Natural-language processing and automatic indexing

C. Korycinski and Alan F. Newell

The task of producing satisfactory indexes by automatic means has been tackled on two fronts: by statistical analysis of text and by attempting content analysis of the text in much the same way as a human indexer does. Though statistical techniques have a lot to offer for free-text database systems, neither method has had much success with back-of-the-book indexing. This review examines some problems associated with the application of natural-language processing techniques to book texts.

Introduction

Indexed books represent the most common example of a subject information retrieval system. Problems associated with both the quality and cost of human intellectual indexing have made automatic indexing an attractive goal. During the 1950s and early 1960s there was great optimism that this goal would be readily achieved. However, many of the ideas proposed for text analysis, indexing, and machine translation have proved to be simplistic; the complexity and richness of natural language defeated these workers.

The indexing problem can be stated quite simply. It requires a content analysis of documents followed by characterization of such content by descriptive labels. This requires the following steps:

(a) conceptual analysis of the significant content of a document;
(b) expressing the analysis by a set of words or phrases to represent the subject;
(c) translating the relevant subject descriptions into standard language;
(d) arranging the standardized subject descriptions according to the syntax of the indexing language used in the information system.

The ease with which the problem can be expressed does not imply ease of solution. For example, a human indexer reads a section of text and can determine its subject content (the ‘aboutness’ of the text) without conscious thought—reading the text normally involves understanding, which implies that content analysis has been accomplished. This can best be seen when the smooth flow of this process is interrupted, as may happen when the subject matter of text is both complex and unfamiliar. The indexer then has repeatedly to re-read and examine text until understanding is acquired. Then subject analysis is the relatively easy matter of selecting the most appropriate indexing phrase. For this solution to be a model for a computational approach to indexing, however, it must be expressed in algorithmic form: a major problem.

To investigate how it has been tackled, a brief look at some applications of statistical indexing techniques will be followed by a more extensive survey of natural-language processing (NLP). No attempt will be made to survey the areas involved in expert systems and of knowledge representation; Walker’s *Knowledge Systems and Prolog* provides a useful introduction to these areas.

Database indexing

At present the literature dealing with indexing topics and concepts is mainly concerned with retrieving information from texts stored in databases, whose function is usually to inform the user of some easily quantifiable fact (e.g., the number of people over a certain age employed by a company) and also to point the user to some relevant document. The pointing function may be either an abstract, or keywords removed from either the original text or the abstract itself.

In neither case is the text manipulated by the system similar to a book. Dealing with such texts is essentially easier than dealing with a book, for the following reasons: the texts normally involve only several sentences as opposed to tens or hundreds of thousands of words; they are usually in the form of abstracted data (with a high proportion of keywords present), and large numbers of significantly different texts are present in the database, and measures of dissimilarity and similarity between texts are thus normally easier to quantify than for a book.

Comparisons between manual and automatic indexing systems show further data on the difficulties of indexing books. Cleverdon noted that:

(1) ‘if two people or groups of people construct thesauri in a given subject area, only 60% of the index terms may be common to both;
(2) if two experienced indexers index a given document using a given thesaurus, only 30% of the index terms may be common to the two sets;

(3) if two intermediaries [person who undertakes a database search on behalf of the user] search the same question on the same database on the same host, only 40% of the output may be common to both searches;

(4) if two scientists or engineers are asked to judge the relevance of a given set of documents to a given question, the area of agreement may not exceed 60%.

Although these are generalizations, and there are likely to be areas where the performance will exceed these figures, in others it will be considerably worse.

Cleverdon suggested that many of these problems could be solved by eliminating the thesaurus and using instead the title or abstract of the article in free-text form; an approach which has had some success, and results from the work of Salton seem to show that performance exceeding that of manual indexing can regularly be achieved for many users, though Blair and Marron come to very different conclusions with the same data, claiming that fewer than 20% of relevant documents are being retrieved.

