LINGUISTICALLY MOTIVATED DESCRIPTIVE TERM SELECTION

K. Sparck Jones and J.I. Tait*
Computer Laboratory, University of Cambridge
Corn Exchange Street, Cambridge CB2 3QG, U.K.

ABSTRACT

A linguistically motivated approach to indexing, that is the provision of descriptive terms for texts of any kind, is presented and illustrated. The approach is designed to achieve good, i.e. accurate and flexible, indexing by identifying index term sources in the meaning representations built by a powerful general purpose analyser, and providing a range of text expressions constituting semantic and syntactic variants for each term concept. Indexing is seen as a legitimate form of shallow text processing, but one requiring serious semantically based language processing, particularly to obtain well-founded complex terms, which is the main objective of the project described. The type of indexing strategy described is further seen as having utility in a range of applications environments.

I INDEXING NEEDS

Indexing terms are required for a variety of purposes, in a variety of contexts. Much effort has gone into indexing, and more especially automatic indexing, for conventional document retrieval; but the extension of automation, e.g. in the area of office systems, implies a wider need for effective indexing, and preferably for effective automatic indexing. Providing index descriptions for access to documents is not necessarily, moreover, a poor substitute for fully understanding documents and incorporating their contents into knowledge bases. Indexing has its own proper function and hence utility, and can be successfully done without deep understanding of the texts being processed. Insofar as access to documents is by way of an explicit textual representation of a user's information need, i.e. a request, this has also to be indexed, and the retrieval problem is selecting relevant documents when matching request and document term descriptions.

Though retrieval experiments hitherto have shown that better indexing (on some criterion of descriptive quality) does not lead to really large improvements in average retrieval performance, careful and sophisticated indexing, especially of the search request, does promote effective retrieval. Sophisticated indexing here means conceptually discriminating, linguistically motivated indexing, i.e. indexing in which terms are linguistically well motivated because they are accurate indicators of complex concepts. Though indexing concepts may in some cases be adequately expressed in single words, the concepts being indexed frequently have an internal structure requiring expression as a so-called 'precoordinate' term, i.e. a linguistically well-defined multi-word unit.

Earlier attempts to obtain such precoordinate terms automatically were not particularly successful, mainly because the text analysis procedures used were primarily syntactic, and even shallowly and crudely syntactic. Further, adopting source text units as terms, when they are only mininmally characterised, limits indexing to one particular expression of the underlying concept, and does not allow for alternatives: requests and documents may therefore not match. (Stemming helps somewhat but, for example, does not change word order.)

The research reported below was thus designed to test a more radical approach to indexing, using an AI-type language analyser exploiting a powerful syntactico-semantic apparatus to analyse texts, and specifically request texts; a term extractor to identify indexing concepts in the resulting text meaning representation and construct their semantic variants; and a language generator to produce a range of alternative syntactic expressions for all the forms of each concept, constituting the terms variant sets for searching the document file. The major operation is the identification of indexing concepts, or term sources, in text meaning representations. If both user requests and stored documents could be processed, there would be no need for lexical expressions of these concepts, since matching would be conducted at the representational level (cf Hobbs et al 1982 or, earlier, Syntol (Bely et al 1970)). However there are many reasons, stemming both from the current state of automatic natural language processing and from naked economics, why full document processing is not feasible, though request processing should be. The generation of alternative text expressions of concepts, for use in searching stored texts, is therefore necessary. We indeed believe that text searching is an important facility for many practical purposes. The provision of indexing descriptions is thus a direct operation only on requests, but the provision of alternative well-founded expressions of request concepts constitutes an indirect indexing of documents aimed at improving request document matching.

There would nevertheless appear to be a major problem with this type of application of AI language

*Current address: Acorn Computers Ltd, Fulbourn Road, Cherry Hinton, Cambridge CB1 4JN, U.K.

This work was supported by the British Library Research and Development Department.

analysers. In general, successful 'deep' language analysis programs have been those working within very limited domains; and the world of ordinary document collections, for example those consisting of tens or hundreds of thousands of scientific papers, is not so limited. Programs like FRUMP (DeJong 1979), on the other hand, though less domain specialised, achieve only partial text analysis. They in any case, like 'deep' analysers, imply an effort in providing an analysis system which can hardly be envisaged for language processing related to large bodies of heterogenous text.

The challenge for the project was therefore whether sophisticated language analysis techniques could be applied in a sufficiently discriminating way, without backup from a large non-linguistic knowledge base, given that only a partial interpretation of texts is required. The partial interpretation must nevertheless be sufficient to generate good, i.e. accurate and significant, index terms; and the important point is therefore that the partial interpretation process has to be a flexible one, driven bottom up from the given text rather than top down by scripts or frames. Thus the crucial issue was whether the desired result could be obtained through a powerful and rich enough general, i.e. non domain-specific, semantics.

