Beyond keywords: accurate retrieval from full text documents

Sutcliffe, R.F.E.; Boersma, P.; Bon, A.; Vossen, P.J.T.M.

1995

document version
Publisher's PDF, also known as Version of record

citation for published version (APA)
Sutcliffe, R. F. E., Boersma, P., Bon, A., & Vossen, P. J. T. M. (1995). *Beyond keywords: accurate retrieval from full text documents*. Vrije Universiteit.

Beyond Keywords: Accurate Retrieval from Full Text Documents

Richard F. E. Sutcliffe[*][1], Paul Boersma†, Annelies Bon†, Ton Donker†,
Michael C. Ferris‡, Peter Hellwig§, Patrick Hyland‡, Heinz-Detlev Koch§,
Pieter Masereeuw†, Annette McElligott*, Donie O'Sullivan*, Liam Relihan*,
Iskandar Serail†, Ingrid Schmidt§, Liam Sheahan*, Bronwyn Slater*,
Henriette Visser§, Piek Vossen†

Lotus Development Ireland‡,   University of Amsterdam†
University of Heidelberg§,   University of Limerick*

SIFT (Selecting Information From Text) is a prototype text retrieval system which is intended to find portions of a technical manual whose meanings are related to that of an input query. This is accomplished by using a semantic representation scheme based on ontological distance. Our initial findings are that the distance measure and utterance matching heuristics are quite robust provided that the problem of sense disambiguation can be circumvented. The first version of the system which is based on 735 titles from the Ami Pro User's Guide is currently being evaluated.

1   Introduction

In the commercial world of the future there will be an ever increasing reliance on full text documents in electronic form. However, as the amount of available material increases the task of finding information becomes more and more difficult.

In the domain of technical documentation a common difficulty is in pinpointing the location of vital facts relating to a specific task. Traditionally the answer lies in a search for one or more *keywords* extracted from an input query. While such methods can be very sophisticated they suffer from a fundamental weakness: a relevant passage may contain a word similar in meaning to the keyword but spelled differently. For example, a search for 'trade' you will not find a passage which contains 'commerce'.

We are investigating an approach to text retrieval which aims to transcend the limitations of keywords. Underlying it is a technique which allows the meanings of two words to be compared directly. Thus in our approach 'trade' will match 'commerce' but not as strongly as 'trade' matches itself.

A prototype text retrieval system called SIFT has been built which tests the ideas in the domain of software instruction manuals. In this article we first describe and justify

---

the chosen task domain. Next, the architecture of SIFT is outlined. This leads on to a discussion of the meaning representations which underlie the system. Results of the work so far are then presented, followed by conclusions.

2 The Task Domain

Our chosen domain is the software instruction manuals which are supplied with PC software. These have a number of interesting characteristics. First, they are relatively short, compared to the large bodies of text typically considered within the paradigm of Information Retrieval (IR). Second, manuals are structured in units such as sections, subsections, side notes and so on. Third, the focus of the text is very narrow. Fourth, the terminology used is very stylised. Fifth, manuals are specifically designed to facilitate the retrieval of information. Sixth, the answer to any reasonable query is supposed to be included within the manual. At present we are focusing on the *Lotus Ami Pro Word Processor for Windows User's Guide Release 3* (Ami Pro, 1993).

Increasingly, instruction manuals are being supplied in on-line form so that they can be queried while the software which they describe is being used. However, simple keyword searches are usually all that are available to help the user to find information. We therefore believe that there is a market for more sophisticated text retrieval software provided that it can give better access to textual data and at the same time operate with acceptable speed.

3 Architecture of SIFT

The SIFT system is a text retrieval prototype which consists of two principal components (Figure 1). The *document processing component* takes as input an SGML tagged computer manual and associates with the different sections, subsections and individual sentences of that manual distributed patterns capturing the meaning of those textual units. (We use the term *utterance* to denote a unit of text falling into one of these categories.) The *query processing component* takes as input a user query about the material covered in the document and produces as output a list of pointers to text portions within it which are ordered by relevance to the query.