Artificial Intelligence (AI) approaches to indexing

Much of this work has been shown to be adequate for abstracts, particularly scientific or technical. Indeed, all current automated text-indexing systems are based largely on such statistical work, and hardly show even moderate sophistication in attempting any form of content analysis. These techniques are less useful when applied to longer texts where keywords may form poor identifiers for the subject matter and recourse still has to be made to the intellectual (human) indexing process to identify correctly the 'aboutness' of a particular section of text, though even this may prove inadequate for research purposes. For example, when it is necessary to determine whether a particular idea has been put forward by a given researcher it is often found that indexes, to books or to databases, form poor tools for solving this problem.

Content analysis

Despite the incursions of a variety of non-textual forms requiring indexing (for example, films and music), the bulk of the material that indexers are likely to deal with is the precursor of books—galleys and paste-ups. To mimic human intellectual analysis of these texts it is necessary to determine their content by attempting content analysis that implies comprehension by the program. It may seem curious that so little progress appears to have been made in content analysis of texts, when human indexers do such analysis with no apparent effort—syntax analysis; semantic analysis; and discourse analysis.

Syntax analysis

If the overall objective is to determine the intellectual content of the text, then a sub-problem presents itself when we have to determine the meaning of a sentence. In most systems this means that sentence structure must be determined and possibly regularized in order to simplify the subsequent semantic analysis, which can now be designed to work with less rich forms of words—and therefore those that are more amenable to computational processing.

According to Grishman, syntax analysis performs two main functions in analyzing natural language input:

(1) determining the structure of the input. Syntax analysis should identify the subject and objects of each verb and determine what each modifying word or phrase modifies. This is most often done by assigning a tree structure to the input, in a process referred to as parsing;

(2) regularizing the syntactic structure. Subsequent processing can be simplified if we map the large number of possible input structures into a whatever, despite the amusing/apocryphal stories of the results of computerized content analysis—for example, 'the editors of the new Macmillan Guide to Britain's Nature Reserves do realize the mammoth is extinct, even though it appears in the book's index. An over-zealous computer, working though the text, seized upon a sentence beginning: 'This guide is a mammoth work ...'.

This apparent lack of progress implies lack neither of effort nor of desire to attend to needs expressed by those involved in the literary field. Many of the problems are well known, if not well understood, for it is the challenge of natural-language processing that a text in its entirety may be less ambiguous than its individual parts. So a natural-language processor must resolve this ambiguity by applying context to select from the various possibilities available, which are governed by the lexicon and grammar. This is hard because the text may contain the information required to resolve ambiguities, or it may be implicit in the shared background of the author and the reader.

To see more clearly the problems that arise when we look for meaning in text, three stages of text analysis will be examined. These (1) are smaller number of structures. Some material in sentences can be, and usually is, omitted without changing the meaning: e.g., 'She earns less than Mary' [earns]. By restoring the omitted text (shown in [ ]), the sentence structure can be regularized. Other transformations can relate sentences with normal word order to passive and cleft constructions: these will allow subsequent processing to concern itself with a much smaller number of structures. Thus, although there is no absolute need for regularization, its absence will require usage of more complex semantic rules.

(2) To determine whether regularization, or even syntactic analysis prior to semantic analysis, is required,
Computational linguists have looked at the work of psychologists interested in the way people deal with unusual or abnormal sentences, because such sentences may produce errors giving some understanding of how people deal with sentences in real life. It may be argued that people can use syntactically correct material which lacks any meaning; for example: 'Twas brillig, and the slithy toves'. Here syntax is the only factor giving an appearance of intelligibility. This shows that sentence parsing by humans can take place on the syntactic level with no need for semantics. Sentences which are ungrammatical, but clear in their intended meaning, for instance 'She like go holiday.', show that semantics may drive an adequate interpretation of a sentence when syntax is poor.

