DOCUMENT CLASSIFICATION, INDEXING AND ABSTRACTING MAY BE INHERENTLY DIFFICULT PROBLEMS

Aviezri S. Fraenkel*

Department of Applied Mathematics
The Weizmann Institute of Science
Rehovot, Israel 76100

Dedicated to the Memory of
Yeshoshua Bar-Hillel

> Words are seldom exactly synonimous; a new term was not introduced, but because the former was thought inadequate: names, therefore, have often many ideas, but few ideas have many names. It was then necessary to use the proximate word, for the deficiency of single terms can very seldom be supplied by circumlocution...The shades of meaning sometimes pass imperceptibly into each other; so that though on one side they apparently differ, yet it is impossible to mark the point of contact. Ideas of the same race, though not exactly alike, are sometimes so little different, that no words can express the dissimilitude, though the mind easily perceives it, when they are exhibited together; and sometimes there is such a confusion of acceptations, that discernment is wearied, and distinction puzzled, and perseverance herself hurries to an end, by crouding together what she cannot separate...this uncertainty of terms, and commixture of ideas, is well known to those who have joined philosophy with grammar; and if I have not expressed them very clearly, it must be remembered that I am speaking of that which words are insufficient to explain.
>
> Samuel Johnson, preface to the "Dictionary", 1755

Abstract. The main features of document indexing are abstracted. It is shown that the easy part of indexing, namely the question whether there is a bounded number of descriptors for indexing a document is NP-complete. Thus even the most efficient algorithm for exact indexing is not, at least at the present time, bounded by a polynomial-time function.

* Partially supported by a grant of Bank Leumi Le-Israel.

1. Introduction

This note is just a little remark, intended to put document classification, indexing and abstracting in their proper niche in the class of problems human beings are interested in, since despite a huge literature on the subject, its true nature has been obscured more often than not. It will be concluded that indexing is probably a very difficult task to perform well. Of course by declaring a problem difficult, it does not simply vanish - unfortunately! However, recognizing the true nature of the difficulty of a problem has important ramifications. For example, it implies that it may be useful to consider certain classes of subproblems, which may be considerably easier, or to devise suboptimal yet easy solutions for the general problem - with bounded error, if possible.

We may divide computational problems, at the present state of human knowledge, into five categories, listed here in the order of increasing difficulty.

I. Problems which can be solved by a polynomial-time algorithm, that is, an algorithm whose number of steps is bounded by a polynomial, such as sorting a file of n real numbers ($O(n \log n)$ algorithm). Any problem which can be solved by a polynomial-time algorithm is commonly called tractable. A problem which can be shown not to have a polynomial-time algorithm is commonly called intractable. See e.g. Garey and Johnson [6].

II. Problems which are likely intractable. To this family belong problems which are NP-complete, NP-hard, Pspace-complete, Pspace-hard or belong to some other class in a hierarchy of complexity classes between polynomial and exponential. An example of an NP-complete problem is the "hitting set" problem: Given a collection of subsets $S_1,\ldots,S_m$ of a finite set S and a positive integer K, does there exist a hitting set $S' \subseteq S$ of size $|S'| \leq K$, such that S' contains at least one element from every subset S_i. Examples of Pspace-hard problems are the games of checkers and go, suitably generalized to an n×n board.

A formal introduction to the notion of NP-completeness can be found e.g. in Garey and Johnson [6]. For our purposes it suffices to point out the following important properties of the collection of NP-complete problems (which currently contain a few hundred members):

(i) There is no known polynomial-time algorithm which solves any single problem in the class.

(ii) The existence of a polynomial-time algorithm for solving any particular problem in the collection would imply that every NP-complete problem can be solved with a polynomial-time algorithm. Hence if any problem in the collection would be proved intractable, then all the collection would be intractable and thus belong to category III below.

It is widely believed that no NP-complete problem can be solved with a polynomial-time algorithm, and hence that all such problems are inherently computationally intractable. Regardless of our beliefs, however, NP-complete problems are at present "practically intractable" in the sense that the best known algorithm for solving any of them is exponential.