Document processing operates by first selecting utterances from the text which are to be used as the basis for retrieval. Each such utterance is converted into a representation which captures its meaning. This involves analysis of terminology to recognise compounds such as 'Ami Pro', syntactic category tagging using the Brill (1992) method, determining the semantic sense of each content word and the construction of a suitable representation for the utterance as a whole. This representation is then linked back to the utterance within the original text from which it was derived. The result of this process is an indexed version of the original document.

The Query processing component uses the indexed document to provide pointers into the text which are relevant to a particular input. The query utterance is subjected to the same analysis as a text utterance, namely compound recognition, tagging and sense determination so that a semantic representation can be constructed for it. A search is then performed in which this representation is compared with those corresponding to
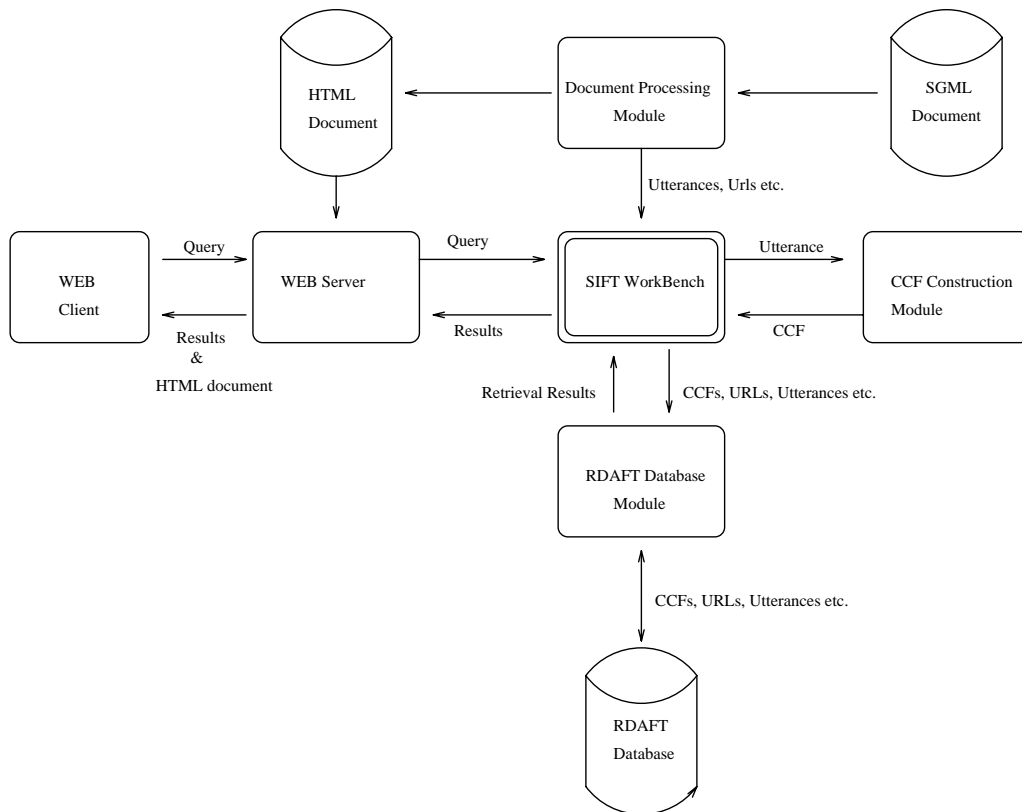
Figure 1: Architecture of the SIFT System

utterances within the indexed text. The comparison is easy because vectors are used to capture word meanings. The result of the search is a list of pointers to text portions ordered by their relevance to the query. An environment is provided in which the user can select a text portion of interest from such a list and display it on the screen.