Thus we have two strategies used to give meaning to sentences: a syntactic-driven strategy which may be used when semantics are weak or non-existent; and a semantic-driven strategy when syntax is weak. Both methods may be used to extract meaning from a sentence if sections are not clear on the first reading.

In other attempts to explore ways of dealing with sentences in real life, psychologists have been interested in sentences where the reader is 'led up the garden path' in assuming that the sentence means one thing, then when the last word or two is read, sees that his assumptions were erroneous. An example of this is: 'The horse raced past the barn fell.', where the reader thinks the horse was racing past the barn—until the last word is met, then backtracking occurs in an attempt to reanalyze the meaning of the sentence, which is interpreted in the sense, 'The horse [which was] raced past the barn fell.' Marcus attempts computer-simulation of the human approach to this problem, and hopes to obtain insights into strategies used for sentence parsing. This should give not only an understanding of how people interpret the written word, but also possible strategies to be adopted by a computer/parsing program.

**Parsers**

Historically, parsing algorithms can be divided into three sections: context-free parsers (late 50s and early 60s); transformational parsers (late 60s); and augmented context-free parsers such as the augmented transition network (70s— ).

**Context-free parsers**

A context-free parser attempts to decompose a sentence by successively applying a series of derivations such as:

1. sentence (S) = noun phrase (NP) + verb phrase (VP)
2. noun phrase = article (T) + noun (N)
3. verb phrase = verb (V) + noun phrase.

With 'The girl had a lamb' the mapping would result in:

```
NP     VP
|      |
|      |
|      |
T      N     V     T     N
```

Applying rule 1
Applying rule 2
Applying rule 3

This type of parsing is known as bottom-up; the parse is started at the lowest level, that of individual words, and then built up towards a more complex construction. It could, however, start at the top of the tree and attempt to fit the structure to the sentence. Thus the task is to find a set of rules in the grammar of the language, which, correctly applied, produces the sentence being parsed. This is the top-down approach, always implemented with backtracking as a large number of possible parses at one level of the analysis turn out to be unsuccessful when the next level is attempted. The program must be able to retrace its steps to a previous stage and re-commence an alternative parse.

There are advantages and disadvantages in both approaches. A top-down approach tends to be a poor choice for ambiguous sentences; a bottom-up approach reduces the possible number of interpretations the program has to consider. Thus, if ambiguities are not a feature of the document, a top-down approach would give a more efficient program.

Already many difficulties are manifest in this system; the grammar rules given are too simple for any real life situation, and extensions have to allow for more complex and variable sentence constructions. This may result in an unacceptable number of possible parses: for example, the sentence: 'Gravely concerned with spreading racial violence, President Kennedy used his press conference to issue counsel to both sides in the struggle' produced 136 possible analyses. An excessive number of possible solutions leads to the abandonment of this parsing strategy and the adoption of transformational grammars.

**Transformational parsers**

Harris proposed the theory of linguistic transformation, elaborated by his pupil Chomsky, which attempts to generate the deep structure of a sentence, representing its syntactic and semantic interpretation. Thus a sentence such as 'The boy broke the window' can have a number of transformations:

1. The window was broken by the boy (passive transformation);
2. It was the boy who broke the window (cleft transformation);
All these sentences have constituents in common (most of the transformations are clearly paraphrastic). If this underlying structure—the deep structure—can be uncovered, the difficulty of ascertaining the meaning may be reduced. In the case of the sentence, 'The boy broke the window', deep structure might be represented as:

```
S
  |  
NP     VP
  |  
T      Aux
  |    T
N      N
```

that is, a sentence consists of a sequence of: noun phrase, verb, noun phrase. And a noun phrase consists of: article, (one or more instances of) adjective, noun.

