Selecting Information From Text
Final Report

LRE-62030 Deliverable D61

Patrick Hyland,
Heinz-Detlev Koch
Richard F. E. Sutcliffe
Piek Vossen

Contact point:
Richard F. E. Sutcliffe, Department of Computer Science
University of Limerick, Limerick, Ireland
Tel: +353 61 202706, Fax: +353 61 330876
Email sutcliffer@ul.ie

Status: Draft
Date: August 1996
Version: 1.0
Status: Public

sponsored by

European Community Linguistic Research and
Engineering Programme, European Commission, DG XIII

1. Introduction

1.1 Objectives

There were three main objectives in SIFT:

- To investigate the potential for using concept based retrieval techniques in the domain of technical documents,
- To develop the technology for exploiting concept based methods,
- To illustrate the use of this technology in a series of prototypes operating via the World Wide Web (WWW).

1.2 Work Undertaken

The work of SIFT can be summarised as follows:

- The domain of technical instruction manuals was investigated,
- Tools and resources were developed or adapted for the project,
- Systems were built and evaluated.

1.3 Results

The main results are:

- Three **text retrieval engines** operating on the Lotus Ami Pro User's Guide (Ami Pro, 1993),
- **Expertise** in the construction of retrieval systems combining keyword and conceptual retrieval strategies,
- A range of **linguistic resources** including a concept ontology based on the Longman Dictionary of Contemporary English (LDOCE) (Proctor, 1978), and a domain specific ontology of over 4,000 computer terms linked to both LDOCE and the Princeton WordNet (Beckwith, Fellbaum, Gross and Miller, 1992)
- Many **tools,** among them a complete lexical database system, two parsers, interfaces to the Brill Tagger (Brill, 1992), Wide Coverage Morphological Analyser (Karp, Schabes, Zaidel and Egedi, 1992) and Link Parser (Sleator and Temperley, 1991), two terminology recognisers and a regular expression matcher,
- A **Conference** on the parsing of software manuals together with a **book** summarising the results (Sutcliffe, Koch and McElligott, 1996),
- Many **publications** and **conference presentations.**

1.4 Findings

- Sources of error need to be controlled in order to exploit concept-based methods.
- More efficient concept-based retrieval mechanisms must be developed.

2. Technical Basis

2.1 VSM Retrieval vs. Concept Retrieval

Information Retrieval (IR) is the study and analysis of methods for retrieving information from large document collections. In a typical scenario a user types in a query such as 'how do I install AmiPro under Windows?' and the system responds with a series of documents which cover this topic. The user then refers to the documents in order to answer their query. The dominant retrieval paradigm within IR today is the Vector Space Model (VSM) of Salton (Salton, 1971). VSM is based on the following assumptions:

- If a document contains a keyword which is mentioned in the user query, this suggests that the document might be relevant to the query,

- The more frequently the keyword occurs in a document, the more likely that document is to be relevant.

These assumptions are embodied in VSM by taking the frequency of a term in a document into account in the retrieval process. The Term Frequency - Inverted Document Frequency (tf*idf) weighting scheme (Salton, 1989) is generally agreed to be one of the best in terms of retrieval performance.

□

□Since VSM was developed in the 1970s there have been many refinements and optimisations. However, it remains fundamentally a keyword based method. Notwithstanding the use of synonym lists, a keyword in the query must actually occur in the document if there is to be any match. VSM does not involve any understanding either of the query or the document collection and so by definition its performance can not exceed a fixed limit.

□

□The purpose of the SIFT project was to investigate whether robust Natural Language Processing (NLP) techniques could be used to augment the performance of VSM in a particular domain: answering queries relating to Personal Computer (PC) software on the basis of the user manual. The main techniques used in SIFT were distributed semantic lexical representations and distributed semantic cases, allowing word and predicate meanings to be represented and compared, and robust parsing, allowing predicate-argument information to be extracted from both documents and queries. These are discussed in the next sections.