4 Meaning Representation and Comparison

In common with many other researchers (e.g. Richardson, 1994), we use a concept ontology as the basis of our meaning representation scheme (see Figure 2). This is a tree which relates word senses by the relations IS-A and IS-PART-OF. For example 'floppy disk' IS-A 'disk' and 'file' IS-PART-OF 'directory'. The similarity in meaning between concepts can be equated directly to the distance between them in the ontology: adjacent words are very similar while distant ones are only peripherally related. Various schemes have been proposed for measuring path length. Our approach is to generate a normalised n-dimensional vector for each word sense by traversing the ontology extracting semantic features from the textual definitions of all concepts above the word being defined (Sutcliffe, O'Sullivan and Meharg, 1994). One word sense can then be compared with another by computing the dot product of their vectors. If the result is one, the words are synonymous. If it is between zero and one, the words are related. Finally, if it is zero, the words are not related.
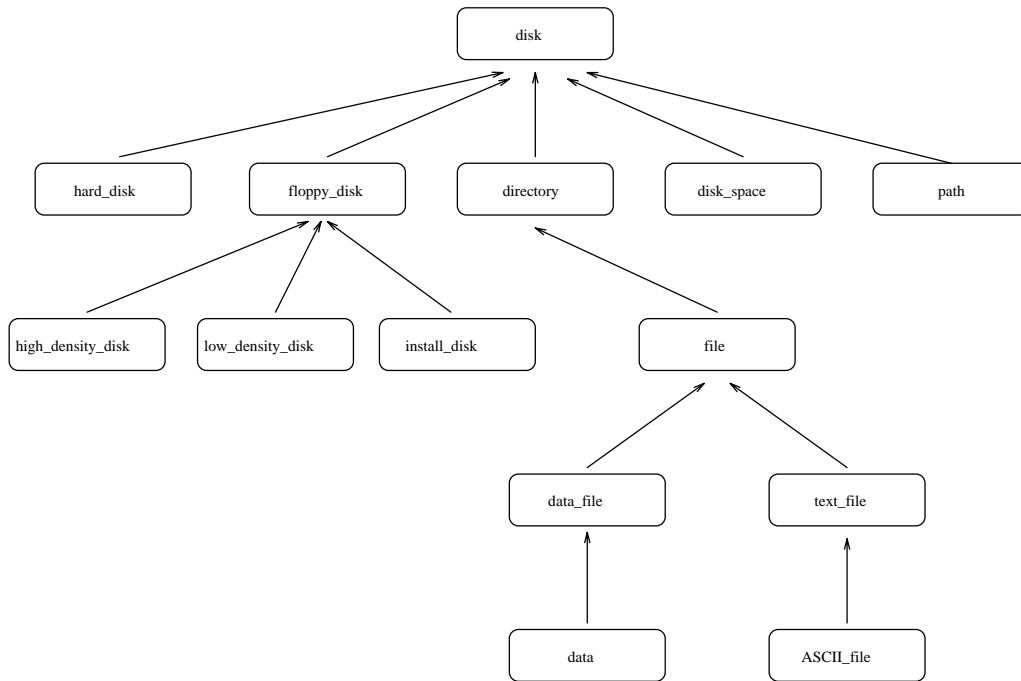
Figure 2: An Example Concept Ontology

The ontology is in two parts. General senses of words are covered by the Princeton WordNet Version 1.4 (Beckwith, Fellbaum, Gross and Miller, 1992) while domain specific senses together with the multiple word terminology of word processing are held in an additional taxonomy which we have developed (O'Sullivan, McElligott and Sutcliffe, 1995).

Each utterance in the text is represented as a set of <syntactic category,vector> pairs. The first element of each pair is either 'noun' or 'verb', other categories of word being discarded. The second element is a word sense representation derived from the ontology as just outlined.

During retrieval, a match must be made between the representation of the query utterance with that of a document utterance. A heuristic method for accomplishing this has been developed as follows. Each query utterance content word representation of type noun is compared with the representations of all nouns in the document utterance and the best match is selected. Similarly, the meaning of each verb in the query is compared with all verb representations in the text utterance and the best match selected. Finally, the sum of all such matches is computed. Because each word–to–word match ranges between zero and one, a query–utterance–to–document–utterance match ranges between zero and n, where n is the number of content words (nouns or verbs) in the query.