This style of transition network is recursive; thus, repetitive patterns, for example a series of adjectives immediately preceding and modifying a noun, are catered for without recursion having to be implemented over the whole network. The TN can perform tests that govern the transition from one state to the next and can be programmed to query sentences where grammar (such as agreement between verb and noun) might be suspect. So far, this is simply another variant on a regular grammar, but it has the advantage of recursion and of being represented in the form of a digraph, or directed graph, consisting of nodes joined by arcs. The arcs are labelled by words or word categories; for example, noun (N) or noun phrase (NP), and it is the task of the parser to traverse the graph from node to node. In this example it can only move from node S1 to S2 if the next item in the parse can be identified as a verb. Representing a grammar as a digraph enables formal mathematical operations to be performed on the graph, as well as defining its properties.

An augmented transition network (ATN) consists of this system with the addition of a system for generating deep structure and allowing grammatical constraints to be imposed, so that it has the important property of performing both surface and deep analysis simultaneously. Generally, an ATN parser works in a similar way to the top-down context-free parser and uses the conventional backtracking routines to escape from possible parses which proved untenable.

Examples of working grammars based on the ATN model include the Linguistic String Parser,25,26 DIAGRAM,27 Proto-RELADES,28 and LAS.29 A comparison between an ATN and a two-stage parser—
known as the sausage machine—may be found in the work of Wanner.30

**Context-sensitivity**

Despite considerable progress in parsing many major problems still remain, at least a proportion of them requiring context-sensitive grammars for their probable resolution in terms both of parsing and semantic processing.

An example of the type of difficulties which have proved hard to resolve can be gained by studying problems associated with parsing compound nouns. The following closely follows Sparck Jones31 who states that three processes apply to compound nominals: they have to be

1. bracketed: wicker (bread basket) vs (wholemeal bread) basket;
2. lexically disambiguated: bread = loaf vs bread = money;
3. given a meaning characterization: basket for bread vs basket with bread.

She states, 'Various strategies, or rather non-strategies, have been adopted to deal with compound nouns. One, for example, is to put compound nouns in the lexicon. A second is to assume that restrictions on the possible interpretations of the constituents, for a specific universe of discourse, will force an appropriate interpretation on the whole. A third is not to attempt a full characterization of the whole. And a fourth is to disregard elements which cannot be processed. These are all unsatisfactory, for obvious reasons. We cannot put all the compounds, even for a limited domain, in the lexicon. We cannot rely on sense restrictions to give us an acceptable interpretation. We cannot assume that an explicit characterization is unnecessary. And we cannot throw parts of the input away.'

Furthermore, even with a considerable quantity of domain-specific knowledge, there must be differences in interpretation32 of:

1. 'These border plants are expensive'—implying that the set of plants being considered are those normally found in a border; and
2. 'These border plants have to be replaced'—that is, these plants in the border have to be replaced. The location of the plants is under consideration irrespective of the actual type of plant found there.

At present no system can cope with such nuances of meaning. We could take the matter still further. If we were to use the same core of words, but applied to a different domain: 'The IRA have done so much bombing that all the border plants have to be replaced' we have a situation where 'border plants' could apply to plants which grow, but is more likely to apply to factories (= plants) located at the Eire/N Ireland border. Disambiguation here would be a far from straightforward task.

The problem lies in parsing having been syntax-driven; most parsers are based on context-free grammar, whilst the underlying problems of ambiguous structures essentially need semantic guidance not only from the context but also by using referents from domains of personal experience of the reader.

However, any conclusions formed from considering these facts can only be tentative, and influenced by the domain of discourse of any particular text. For example, much work has been done on the automatic indexing of chemical texts.33,34 The interest for many potential readers in this field is likely to be bound up with chemical structures, physical processes and chemical reactions. By their nature these are domain-restricted, and, due to the logical way chemical names are built up, eminently suitable for the application of the process of stripping off prefixes and suffixes will be able to identify the structure, elements or constituent groups of the compound under discussion—followed by statistical analysis. Few other subjects will be so amenable to this type of analysis.