□

2.2 Semantic Representation

The idea of classifying objects in a hierarchical fashion is well known in many areas of science. Links between word senses have been widely studied within linguistics leading to the construction of concept ontologies such as WordNet (Beckwith, Fellbaum, Gross and Miller, 1992). This in turn has led to the idea that the similarity in meaning of a pair of concepts can be correlated with the distance between them in the ontology (Rada, Mili, Bicknell and Blettner, 1989; Resnik, 1994; Richardson, 1994; Sussna, 1993). Prior to SIFT we had experimented with a class of algorithms which traverse a concept ontology extracting semantic features at each node and associating with each a numerical strength based on the distance travelled from the start node (Sutcliffe, O Sullivan and Meharg, 1994). We call this process *taxonomic traversal.*

□

Taxonomic traversal can be understood by reference to Figure 1 which shows part of the WordNet ontology starting at the word 'software'. The following description outlines one version of the algorithm. In the first stage, lexemes in the gloss for 'software' itself are extracted and stemmed. Function words are removed. The remainder become semantic features with

```
software, software system -- (written programs or procedures or rules and associated documentation
                                              pertaining to the operation of a computer system)
    => product, production -- (an artifact that has been produced by someone or some process)
       => creation -- (something that has been brought into existence by someone)
          => artifact, article, artefact -- (a man-made object)
             => object, inanimate object, physical object -- (a nonliving entity)
                => entity -- (something having concrete existence; living or nonliving)
```

Figure 1: WordNet ontology for the word 'software'

strength 1. This leads to a list `[write/1, program/1, procedure/1...]`. Next we ascend to the next level of the onotology ('product', 'production') and extract terms from the associated gloss, giving each a strength 0.9. This leads to `[artifact/0.9, have/0.9, be/0.9, produce/0.9 ...]`. The procedure is repeated until the top of the ontology is reached or the strength associated with the level falls to zero. All terms extracted in this fashion are then collated. Finally, all strength values are scaled linearly so that the sum of their squares is one. The resulting list serves as a meaning representation for the concept 'software'.

□

□Two meanings can be compared by computing the dot product of their representations. The normalisation process ensures that the resulting value lies between one (an exact match, implying that the meanings of the two concepts are identical) and zero (no match, implying that the concepts have nothing in common).

□

□In SIFT, semantic lexica for use in the retrieval systems were created by performing taxonomic traversal on both the LDOCE ontology and WordNet Version 1.4.

□

2.3 Robust Parsing

Grammatical analysis within SIFT is based on the Dependency Unification Grammar (DUG) and the PLAIN parser, developed at the University of Heidelberg in the ESPRIT project *Translator's Workbench* (TWB).

The formal language used to represent knowledge in the PLAIN system is called Dependency Representation Language (DRL). The contents and structure of natural language input are represented by dependency trees which carry a syntagmatic role (e.g. Predicate, Subject, Object) and a lexeme with each node. In order to describe surface constructs in the language, DRL includes complex categories which can be associated with each node in the dependency tree and which consist of sets of grammatical feature types and feature values.

The DUG approach is extremely lexicalistic. Unlike common generative grammars, a DUG does not consist of a set of production rules by which a formal representation is derived, but rather describes the syntagmatic relations directly by so-called valency templates for the complements of individual lexical items. A complete grammatical description of a language consists of the following components:

- a morphosyntactic lexicon which maps elementary strings of the input language to basic DRL descriptions which, as a rule, comprise a lexeme and a complex morphosyntactic category; the morphosyntactic lexicon is divided into four parts:

  - a set of patterns which allow new vocabulary to be entered in the form of *paradigms* (e.g. call, calls, called, called, calling),

  - a description of the morphology in the form of a transition network; edges are labelled with inflectional elements, leading into sub-networks which provide the grammatical features which are appropriate for the word form,

  - the set of paradigms which allow the system to link individual words to the morphological network.

- a syntagmatic lexicon which describes the (recursive) combination capability of the elementary units and which consists of two data sets:

  - a set of valency templates which describe syntagmatic relationships,

  - a set of references which associates the vocabulary of the input language with the appropriate templates. The specification of a template for a lexical item is called its *valency frame*.

The fundamental operation of the parser is the matching of templates with actual representations or, metaphorically speaking, fitting fillers into slots in a bottom-up fashion. The PLAIN parser is extremely data-driven and so the words in the input determine the actions to be taken and the structures to be built. Thus if we wish to construct a partial parse working only with those words in a sentence which are known to be salient, it is only necessary to remove the non-salient material from the input before parsing begins.