5   Progress and Findings

A version of SIFT has been built which uses headings and subheadings to index the text. There are 735 of these in the Ami Pro manual. At present full evaluation of SIFT has not yet taken place. However, an idea of the operation of the system can be obtained

from Figure 3 which shows the output for the query 'rotate object'. This illustrates the effect of the semantic matching algorithm which underlies the whole project. For example 'rotate' not only matches itself with strength 1.0, but also matches 'flip', and 'move' with strengths 0.59 and 0.51 respectively. This is why utterance 02 ('FLIP an OBJECT') matches the query more strongly than utterance 03 ('MOVE an OBJECT'), but not as strongly as utterance 01 ('ROTATE an OBJECT'). Similar effects can be discerned for nouns. For example 'object' matches 'frame', 'template' and 'power field' with strengths 0.86, 0.74, and 0.63. This is why utterance 04 ('to MOVE the insertion point inside a FRAME') comes before 09 ('to MOVE the insertion point between input boxes in a TEMPLATE') which in turn precedes 13 ('to MOVE or copy a POWER FIELD').

The findings of this project so far can be summarised as follows. Firstly, the semantic distance measure works well for nouns. Ontological relationships between nouns are clearly specified in WordNet and it was relatively straightforward to augment this taxonomy in order to handle domain specific vocabulary. Secondly, the distance measure works reasonably well for verbs. However, verb meanings can not be fully captured by ontological relationships alone. For example the characteristics of a verb's arguments (subject, object, prepositional phrases, etc.) must be considered also. Third, the distance measure which we have developed appears to be usable as a method for matching the meaning of a simple query to that of a text utterance. It is robust and only requires syntactic category tagging of the text. Whether this approach is viable as a method of text retrieval remains to be shown.

A crucial aspect of conceptual matching is the ability to determine the correct sense of each word in an utterance relative to a particular lexical database. For example in Figure 3 the term 'object' has a sense (roughly 'textual object') which is specific to the domain. This is quite different from two other common senses, namely 'goal of an endeavour' and 'something which exists'. If incorrect senses are inadvertently used by a retrieval program then any advantage to be gained from the use of semantic processing is soon lost. We have experimented with various automatic disambiguation algorithms but none of these is reliable enough for SIFT. On the other hand, multiple word terms such as 'paragraph style' are not ambiguous and can be recognised reliably. In addition the vocabulary of the manuals is very restricted with the same words and phrases being used repeatedly. These points together suggest that the lack of a reliable automatic disambiguation algorithm is not the major limitation of conceptual retrieval, provided that the domain is sufficiently controlled.

```
01   To ROTATE an OBJECT (2.000000)
02   To FLIP an OBJECT (1.591837)
03   To MOVE an OBJECT (1.511101)
04   To MOVE the insertion point outside a FRAME (1.374213)
05   To MOVE the insertion point inside a FRAME (1.374213)
06   To MOVE or copy a FRAME to another page (1.374213)
07   To MOVE a FRAME on the same page (1.374213)
08   To copy or MOVE an EQUATION or part of an equation (1.248765)
09   To MOVE the insertion point between input boxes in a TEMPLATE (1.248765)
10   To MOVE the insertion point into a TEMPLATE (1.248765)
11   To MOVE TEXT (1.183059)
12   To use Drag & Drop to MOVE and copy TEXT (1.183059)
13   To MOVE or copy a POWER FIELD (1.141547)
14   To use Drag & Drop to MOVE or copy a POWER FIELD (1.141547)
15   To MOVE or copy text between DOCUMENTS (1.126179)
16   To MOVE a PARAGRAPH STYLE (1.037062)
17   To edit an OLE OBJECT (1.000000)
18   Editing an OLE OBJECT (1.000000)
19   To embed existing data as an OLE OBJECT (1.000000)
20   To embed new data as an OLE OBJECT (1.000000)
21   Embedding an OBJECT (1.000000)
22   Copying a drawing or an OBJECT (1.000000)
23   To save a drawing or an OBJECT as a graphic file (1.000000)
24   Saving a drawing or an OBJECT as a graphic file (1.000000)
25   To apply the current line style and fill pattern to an OBJECT (1.000000)
26   To extract the line style & fill pattern of an OBJECT (1.000000)
27   Modifying an OBJECT (1.000000)
28   To delete an OBJECT (1.000000)
29   To modify the shape of an OBJECT (1.000000)
30   To size an OBJECT (1.000000)
31   To copy an OBJECT (1.000000)
32   To deselect an OBJECT (1.000000)
33   To select an OBJECT (1.000000)
34   To create an OBJECT (1.000000)
35   To MOVE SELECTED PARAGRAPHS (0.986588)
36   To align OBJECTS to the grid (0.906977)
37   Using layered OBJECTS (0.906977)
38   Grouping OBJECTS (0.906977)
39   To select all OBJECTS (0.906977)
40   To select multiple adjacent OBJECTS (0.906977)
41   To select multiple OBJECTS (0.906977)
42   Selecting OBJECTS in a drawing (0.906977)
43   To modify the shape of a POLYLINE or polygon (0.867524)
44   moving a picture in a FRAME (0.863112)
45   Examples of creating PICTURE FRAMES (0.863112)
46   To delete a picture inside a FRAME (0.863112)
47   Using a picture in a FRAME (0.863112)
48   To modify the size or position of a FRAME (0.863112)
49   To modify the type of FRAME (0.863112)
50   Using text in a FRAME (0.863112)
```