III. Intractable problems. For example, Fischer and Rabin [3] showed that deciding whether a given Presburger arithmetic formula is true requires at least $O(2^{2^{cn}})$ operations for some formulas of sufficiently large length n, where $c > 0$ is a constant. Stockmeyer and Chandra [10] showed that certain games are intractable. In fact, the game of chess, properly generalized to an n×n board is complete in exponential time. This implies that there are positions in n×n chess for which the problem of determining who can win from that position requires an amount of time which is at least exponential in n. In chess, checkers and go, the problem which is hard is that of an exact strategy. The cited complexity results do not purport to assert anything about heuristic approaches for playing these games. It is interesting, though, that even restricting attention to heuristics, the workers in the field did not realize that they tackle a difficult problem. Regarding chess, the prediction was made in 1957 that within ten years a computer would be the world's chess champion [8]. When in 1968 Levy [7] voiced the opinion that within ten years there would be no chess program that would beat him, some of the world's leaders in artificial intelligence were quick to state categorically that Levy was totally wrong. A bet was arranged and until 1974 a number of the leading workers in mechanical chess joined in the bet against Levy.

IV. Undecidable problems, that is, problems which can be shown not to have an algorithm for their solution. An example is provided by Hilbert's tenth problem, posed in 1900, which asked whether there exists an algorithm for determining whether an arbitrarily prescribed polynomial equation with integer coefficients has integer solutions. Yuri Matijasevič, building on earlier work of Julia Robinson and Hilary Putnam, gave a negative answer to this problem in 1970: Hilbert's tenth problem is undecidable. See e.g. M. Davis [2].

V. Problems which are not well-formulated. Certain aspects of mechanical translation, computational linguistics and information retrieval belong to this category.

The lack of formalism for problems in category V sometimes creates the illusion that these problems are easy. Thus, paradoxically, some of the most difficult problems are considered easiest. This is what happened to mechanical translation in the 50's. It took a good number of years until a large number of the workers in mechanical translation realized that they tackle a difficult problem! Another case in point is classifying, indexing or abstracting of documents, as we suggest more formally below.

2. Modeling the Process of Document Indexing

Some people view indexing as a trivial task, and lightheartedly undertake indexing of large bodies of texts. More cautious people think it advisable to make some experiments before using it on a large scale. Having done this while designing a retrieval system at Oxford for British case law, Tapper [11] wrote: "The plan...was to index all the materials in a given area for both legal concepts and fact situations, and then to attempt sample searches. The area selected was the law relating to the admissibility in evidence of confessions. It was felt that this area was well-defined, self-contained, and, in this country at least, of relatively small compass.

Some preliminary indexing was carried out on this basis. It proved to be disastrous. For one thing, the legal concepts, so well-defined in the books, proved to be much less so in practice. It was found that the indexers were quite inconsistent in their attempts to index the same document, and that even the same indexer was liable to index a document differently at different times. This was especially true when it was presented in a different context".

Document indexing, as used here, means the assignment of a bounded set of terms to represent each document. This assignment depends on global considerations pertaining to the entire collection of documents to be indexed, and, in particular, on every document in its entirety, rather than on local considerations, i.e., considerations depending only on the word currently being examined, sometimes including its immediate neighborhood. The assignment can be made by man or machine.

In particular, full text dictionaries, concordances, KWIC (Key-Words-In-Context), or KWOC (Key-Words-Out-Of-Context) indexes are excluded from the present discussion since the exclusion or inclusion of a word in such indexes is decided upon locally, by comparing the word currently examined with an a priori constructed list of words (which might be empty) - either a list of words to be excluded from or a list of words to be included in the index. For similar reasons, citation indexes are excluded.

Manual indexing consists of the following steps:

(1) Read the current document to be indexed.

(2) Decide which are the topics (also called subjects) discussed in the document. These topics are not necessarily given a priori; they may be updated and their meanings defined more sharply as we proceed with the indexing of the documents in the collection.

(3) Separate the essential from the inessential, retaining the former and discarding the latter.

(4) Assign a number of index terms to the document,

either from a predetermined set of index terms, usually called descriptors, or from the document itself, in which case the selected terms are usually called keywords or uniterms. The index terms may actually be short expressions, such as word-pairs or triples.

It was concluded in [4] , that the general problem of indexing is ill-defined, and that, wherever possible, it is best to avoid it by both man and machine (replacing it e.g. by full text retrieval).

