. Vectors may be compared and measured for similarity. On this basis, documents in the data base may be ranked—the docu-

35. CONTEXT is a text retrieval system developed by the Swiss firm UNIDATA AG, founded in 1968 by Swiss lawyers. This system is now being marketed by DATAPLUS. Designed during 1968-1970, CONTEXT has been demonstrated at several conferences, and the company, Juristische Datenbank AG, has been founded to operate the system. Though generating a great deal of interest, the system has not been adopted by any operational legal documentation center. The main point of interest is the basic philosophy of the system, which uses vector retrieval and natural language queries. In a comparative test with the Belgian CREDOC system (*see* note 14 *supra*), CONTEXT performed relatively better. *See* Prestel, *Datenverarbeitung im Dienste juristischer Dokumentation*, 3 EDV UND RECHT. *See also* BING & HARVOLD, *supra* note 21, at 140-42.

36. Vector retrieval is supported by the VEXT extension to NOVA*STATUS. *See* note 26 *supra*.

ment vector having the greatest similarity to the query vector being ranked first.[37]

In an experiment conducted at the Norwegian Research Center for Computers and Law, ranking on the basis of vectors was compared with ranking on the basis of word frequency. As shown in Figure 2, the results were very similar.[38]

Figure 2
Experimental results NORIS (8) III

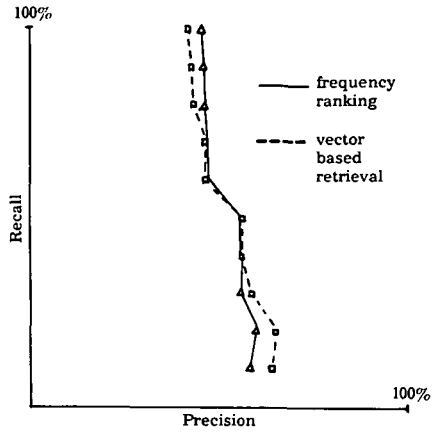
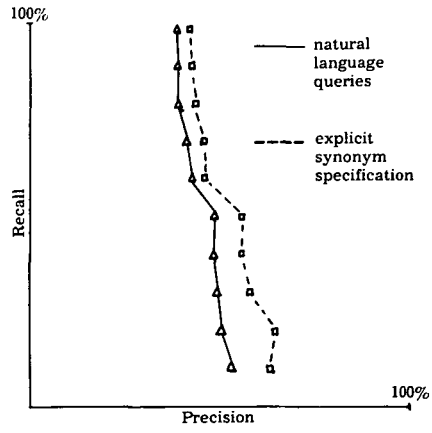


Figure 3
Experimental results NORIS (8) III[39]



37. The measure for similarity is usually the cosine value, but there are variations. In the VEXT extension of NOVA*STATUS, the cosine value is used as the criterium for ranking. FJELDVIG, *supra* note 20, at 116.

38. *Id.* at 135. In this comparison, identical queries were used, though not natural language queries but queries where synonyms were specified. The word frequency ranking was adjusted for document length.

39. *Id.* at 132.

Figure 2 indicates the relative performance of word frequency ranking and vector-based retrieval. In the experiment, vectors were also used for natural language queries. As shown in Figure 3 above, when comparing the results obtained by natural language queries with those obtained through queries where synonyms were specified, the difference against natural language queries was surprisingly small.

In preparing this article, the author performed a small experiment using a portion of the questions basic to the results illustrated in Figures 2 and 3 *supra*. The results of this experiment are set forth in Figure 4.