Figure 3: Output from SIFT-1 for the query 'rotate object'. Only headings in the manual are being searched. This example illustrates how the similarity measure allows the query to match utterances which are semantically related even though different vocabulary is used. The terms in each utterance which are causing the match are capitalised. The reason for the match by query number is as follows: 01: perfect match of verb and noun. 02-03: partial match of verb, perfect match of noun. 04-16: partial match of both verb and noun. 17-34: no match of verb, perfect match of noun. 35: partial match of verb and noun. 36-50: no match of verb, partial match of noun.

6   Conclusion

We have outlined the SIFT project which aims to investigate the efficacy of using a conceptual distance measure as the basis for text retrieval on software instruction manuals. Provided that the problem of disambiguation can be circumvented, the essential meaning of two simple utterances can be compared by our methods, thus avoiding the limitations of keyword-based searches. The approach thus offers the potential of high performance text retrieval in certain restricted domains as well as being applicable to related language engineering tasks such as machine-assisted translation. We are currently engaged in comparing the performance of SIFT with that of the well-known tf*idf algorithm (Salton, 1989) which is an optimised form of keyword search.

Two other steps currently being undertaken are as follows. First, we are extending the indexing process to utterances from the text of the manual to investigate their effect on retrieval performance. Second, we are extracting more detailed predicate-argument information from the manual by syntactic parsing so that the utterance matching algorithm can take this into account.

References

Ami Pro (1993). *Lotus Ami Pro Word Processor for Windows User's Guide Release 3.* Atlanta, GA: Lotus Development Corporation, Word Processing Division.

Beckwith, R., Fellbaum, C., Gross, D., & Miller, G. A. (1992). WordNet: A Lexical Database Organised on Psycholinguistic Principles. In U. Zernik (Ed.) *Using On–line Resources to Build a Lexicon.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Brill, E. (1992). A simple rule-based part of speech tagger. *Proceedings of the Third Conference on Applied Natural Language Processing, ACL, Trento, Italy, 1992.*

O'Sullivan, D., McElligott, A. & Sutcliffe, R. F. E. (1995). Augmenting the Princeton Word-Net with a Domain Specific Ontology. *Proceedings of the IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing, 19-21 August, 1995, Montreal, Canada.* Also available as a Technical Note, Department of Computer Science and Information Systems, University of Limerick, 15 April, 1995.

Richardson, R. (1994). A Semantic-based Approach to Information Processing. Doctoral Dissertation, School of Computer Applications, Dublin City University.

Salton, G. (1989). *Automatic Text Processing.* Reading, MA: Addison-Wesley.

Sutcliffe, R. F. E., O'Sullivan, D., & Meharg, F. (1994). A Lexicon of Distributed Noun Representations Constructed by Taxonomic Traversal. Proceedings of the 15th International Conference on Computational Linguistics, (COLING'94), Kyoto, Japan, 827-831.