The common feature of document classification, indexing and abstracting is that the document is replaced by a set of clues. For indexing, the clues are index terms; for classification they are numerical codes for the different classes and subclasses a document belongs to, such as the AMS Subject Classification Scheme; and for abstracting the clues are a small number of sentences. Thus also for classification and abstracting, steps (1)-(3) are valid. For classification, step (4) is replaced by: assign a number of class codes to the document, and for abstracting it is replaced by: assign a number of sentences to the document. In view of the similarity of these processes, we shall restrict attention in the following to indexing, and leave it to the reader to supply the simple additional arguments needed for adapting the indexing model to that of classification and abstracting.

Step (2), depending on the nature of the text, is usually difficult and belongs to category V. Also step (3) is ill-defined. We show below that step (4), which has traditionally been performed by either man or machine, and is perhaps considered the "easy" part of indexing, is NP-complete. We proceed by modeling the indexing process more closely.

Suppose a collection C of documents is given which have to be indexed. Usually a uniform bound K is given, such that each document has to be indexed using no more than K index terms. Let D be any document from the collection C. Suppose that it contains a set T = T(D) of "semantically non-identical topical words". Thus "fable" and "fables", which are morphologically distinct, are probably semantically identical in most texts. If both appear in D, one "standard form" of them (say "fable") is put into T. On the other hand, if "game" and "play" appear in D, they will probably be put into T as distinct words, since, though semantically related, they are not semantically identical. Also if a homograph like "game" appears in D with different meanings such as game of chess, game-act for hunters, it is put into T(D) as game(a) and game(b).

We now put each $w_j \in T(D)$ into a set S_j of "semantically related words", or "topics", selected from among all the words in a set which includes at least all the words occurring in all the documents comprising C. In the above example, there will be a set S_j containing "game(a)" (which may contain "play") and a set S_k for "play" (which may contain "game(a)"). Thus S_j and S_k may have a nonempty intersection. Also "game(b)" may be in both the set of "deer" and that of "hunter". To give a concrete example of the multitude of situations that may arise, a collection of Pennsylvania Health Laws contains a subcollection of statutes about "treatment of infants in hospitals". Two of the statutes use the words "institution" and "penitentiary" as synonyms for "hospital" [4, Appendix II] . So these two statutes induce generation of "hospital(a)", "institution(b)" and "penitentiary(c)", such that the set S_j containing "hospital(a)" also contains "institution(b)" and "penitentiary(c)".

Of course the situation is actually more complicated. For example, a local feedback method produced the searchonym "slope" of "resistance" in a search on "negative-resistance transistor". "Slope" and "resistance" indeed turned out to be searchonymous in the sense that also "negative-slope (transistor)" retrieved documents relevant to this topic [1] . Should the "local connection" between "slope" and "negative" be reflected in their sets S_j?

Be this as it may, the creation of the topics S_j corresponds to steps (1) and (2) of the indexing process which belongs to category V, since the concept of "semantic value" is not well understood and therefore has not yet been well formalized. However, we shall assume that the topics S_j were created somehow, perhaps by using a thesaurus or some ad hoc approximation from among those used to solve this problem in practice, and not necessarily by the process indicated above. We emphasize that the topics S_j consist of words with as unique and unambiguous a meaning as possible. For convenience one writes, in practice, say "game-playing" instead of "game(a)", and "game-hunting" instead of "game(b)".

Step (3) is done by discarding topics S_j deemed inessential. A special case is, of course, when nothing is discarded.

3. Document Indexing Is NP-Complete

Suppose that the topics remaining after applying step (3) to the indexing of a document D are $S_1,\dots,S_m$. The remaining step (4) of the indexing process is to select a set S' of representatives, with $|S'| \leqslant K$, such that S' intersects each of the S_i. But even the problem of deciding whether such S' exists is the NP-complete hitting set problem.

Incidentally, if no set S' of size at most K is found which intersects all the sets S_i, then the indexer goes back to step (3) and reclassifies some of the essential topics as inessential. Alternatively, he may replace some intersecting topics by their unions. Either procedure will normally lead to loss or distortion of information.

We close with a few remarks.

(i) The problem of constructing a hitting set of minimal size is NP-hard, which is at least as difficult as any NP-complete problem.