II REQUEST ANALYSIS

To test the proposition that the desired result could be obtained, we exploited Boguraev's analyser (Boguraev and Sparck Jones, in press), which applies primitive-based semantic pattern matching in conjunction with conventional syntactic analysis, to obtain a request meaning representation in the form of a case labelled dependency tree relating word senses characterised by primitive formulae. Thus a primary objective was to see whether the type of word and message meaning characterisation allowed by the general semantic primitives used by the analyser could suffice for the interpretation of technical text for the purpose in hand. There is an early limit to the refinement of lexical characterisation which can be achieved with about 100 general-purpose primitives like THING and WHERE for a vocabulary containing words like "transistor", "oscillator" and "circuit"; and with semantic lexical entries for individual word senses at the level of 'oscillator: THING', structural disambiguation of the sentence as a whole may be difficult to attain. In this situation, the analyser is unlikely to be able to achieve comprehensive ambiguity resolution; but the project belief was that lower-level sentence components could be fairly unequivocally identified, which may be adequate for indexing, since it is not clear how far comprehensive higher-level structural links should be reflected in terms. A modest level of lexical resolution may also be sufficient as long as some trace of the input word is preserved to use for output variant generation (which may of course include synonym generation).

The fact that the semantic apparatus supporting Boguraev's analyser is rich and robust enough to tolerate some 'degradation' or 'relaxation' was one reason for using this analyser. The second was the nature of the meaning representations it delivers. The output case-labelled dependency tree provides a clear, semantically characterised representation of the essential propositional structure of the input text. This should in principle facilitate the identification of tree components as term sources, according to more or less comprehensive scope criteria, as suggested by the needs of request-document matching.

The third reason for adopting Boguraev's analyser was the fact that it has been used for a concurrent project on a query interpretation front end for accessing formatted databases, and hence was viewed as an analyser capable of supporting an integrated information inquiry system. The principle underlying the projects taken together was that it should be recognised that information systems consist of a range of different types of information source, which it should be possible to reach from a single input user question. That is, the user should be able to express an information need, and the system should be able to couch this in the different forms appropriate to seeking response items of different sorts from the range of available information source types. Thus a question could be treated both as a query addressed to a formatted database, and as a request addressed to a document collection, without presuppositions as to what type of information should be sought, in order to maximise the chances of finding something germane. In other projects, e.g. LUNAR (Woods et al 1972), treating questions as document requests was either triggered by specific words like "papers", or by a failure to process the question as a database query. We regard the treatment of the user's question in various styles at once as a normal requirement of a true integrated information system.

In the event, Boguraev's analyser had to be extended significantly for the document retrieval project, primarily to handle compound nouns. These are a very common feature of technical prose, so some means of processing them during analysis, and some way of representing them in the analyser's output, is required, even if they cannot be fully interpreted without, for example, inference calling on pragmatic (domain) knowledge. The necessarily somewhat minimal procedure adopted was to represent compounds as a string of modifiers plus head noun without requiring an explicit bracketing or reconstruction of implicit semantic relations. (Sense selection on modifiers thus cannot in general be expected.) In general, such a strategy implies that little term variation can be achieved; however, as detailed below, some follows from limited semantic inference.

The type of meaning representation provided by the analyser for a typical request is illustrated (in a simplified form) in Figure 1a.