Figure 4

Experimental results: natural language queries—word frequency ranking and vector based retrieval

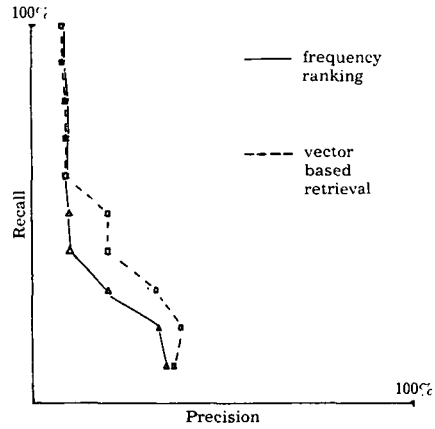


Figure 4 is in some ways a variation of the graphs shown in Figures 2 and 3, since a subset of the same questions was used. The query formulation was different, however, since it was based on plain, natural language queries. Additionally, word frequency ranking did not compensate for document length, since NOVA*STATUS, which permits word frequency ranking, does not adjust for document length. Only the relative differences between the curves should be considered since the absolute result is incidental and heavily dependent upon the small number of curves used for the calculation of averages.

While a conclusion drawn from this discussion is necessarily tentative, it appears that natural language queries perform less satisfactorily than queries formulated in a more complex manner. Nat-

ural language queries, however, in no way perform so badly that they should be rejected out of hand. Frequency ranking has, in one experiment at least,[40] given recall values of fifty percent at a precision of fifty percent, indicating that about half of the relevant documents are retrieved with no more than every second document being irrelevant. Vector-based retrieval performs at the same general level as frequency ranking,[41] while in vector-based retrieval systems, the performance reduction which results from using natural language queries does not appear excessive.[42] However, there are possibilities for improving the results beyond those that have been obtained. In the next two sections, several possibilities are examined.

VI. Query Expansion by Mechanical Truncation

One of the general problems encountered in achieving high performance in text retrieval systems is that of specificity[43]—known as the "synonym problem." Documents in text retrieval systems are normally written in a natural, free language. There is a great profusion of different word-forms and synonyms in use, and one of the major, practical problems for the user is specifying a sufficient number of synonyms in his query.

The synonym problem becomes severe in natural language queries since the user normally specifies only one word for each concept. This word is usually the most convenient one for him. This can be illustrated by the example mentioned above.[44] The user may select to search on the phrase "motor vehicle accident," and not specify the words "car" or "automobile." Yet, these two words would be highly characteristic of cases dealing with personal injuries from motor vehicle accidents. The problem is to find a way in which the retrieval system can expand the word "motor vehicle" to include synonyms such as "car," "automobile," "van," and "lorry" and even such trade names as "Ford," "Lancia," or "Volvo."

One well-known solution to this problem is the establishment of a synonym thesaurus. Such a thesaurus specifies links between a great number of words and word-forms. A query is then compared against the entries in the thesaurus, and the query is expanded to include all words defined as synonymous by the thesaurus.

---

40. *See* Figure 2 *supra*.

41. *Id.*

42. *See* Figure 3 *supra*.

43. Specificity as a cause of performance failure has been addressed in some of the controlled experiments within the NORIS research program. *See* Bing & Harvold, *supra* note 21, at 220-21.

44. See § IV *supra*.

Thesauri defining context-independent synonyms are, in principle, feasible to create. By "context-independent synonyms" is meant words that in all (or most) contexts can replace each other without changing the meaning of the phrase or sentence. Examples of such synonyms are grammatical variations (*e.g.*, "car" and "cars") or, more rarely, true synonyms (*e.g.*, "automobile" and "car"). Context-dependent synonyms, however, are more difficult to link. For example, "car," "van" and "lorry" are synonymous, but it is not difficult to locate contexts where this is not the case. Specifying synonymity between words other than context-independent synonyms is risky, can only be done for a rather restricted subject area, and demands a great deal of resources—both human resources to specify the relationships between the words, and computer resources for storage of the thesaurus structure. Thesauri are in common use in computerized text retrieval systems,[45] and users put great value on them.[46] Thesauri are not, however, used in these systems to expand natural language queries, but only to facilitate ordinary (usually Boolean) query formulations.