**Indexing quality**

One also needs to ask 'How good an index do we need?' These ambiguities may simply be ignored, and reliance placed on other, clearer, sections of the text to fill in the doubtful areas, and/or human intervention for disambiguation. This crucial question could only be answered according to individual needs. Factors to consider include the costs of checking entries manually for possible errors; of running a program that produces fewer errors than the one currently in use; and of errors—if the error rate is low enough users are likely to be satisfied with the performance.

At present, linguistic analysis of text by computer is not far enough advanced for such questions to be answered, but if we draw on results obtained from statistical, non-linguistic, indexing systems, we can surmise that 100% accuracy is not needed for acceptable performance.35 The caveat should be added that, as with most research on indexing, these results apply to large databases, and may well be inapplicable to texts such as books.

**Semantic and discourse analysis**

The object of parsing a sentence is only the first stage in a strategy to attempt to find some way in which the meaning of a sentence (its semantic content) and the meanings inherent in a series of sentences (the object of discourse analysis) could be represented in a computer with the aim of manipulating concepts and ideas inherent in the text with the same facility as the computer can manipulate characters. It is only when text can be understood by a computer (by 'understood' is meant some form of representation giving the appearance of
understanding so that—for example—questions which involve concepts not directly stated in the text could be posed to the computer with satisfactory answers being received) that we can hope to simulate the intellectual actions of a human indexer.

As in the case of syntactic analysis, a realistic solution to the problem of semantic analysis must imply simplification—in which case we need to specify a formal language on which to map natural language. Furthermore, in the case of indexing, we must decide also whether any other specialized needs must be met either in the form of extra semantic information, or, conversely, whether all the information obtained from a full semantic analysis would be required for our purposes. It has been suggested that the task is to examine the semantic component of

(1) methods of identifying meaningful units in the input document;
(2) the derivation of document descriptions from these;
(3) the construction and use of classifications and other forms of index language organization.

Unfortunately none of these aims is achievable without the addition of substantial quantities of outside knowledge. Part of this may come from the text itself, but many more assumptions of knowledge common to the author and the intended audience are made than may initially be obvious. Difficulties arising here may be illustrated by reference to a straightforward description: ‘Just before dawn, the Valiant sighted the Zwiebel and fired two torpedoes. It sank swiftly, leaving few survivors.’

Understanding this simple story is more than understanding the sum of its parts. We have to consider what the ‘it’ refers to (sometimes called anaphoric resolution). There are four possible antecedents: dawn; Valiant; Zwiebel; torpedoes. Straightforward syntactic analysis enables us to exclude torpedoes (incorrect singular/plural agreement) and semantic analysis excludes dawn, but only general world knowledge enables us to surmise that Valiant and Zwiebel must both be ships, and there is a causal relationship between firing torpedoes and sinking another ship.

This type of implicit information, inherent in all texts, must be recovered to enable an automated indexing system to make an intelligent assessment of the suitability of particular phrases or concepts as index entries. This search for meaning is at the root of both semantic analysis (an attempt to determine the meaning of a sentence) and discourse analysis (considering a longer text) and this aspect of computational linguistics has so far proved intractable. Most attempts to solve the problem have been restricted in their domain as a consequence of the method used to overcome the problem, which is attempting to assemble all the knowledge needed to answer any particular problem. This knowledge could be used at the syntactic analysis stage by removing some syntactically plausible but semantically unlikely parses.

The organization of such quantities of knowledge is no trivial undertaking, though it has been attempted. One notable method was that of frames, based on the observation that new knowledge tends to be assimilated by identifying at least a portion of it with patterns inherent in pre-existing information. These frames may be:

(1) Prototypes, describing classes of objects, actions, or situations: e.g., chairs, running, attending the theatre. They form a general framework of expectations—so whether one goes to a theatre in Edinburgh or Vienna, the general outlines of what one does, how one behaves, and the point of the visit remain broadly the same. Consequently the ‘going to the theatre’ frame is likely to remain valid.

(2) Instance frames, describing particular instances of action, etc., e.g., yesterday’s visit to the theatre.