III TERM EXTRACTION

From the indexing point of view, the most important operation is the selection of elements of the analyser's output meaning representation(s) as term sources. Subject to the way the representation defines well-formed units, the criteria for term source selection must stem ultimately from the empirical requirements mainly of request-document matching, but also, since index descriptions can have other functions than pure matching, from the requirements for descriptions which are, for example, comprehensible and indicative to the quickly scanning human reader. The particular requirements to be met

```
Request:
GIVE ME INFORMATION ABOUT THE PRACTICAL CIRCUIT
DETAILS OF HIGH FREQUENCY OSCILLATORS USING
TRANSISTORS
a) 18 analyses including (simplified illustration):
(clause...
   (v...
      (@@agent...)(@@recipient...)
      (@@object...(@@mental object ...
 <n(detail1 sign
    (@@atttribute (trace (clause v agent)
     (clause
     (v (use1 use
         (@@agent (n (oscillator1 thing
           (##nmod (trace (clause v agent)
             (clause
             (v (be2 be
               (@@agent (n (frequency1 sign)))
               (@@state ...high3 kind) )) )) ) )) )
        (@@object (n (transistor1 thing)) ) )) )) )
    (##nmod (trace (clause v agent)
     (clause
      (v (be2 be (@@agent (n (circuit1 thing)))
               (@@state ...practical2 kind) )) )) ) ) >
)))
b) 10 term sources of scale 2 for this analysis
   including:

   (n (detail1 sign
     (##nmod (((n (circuit1 thing)))) )))
   ((trace (clause v agent))
     (clause (type rel)
       (v (use1 use
         (@@agent (n (oscillator1 thing)))) )))
   ((trace (clause v object))
     (clause (type rel)
       (v (use1 use
         (@@object (n (transistor1 thing)))) )))
c) semantic variants using inference for compound
   nouns, selecting prepositional cases from
   17 possible:

   for 'circuit detail' in this analysis
   3 new variants:

   (n (detail1 sign
     (@@abstract location (n (circuit1 thing)))) )
   (n(detail1 sign
     (@@mental object (n (circuit1 thing)))) )
   (n(detail1 sign
     (@@attribute (n (circuit1 thing)))) )
d) 15 term search specification for the request
   using terms of scale 2, with compound noun
   inference:

   variant set of 5 for 'frequency oscillator'
   including:

       "a frequency oscillator"
       "frequency oscillators"

   variant set of 25 for 'circuit detail' interpreted
   as 'detail about circuit' including:

       "the details about the circuits"
       "detail about circuits"
       "details about a circuit"
```

Figure 1. Example request processing

can only be determined by extensive and onerous experiment. However some of the possibilities open can be indicated here, since specific decisions had to be made for the first, very small scale, tests we have already conducted.

Roughly speaking, the definition of term sources is a matter of scale, i.e. of the larger or smaller scope of dependency tree connections. At the surface text level this is reflected in (on average) larger or smaller word strings, corresponding to more or less elaborately modified concepts, or more or less extensively linked concepts. Given the type of propositional structure defined by the analyser's dependency trees, it was natural to define term sources by a scale count exploiting case constructions. In the simplest case the scale count is effectively applied to a verb and its case role filler nouns. Thus a count of 3 takes a verb and any pair of its role-filling nouns, a count of 2 takes the verb and any one of its nouns, while a count of 1 takes just verb or noun. A structure with a verb and three noun case fillers will therefore produce three scale 3 terms, three scale 2, and 4 scale 1 sources. Figure 1b shows sources of scale 2 extracted from the dependency structure representing the concept 'oscillator use transistor' for the example request.

It should be emphasised that some types of linguistic construction, e.g. states, are represented in a verb-based way, and that other dependency tree structures are handled in an analogous manner. Equally, the definition of scale count is in fact more complicated, to take account of modifiers on nouns like quantifiers. Moreover an important part of the term source selection process is the elimination of 'unhelpful' parts of the sentence representation, for example those derived from the input text string "Give me papers on". This elimination is achieved by 'stop structures' tied to individual word senses, and can be quite discriminating, e.g. distinguishing significant from non-significant uses of "paper". Term sources are then derived from the resulting 'partial' sentence structures. (In Figure 1a this is the structure bounded by < >.)

Overall, the effect of the term source derivation procedure is a list of source structures, representing propositions or subpropositions, which overlap through the presence of common individual conceptual elements, namely word senses. It is indeed important that the indexing of a text is 'redundant' in this way.

If this conceptual indexing were to be carried out on both requests and stored documents, such lists would be the base for searching and matching. The fragmentation characteristic of indexing suggests that considerable mileage could be got simply from the lists of extracted term sources, without extensive 'inferential' processing either to generate additional sources or to support complex matching in the style advocated by Hobbs et al. However the objectives of indexing are unlikely to be achieved by restricting indexing concepts to the precise detailed forms they have in the analyser's meaning representation. In general one is interested in the essential concept, rather than in its fine detail; for instance, in most cases it is immaterial whether singular or plural, definite or indefinite, apply to nominals. Indexing only at the conceptual level would simply throw such information away, to emerge with a 'reduced' or 'normalised' version of the concept, though one which conveys more specific structural information than the 'association' or 'coordination' ordinarily used in indexing. However if searching is to be at the text level, proper bases for the text expressions involved must be retained. Moreover 'paring down' representations may lead to the lack of precision in term characterisation which it is the aim of the whole enterprise to avoid, so an alternative strategy, allowing for more control, is required. The one we adopted was to define a set of permitted semantic variations, for example deriving plural and/or indefinite nominals from a given single definite construction.