An alternative to thesauri is a grammatical generator.[47] This device accepts words from the query as input, strips them of prefixes or suffixes arising from grammatical variations, and produces the variations expected for a regular verb or noun. For instance, a grammatical generator on the input of the word "car," would produce the words "car" and "cars" or, on the basis of the word "resulted," would produce the word forms "result," "results," and "resulted."

---

45. ITALGIURE is only one of several systems using thesauri. *See* note 1 *supra*. The Canadian DATUM system even has a bilingual thesaurus, which specifies synonymity between French and English words. *See* Mackaay, *The Creation of a Bilingual Thesaurus for a Full-Text Retrieval System*, 6 LAW & COMPUTER TECH. 2-12 (1973).

46. *See* TECHNICAL STUDY, *supra* note 9, at 55.

47. A grammatic generator has been developed as a supplement to the IBM STAIRS system. *See* note 7 *supra*. The basic work on this generator was done during 1971-1972, when IBM Austria, in cooperation with the Bundeskansleramt, constructed an experimental STAIRS supplement known as FAIR ("Fully Automated Information Retrieval") or simply "The Vienna System." *See* Brun & Pfeiffer, *Automatiserte Generierung von Flexionsformen als linguistisches Hilfsmittel zur Informationssuche in juristischen Volltexten*, in LANG & BOCH, WIENER BEITRAGE ZUR ELEKTRONISCHEN ERSCHLIESSUNG DER INFORMATION IM RECHT 239-59 (1973). The FAIR system was the basis of the STAIRS "preprocessor," which is marketed under the designation TLS, and which incorporates a grammatic generator. TLS is used in legal information retrieval, for instance, by the French Centre d'informatique juridique (CEDIJ), which is also exploring the problem of natural language queries. As part of the project of the Centro per la documentazione automatica at the Italian parliament (Camera dei deputati), an alternative and innovative addition to STAIRS has been developed, which also incorporates an ambitious grammatic generator. This work is still not reported in the literature.

Of course, only context-independent synonymity may be resolved in this manner. While grammatical generation has the advantage over thesauri of requiring less computer resources, it easily grows complex, since gramatical rules are riddled with exceptions and the generator must be provided with lists of irregular verbs and nouns. The drawbacks of both the thesauri and grammatical generator are that (1) some manual work is required to create and maintain the thesauri and establish lists of exceptions; and (2) additional computer resources are required. It is doubtful that the establishment of a thesaurus or construction of a grammatical generator can be justified solely to accommodate a natural language retrieval strategy.

An obvious alternative is right-hand truncation, which is a normal, straighforward method of solving some simple problems concerning suffixes and concatenated words. In the legal context, it seems a promising alternative, as the results shown in Figure 1 above were obtained by manual truncation of words in a natural language query.

Despite great popularity, little is known about the effects of truncation. It is known that truncation eliminates variations in the endings of a word. For example, the truncated word "car" can be derived from both "car" and "cars." Unfortunately, it also results from the truncation of words which are *not* synonyms, like "careful" or "carpenter." Due to this unwanted expansion by truncation, one may doubt the effectiveness of this strategy.

Some research has been done on this question. One example is an experiment carried out within the NORIS research program. It was found that seventy-five per cent of the words in synonym groups were matched by the truncated terms, while only twenty per cent of the words fell outside the synonym group.[48] Similar results have been obtained in Germany using three-character, right-hand truncation.[49]

Spurred by these optimistic results, truncation can be taken one step further—*mechanical truncation*. In such a system, each word in a natural language query is truncated according to a simple, mechanical rule. Such a rule should, of course, be constructed on the basis of linguistic knowledge[50] and experiments which establish the relative effectiveness of several alternative rules. In the examples below, no such factors have been considered. The examples