There are obvious similarities between frames and context-free grammar: prototype frames correspond to the rules of the grammar itself and instance frames to detailed grammatical knowledge. As we shall see later, the concept of frames may also be considered as a context-free description of world knowledge.

Some frame-based knowledge representation languages (KRL and FRL), have been developed and this type of work has lead to the development of scripts that attempt to contain the meta-knowledge (knowledge implicit in a given situation) needed to understand a given simple narrative. Thus, if a computer program has the appropriate script for a particular scene, it is possible to arrange for it to query anomalous situations, ask apparently intelligent questions and make responses to queries that involve concepts not directly mentioned in the text.

The classic example of a script is that of a restaurant where information is present about the characters (waiters, chefs, customers), scene (location of the restaurant, table arrangement, food preparation area) and typical actions (ordering a meal, paying for a meal). Anaphoric resolution thus becomes easier. For example: ‘The customer told the waiter that he would like some wine.’ We know that waiters serve food and drink, and customers order it, so it is clear that he must refer to the customer, not the waiter; whereas ‘The customer told the waiter that he should bring the wine list’ would result in he applying to the waiter, as it is the job of the waiter to bring this and similar items.

This example shows that scripts cannot be used in novel situations: for these plans can be used, with attempts made to construct a causal chain. This is no trivial problem: many of the required connections may
be implicit, rather than directly stated, in the text. However, a system has been implemented by Wilensky as PAM (plan applier module) and as a simplified system called micro-PAM. These systems produce a complex series of inferences that raise the problem of which inference should be followed.47

The database of rules and scripts needed is very large even for what would normally be considered as fairly trivial situations, such as the restaurant scenario. A system requires considerable development effort to 'understand' even those episodes which consist of relatively few sentences, even only one sentence. Avoiding generalization, it can be seen that if a large number of sentences deal with essentially the same concept, then processing of the information becomes easier than if rapid changes of content in the sentences necessitated the implementation of a large number of plans in the system. A natural-language indexing system might use semantic analysis of text to produce suitable index entries, but only if the domain of discourse was very restricted. Unfortunately, this is also the area where statistical analysis and word stemming techniques can be quite successfully applied; the more diffuse areas prove intractable both for statistical analysis and discourse analysis. The former is unsuited to this work, as keyword extraction may have little relevance to the production of useful index entries, and the very vastness of the domain in the latter poses difficult problems despite efforts to produce generalizations of text.49 One way of dealing with this problem is to develop a system, goal subsumption,49 which tries to plan for the satisfaction of several goals at the same time. In spite of all these strategies, however, the problem in essence remained unsolved, particularly in attempts to transfer laboratory results to the domain of ordinary texts.

Conclusion

Implicit in both semantic and discourse analysis, when tackled by creating plans and scripts, is a particular world view: that rules can be stated and scripts designed which, by logical extension, may be made applicable to new events, making plausible assumptions about novel situations possible. This view has its critics,48 who deny the presence of an objective reality outwith the confines of a specific environment of concerns, actions and beliefs. This stance implies that questions cannot be answered without reference to their context. Once this alternative framework is accepted, the underpinnings of the frames, scripts and plans approach to discourse comprehension are removed and any attempt to apply such methodology must usually fail. This will especially be true where superficial similarities exist between the script and the text of a particular event, but with no cognizance being taken of the wider circumstances in which the script is being used. These are the areas where human ability to have an overall view of the text, and thus be able to select and reject selectively, shows the limitations inherent in present computer-based linguistic analysis systems.