Such semantic variants are easily obtained. Compound nouns present more interesting problems, and we have adopted a semantic variant strategy for these which may be described as embodying a very crude form of linguistic inference. Variants on given compounds are created by applying, in reverse, the semantic patterns designed to interpret and attach prepositional phrases in text input. That is, if the semantic formulae for a pair of nouns in a compound satisfy the requirements for linking these with some (sense of a) preposition, the preposition sense, which embodies a case relationship, is supplied explicitly. Figure 1c shows some inferred variants for the example request. Clearly this technique (to be described in detail in the full paper) could be extended to the linking of nouns in a compound by verbs.

But further, indexing strategies involve more than choices of term source and semantic variant types. Indexing implies coverage of text content, and it may in practice be the case that text content is not fully covered if indexing is confined to terms of a certain type, and specifically those of a more exigent, higher scale. Thus an exclusive indexing strategy may be restricted in coverage, where a relaxed one accepts terms of lower scale if ones of the preferred higher scale are not available, and so increases coverage. Moreover it may be desirable, to increase matching chances, to index with an inclusive strategy, with subcomponent terms of lower scale as well as their parents of higher scale, treating subcomponents as variants. The relative merits of these alternatives can only be established by experiment.

IV VARIANT EXPRESSION

More importantly, indexing cannot in practice stop at the level of term sources and their semantic variants, i.e. operate with the components of text meaning representations. The volumes of material to be scanned imply searching for request-document matches at the textual rather than the underlying conceptual level. This is not only a matter of the limited capacity for full text (or even abstract) processing of current language processing systems. It can be argued that text level scanning without proper meaning interpretation is a valid activity in its own right, for example as a precursor to deeper processing.

The final stage of request processing is therefore the generation of text equivalents for the given term sources (i.e. for all the variants of each source). This includes the generation of syntactic variants, exploiting further the power given by explicit descriptions of linguistic constructs: though relations between words are implicit in word strings pulled out of texts, they cannot be accessed to produce alternative forms. What constitutes a syntactic as opposed to a semantic variant is ultimately arbitrary; in the implemented generator it includes, for example, variations on aspect. This generator, a replacement of Boguraev's original, builds a surface syntactic tree from a meaning representation fragment, from which the output word string is derived. The process includes the listing (if these are available) of lexical variants, i.e. words which are sense synonymous with the input ones. The final step in the production of the search formulation for the input request is the packaging of the sets of variants derived from the request's constituent concepts into a Boolean expression, with the variants in the set for each source linked by 'or' and the sets, representing terms, linked by 'and'. This stage includes merging the results of alternative analyses of the input request. Figure 1d illustrates some of the text expressions of semantic and syntactic variants for the example request.

From the retrieval point of view, our tests have been very limited. As noted, text searching is extremely costly, and requires a highly optimised program. Our initial experiment was therefore in the nature of a feasibility study, aimed at showing that real requests could be processed, and the output query specifications searched against real abstract texts. We matched 10 requests against 11429 abstracts, in the area of electronics, using terms of scales 3, 2, and 1, and also 2 with compound noun inference, and the exclusive strategy. The strategies performed identically, but it has to be said that otherwise the results, especially for the higher scales, were not impressive. However, as retrieval testing over the past twenty years has demonstrated, the request sample is too small to support any valid performance conclusions about the merits of the indexing methods studied: a much larger sample is needed. Moreover much more work is needed on the best ways of forming search specifications from the mass of term material available: this is currently fairly ad hoc.

V CONCLUSION

The work described represents a first study of the systematic use of a powerful language processing tool for indexing purposes. It could in principle be used to manipulate terms at the meaning representation level, which would have the advantage of permitting more flexible matches between requests and documents differing at the detailed text level (e.g. "retrieval of information" and "retrieval of relevant information"). More practically, the indexing is extended to provide alternative text expressions of indexing concepts, for text matching. The claim for the approach is that useful indexing can be achieved by general semantic rather than domain-specific knowledge, though much more testing, includng tests with different indexing applications, is needed.

VI ACKNOWLEDGEMENT

We are grateful to Dr. B. K. Boguraev for his advice and assistance throughout the project.

VII REFERENCES

Bely, N. et al, Procedures d'Analyse Semantiques Appliquees a la Documentation Scientifique, Paris: Gauthier-Villars, 1970.

Boguraev, B. and Sparck Jones, K. 'A natural language front end to databases with evaluative feedback' in New Applications of Databases (ed Gardarin and Gelenbe), London: Academic Press (in press).

DeJong, G. Skimming Stories in Real Time, Report 158, Department of Computer Science, Yale University, 1979.

Hobbs, J.R. et al, 'Natural language access to structured texts' in COLING 82 (ed Horecky), Amsterdam: North-Holland, 1982.

Woods, W.A. et al, The LUNAR Sciences Natural Language Information System, Report 2378, Bolt Beranek and Newman Inc., Cambridge MA, 1972.