48. HARVOLD, BELYSNING AV SYNONYMPROBLEMET I NORSKE, FORMUERETTSLIGE LOVER (1974).

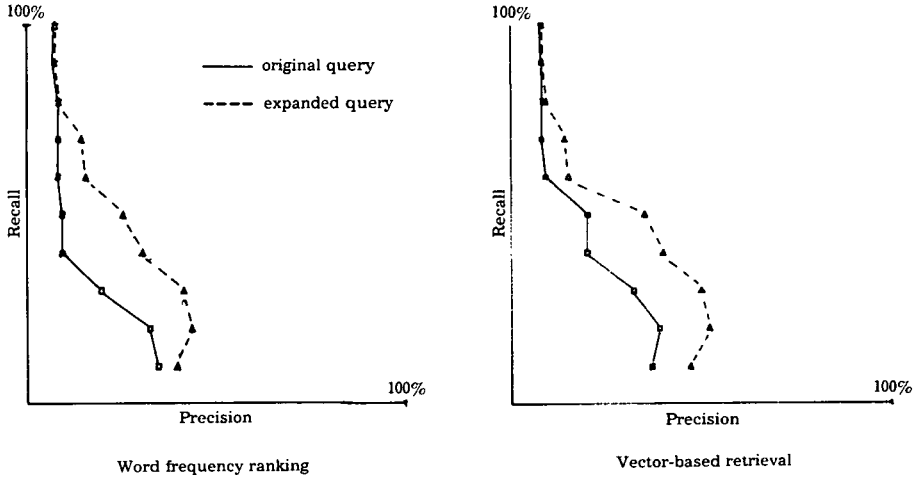
49. GEBHARDT, RECENT RESULTS RELATED TO JURIS (GERMANY), COUNCIL OF EUROPE, SYMP/INFORMATIQUE JUR. (76) 1 (1975).

50. One example of this is the average number of characters per word.

were produced solely as illustrations for this article; they show the increase in performance on the same natural language queries as discussed in Figure 4 above, achieved by mechanical truncation according to a simple, and rather incidental, rule.

Figure 5.1 and 5.2

Experimental results: natural language queries
Query expansion by mechanical truncation



The mechanical rule used for this experiment was:

> If a word has three characters or less, truncate after the last character; if a word has four to six characters, eliminate the last two characters and truncate; if the word has seven to ten characters, eliminate the last three characters and truncate; for words with more than ten characters, eliminate the last four characters and truncate.

Another alternative, at least as simple, would be to truncate after the last character for words up to six characters, and for longer words, truncate after the sixth character.

As illustrated in Figures 5.1 and 5.2 above, the use of this truncation method increases performance for both vector-based and frequency ranking strategies. This was to be expected, since query expansion in a text retrieval system, where the text of the documents is in a natural language, would theoretically always increase performance.[51] However, the increase in performance *is* rather promising. While the rule for mechanical truncation used in the example was extremely simple and selected rather at random, the in-

51. HARVOLD, PERFORMANCE OF TEXT RETRIEVAL SYSTEMS 40-41 (Norwegian Research Center for Computers & Law, Paper #15, 1976).

crease was significant.[52]

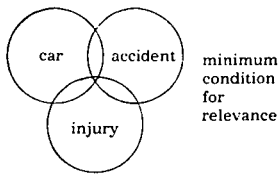
VII. Improving Performance Through Class Ranking

One retrieval strategy used with some success is known as "conceptor-based retrieval." This may be regarded as a marriage of word frequency ranking and Boolean queries. Basic to the conceptor-based query is the dissection of a question into "ideas." An idea is a concept which may be described by a number of synonyms.

Returning to the previous example: "Cases discussing personal injuries due to motor vehicle accidents," this query may be divided into three ideas: (1) the idea of a "car," (2) the idea of an "accident," and (3) the idea of an "injury."

Example 2

Question: "Cases discussing personal injuries due to motor vehicle accidents" represented as three ideas.



After the user has performed this idea dissection, he knows something important about the documents for which he is searching, *viz.*, that all relevant documents must contain all three ideas. An irrelevant document is one that does *not* contain all three ideas. If the user changes his mind as to relevance, he must also changes his "idea representation" of the query.