Despite these problems, the only way forward for long-term researchers into indexing is to make use of computational linguistic techniques. For as Borko said, "During the past decade a great deal of progress has been made. . . . Parsing deserves our attention, for it is an essential element of question answering, fact retrieval, decision support and expert systems. If information retrieval is to progress beyond the simple retrieval of bibliographic references, it will need to analyze and restructure text, and it will need effective parsing procedures."52

Present-day working commercial indexing systems, however, are all based on statistical analysis. Attempts to improve the efficiency of these by adding parsing and semantic analysis using partial parsing techniques, whereby a mixture of (for example) stemming, statistical and syntactic analysis techniques is used53,54 have been no more effective than applying non-linguistic, statistical models55 to these systems. In indexing books, as opposed to abstracts, these techniques produce too many keywords to be of use.56

The final picture is far from clear. Statistical techniques have a good track record, perform effectively enough in their domain and are commercially available; but whether they can be improved significantly for general, as opposed to specialized, use is a moot point. Furthermore, solution of the indexing problem implies comprehension of text. Textual analysis has as yet proved inadequate for the task, though ultimately it is seen as the only way to extract complex concepts from the texts under consideration.

The best that can be hoped for at present seems to be a computer-assisted indexing system that would attempt to guide and assist the indexer with a variety of both linguistic and statistical methods to analyze the text on the basis of the indexer's previous interaction with the system; i.e., what relationship(s) can be found between the index entry and the text of the book? The creation of syntactic structure could then be tackled by attempting to match previous index entries with the current text and inserting the appropriate references (see, see also, see under, plus the old reference).

This is also a hard problem, but has the advantage that perfection is not being aimed at. The goal can be set by the level of compromise acceptable at any given time, and deficiencies in the system can be compensated for by human intervention. This is an area where techniques of knowledge representation may be used, and is typified by such systems as the Indexing Aid Project,57 which uses a frame-based knowledge representation language (FrameKit) to help human indexers to index medical literature. Systems such as these may indicate the likely direction book indexing will take in the future.
NATURAL-LANGUAGE PROCESSING AND AUTOMATIC INDEXING

References


11. Weinberg, B. H. *ibid*.


32. Sparck Jones, K.: *ibid*.


Librarians index legally

It is not only indexers who index. Librarians—particularly those in special libraries—know something of the art; and to judge by some articles in a recent issue of *The Law Librarian,* are well qualified to make a very good job of it.

The first of these articles, by Nigel Smith, describes *Legal Journals Index,* which covers all legal journals in England, Scotland and Northern Ireland. The writer is one of the originators of the project, who also produced a legal thesaurus for the purpose. The indexers quickly realized that consistency of terms was a vital factor, and the description of their examination of existing thesauri and their methods of producing their own is of considerable interest to all legal indexers.

The second article, ‘Some issues in legal indexing’ by Andrew Green, is more general and reflective in character, but still succeeds in clarifying some concepts which may otherwise be only dimly perceived in day-to-day work: for instance, the two rival traditions—the ‘classified’ method (which offers a small number of main headings with all others grouped under them), and the ‘specific entry’ method. A surprising fact (among others) is the long history of legal indexing, with examples quoted from 1657 and 1705. The survey comes up to date with comments on WESTLAW and LEXIS database systems.

Finally there is a paper about Butterworths’ ‘Current Status’ service covering E. E. C. legislation—notorious for its volume and complexity. Of the three annual volumes (which contain summaries and reference detail rather than full text), the third is an alphabetical subject index. The most interesting and admirable thing about this publication is the intensive market research that preceded the launching of the project. Practitioners and librarians were canvassed to discover their needs and how they would set about using such a service. Their views were taken into account at the planning stage.

All three articles are of considerable interest to indexers in general; but to legal indexers in particular, the appeal is that of voices speaking articulately about problems peculiar to this area of indexing.

B. J. S.

---

*The Law Librarian* 20(1) April 1989

The Napoleon of information retrieval

The Code Napoléon (i.e. the Civil Code), as printed in 1810 contained less than 120,000 words, and could be carried in the pocket. Its 2,281 Articles, numbered consecutively, were grouped into 3 Livres, containing 11, 4 and 20 Titres; and a 60-page Table des matières enabled the reader to turn up any point in a moment. A French citizen had no right to claim ignorance of the law.


The Indexer Vol. 17 No. 1 April 1990