(ii) An NP-complete problem may be solved approximately by some suboptimal yet polynomial algorithm. In the present case, let $S = \bigcup_{i=1}^{m} S_i = \{w_1,\dots,w_n\}$. Suppose that we already constructed a partial hitting set $S'' = \{w_{i_1},\dots,w_{i_k}\}$. Let $w \in S - S''$. Then w is adjoined to S'' if and only if $w \notin \cup S_j$, where the union is taken over all S_j containing some w_{i_ℓ} $(1 \leqslant \ell \leqslant k)$. (If w is adjoined, a sequential scan of S'' may enable removal of some w_{i_ℓ} without impairing the hitting power of S''.) This process is clearly bounded by a polynomial-time algorithm, but it may produce a hitting set whose size is far from minimal.

(iii) An NP-complete problem may of course contain subfamilies of problems which can be solved by a polynomial-time algorithm. For example, if indexing is request-oriented rather than document-oriented (see e.g. Soergel [9, B5.2] , the problem becomes bounded, and if the number of anticipated requests is not too large, the problem may become quite manageable.

(iv) The general indexing problem, however, is unbounded. A real-life illustration of this is given e.g. by the "Responsa Retrieval Project", whose database consists of "legal cases" [5] . Searches run to date were in the areas of law, history, economics, philosophy, religion, sociology, linguistics, musicology, folklore, personages, realia, geographic sites, saints, scholars, wars, kings, marriage and death customs, recipes, taxes, medicine, education, dance and ballet, abortion, mediaeval geometry - to mention just a few. If this does not convince the reader that requests cannot be anticipated, the following should: Every now and then a request is received to retrieve certain passages or citations on all occurrences of an idiomatic expression and its variations, according to the interest of the submitting person at any given time. These are easy to do if the full text is machine readable, but impossible if the difficult task of general document-indexing has been attempted.

Conclusion

It has been shown that if the indexing process is modeled appropriately, its easy part is NP-complete; its difficult parts belong to the category of the most difficult problems, those which are not well-defined. The implication is that indexing exactly is probably a very difficult task. This indicates that subproblems for which there are polynomial solutions should be sought, as well as efficient heuristics, with bounded errors, to attack the general problem of indexing. Similar statements hold for the problems of classifying and abstracting documents.

This note uses only the simplest facts from complexity theory and the simplest facts from documentation theory. If there is anything new in the note, it is only in using the former to illuminate the latter.

References

1. R. Attar and A.S. Fraenkel, Local feedback in full-text retrieval systems, J. Assoc. Comp. Mach. 24 (1977), 397-417.

2. M. Davis, Hilbert's tenth problem is unsolvable, The American Mathematical Monthly 80 (1973), 233-269.

3. M.J. Fischer and M.O. Rabin, Super-exponential complexity of Presburger arithmetic, in: Complexity of Computation (R. Karp, Ed.),SIAM-AMS Proc. 7, American Mathematical Society, Providence, RI, 27-41, 1974.

4. A.S. Fraenkel, Legal information retrieval, in: Advances in Computers (F.L. Alt and M. Rubinoff, Eds.), Vol. 9, Academic Press, New York, NY , 113-178, 1968.

5. A.S. Fraenkel, All about the Responsa Retrieval Project you always wanted to know but were afraid to ask, Expanded Summary, Proc. 3rd Symp. Legal Data Processing in Europe, Oslo, 1975, Council of Europe, Strasbourg (1976), 131-141. Also appeared in Jurimetrics J. 16 (1976), 149-156 and in Informatica e Diritto 11 (1976), 362-370.

6. M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, CA, 1979.

7. D. Levy, Chess and Computers, Computer Science Press, Woodland Hills, CA, 1976.

8. H.A. Simon and A. Newell, Heuristic program solving: the next advance in operations research, Oper. Res. 6 (1958), 1-10.

9. D. Soergel, Indexing Languages and Thesauri: Construction and Maintenance, Melville, Los Angeles, CA, 1974.

10. L.J. Stockmeyer and A.K. Chandra, Provably difficult combinatorial games, SIAM J. of Computing 8 (1979), 151-174.

11. C. Tapper, British experience in legal information retrieval, Modern Uses of Logic in Law (1964), 127-134.