On the other hand, one can easily imagine an irrelevant document containing all three ideas: for instance, a case against the owner of a wheelbarrow over which a man stumbled and broke his leg on the way to his car early in the morning. This document contains words representing all three of the "ideas"—"injury," "accident" and "car"—but is clearly not relevant. When constructing a query, the user must utilize this extra knowledge and structure the query using word classes corresponding to each of the ideas identified as part of the issue. Such classes are known as *conceptors.* For example, a conceptor for the idea 'car' would include, *inter alia*, CAR, CARS, VEHICLE, AUTOMOBILE, BUS, AUTO, TRUCK, JEEP, VAN, LORRY, FORD, LANCIA and VOLVO.

In a conceptor-based retrieval strategy, ranking of the retrieved

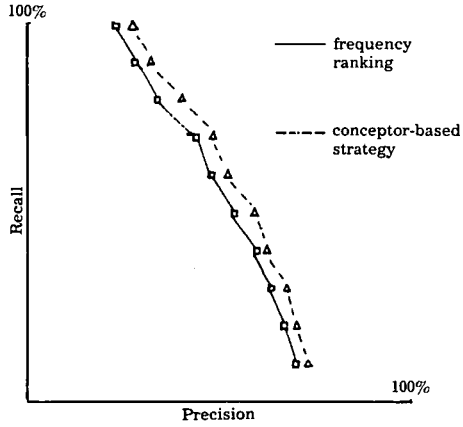
52. The average increase in precision values for recall strategies was 10-50%; for vector-based retrieval strategies, 51% and for frequency ranking strategies, 70%.

documents occurs in two stages. First, the documents are ranked according to conceptor frequency. In the example, the maximum frequency would be three conceptors. Second, the documents are ranked according to word frequency. Applying this two-stage strategy, a document with three words from the query, each word originating within one conceptor, will be ranked above a document with ten words from the query, five from each of *two* conceptors.

Experiments have demonstrated that this two-stage ranking on conceptor and word frequency improves performance and, on an average, this strategy gives the best performance of all of those strategies examined.[53] As an example, if one compares the ranking of conceptors with word frequency ranking for the same queries, the results are as shown in Figure 6:

Figure 6

Experimental results: NORIS (8) II
Conceptor and word frequency ranking[54]



For the purposes of this article, there is no need to examine further the merits of retrieval strategies based on conceptors. It is, however, an obvious method for improving the performance of natural language strategies.

In a conceptor-based system, each query corresponds to an idea in the question. This naturally assumes that the user has invested the intellectual effort required to dissect his own question. The system cannot make a similar dissection of a natural language query, since that would necessitate knowledge of the natural language and its semantic content not presently available. Consequently, it is not possible to construct true conceptors without user interaction.

53. *Cf.* BING & HARVOLD, *supra* note 20, at 76; BING, HARVOLD, KJØNSTAD & STABELL, *supra* note 5, at 49; FJELDVIG, *supra* note 20, at 85, 136.

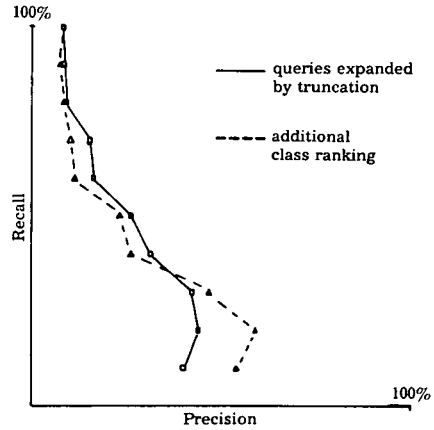
54. BING & HARVOLD, *supra* note 20, at 76.

There is, however, the possibility of using a frequency ranking on the basis of word classes *not* corresponding to true conceptors. In Section VI above, query expansion by mechanical truncation was discussed. When a word is truncated, a word class is created, each member of that class having the first characters in common. In spite of its being created purely on a syntactical basis, that class has considerable synonymity. It was also demonstrated that the performance of frequency ranking increased with query expansion through mechanical truncation. It would be a natural hypothesis to assert that a two-stage ranking, first by class frequency, and second by word frequency, would also improve this result.[55]

Ranking on word classes created by mechanical truncation should not be confused with true conceptor-based ranking. The semantic consistency within the word class is not assured and, of course, there usually will be more word classes than conceptors. These factors introduce additional, and unnecessary, conditions for documents being included in the top-ranked set. One would, therefore, expect results somewhat better than pure word frequency ranking, but somewhat less satisfactory than true conceptor ranking.

Figure 7 illustrates the increase in performance with reference to pure word frequency ranking.[56]

Figure 7
Experimental result: natural language queries
Mechanical truncation and class ranking



55. This may be supported by actual experimental results. Word frequency ranking, based on the number of *different* words from the query which occur in the document, has proven somewhat more efficient than ranking the *total* number of words from the query occurring in the document. FJELDVIG, *supra* note 20, at 78. This result may be because the occurrence of many *different* words from the query indicates an increased probability that many of the ideas in the question are contained in the subject document.

56. *See* Figure 5 *supra*.

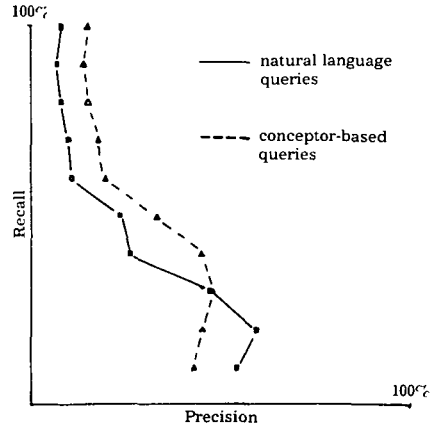
Note that the queries are identical. The increase in performance must be attributed solely to class ranking. For the recall values of 10-50%, the precision values have increased 14% compared to pure word frequency ranking *with* mechanical truncation, and 94% compared to pure word frequency ranking *without* mechanical truncation.

It would be interesting to compare this result with that achieved by an experienced user employing the best strategies available. This, however, would require changing one more factor in the controlled experiment—the experienced user would not use natural language queries but, for instance, would employ a true conceptor-based strategy with explicit synonym specification.

While an attempt was made to include this in the examples given, it should be remembered that due to the limited purposes of the experiment—to provide illustrations for the article—the result may owe as much to the experimenter's subjective choices as to the merits of the two strategies. The results are presented in Figure 8.

Figure 8

Experimental result: natural language queries
Natural language queries compared to
conceptor-based queries



Since the conceptor-based strategy approximates the performance an experienced user may expect from a text retrieval system, the natural language strategy compares very favourably.

VIII. Conclusion

This article has suggested that there is a need to develop a simpler user interface to legal text retrieval systems. Queries in natural

language seem an obvious possibility. Two alternatives for processing natural language queries have been discussed, one based on word frequency ranking and one on vectors. Some experiences with such strategies have also been indicated. Additionally, the possibility of improving performance by mechanical truncation and class-ranking has been suggested.

Due to the small question set and limited resources devoted to the experiments performed, the results should not be regarded as more than exemplary. One should not draw any conclusions from these examples as to the relative merits of the different strategies. Some may argue that this article has already overtaxed the information to be gleaned from these experiments.

While it is not possible to draw conclusions concerning the future of natural language queries, it would seem that the examples presented justify a hypothesis: "It is possible through simple means to increase and, perhaps radically increase, the performance of natural language retrieval strategies." This hypothesis justifies the hope that the possibilities of natural language queries will be examined in more depth.[57]

57. There are reasons to believe that this hope will be fulfilled. As mentioned above (note 47 *supra*), the French Centre d'informatique juridique (CEDIJ) is working on the development of natural language queries, and the Norwegian Research Center for Computers and Law has launched a two-year study of this and related problems. There are also indications that natural language query options are being developed as part of general retrieval systems scheduled for release in the relatively near future.