2.4 System Architecture

A major objective in the project was to develop three text retrieval prototypes called SIFT-1, SIFT-2 and SIFT-3. Each prototype carries out two main processes: *document processing* and *query processing*. During document processing, utterances (sentences, headings, etc.) from an input document are analysed to determine their meaning and a distributed semantic representation for each is produced. This representation is then stored in a document database along with a pointer back to the place in the document from which it was derived. User queries can subsequently be accepted during query processing. Each query is analysed to determine its meaning, and a distributed representation is created for it. This representation is then matched with those previously stored in the database during document processing. Pointers to strongly matching utterances are collated and presented to the user who can then in turn view the corresponding sections of the document at the press of a button.

The display mechanism used for the project was the World Wide Web (WWW). This means that input is typed into a Hyper-Text Markup Language (HTML) form and that the resulting output is an HTML document. The SIFT system itself acts as a WWW server waiting for requests from browser programs and processing them in sequence. One ramification of this approach is that tools had to be developed for converting documents from word processor formats (such as Ami Pro) into HTML.

Figure 2 shows the architecture of the SIFT systems. There are two main components: the *Web Client* and the *SIFT Workbench.* The former is simply a browser program such as

Netscape. The latter is an experimental retrieval and indexing engine which links all other components of the system. We now outline the principal modules within SIFT.

The *Utterance Extraction Module* extracts utterances and their Uniform Resources Locators (URLs) from an input document. It can also convert portions of a document from SGML in the SIFTREP Data Type Definition (DTD) into HTML. The *CCF Construction Module* takes as

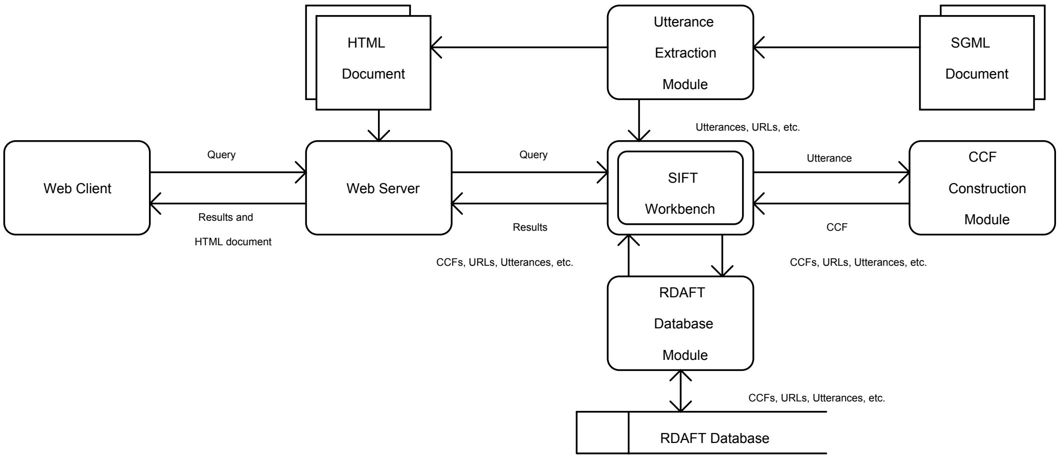

Figure 2: SIFT Architecture

□input an utterance, carries out an analysis of it and hence produces a semantic representation for it in the form of a Co-Ordinated Case Frame (CCF). The *RDAFT Database Module* can take a CCF and store it in a document database. In addition it can search a document database for CCFs whose meaning is similar to an input CCF and report back the matches which it finds. Finally, the *Web Server Module* allows the workbench to communicate with the WWW.

The three different SIFT systems have different capabilities. SIFT-1 tags an utterance for Part-of-Speech (POS) and recognises terminology but matches on these alone with no further grammatical information. SIFT-2 and SIFT-3 both use a parser to extract predicate-argument information and hence construct a semantic case frame for an utterance. In addition, SIFT-3 carries out a more sophisticated type of search based on knowledge of an input document's structure.

3. Research and Development Undertaken

3.1 Lexicon

3.1.1 Objectives

The main aims within the lexical side of the work were:

- To develop a paradigm of distributed semantic representations,
- To create these representations for the SIFT vocabulary,
- To build a lexical database system for storing them.

3.1.2 Work Undertaken

- A detailed study of vocabulary and terminology within the Ami Pro domain was carried out,
- An ontology derived from the Longman Dictionary of Contemporary English (LDOCE) (Proctor, 1978) was adapted and augmented with additional information,

- A separate ontology was constructed to capture relationships between domain dependent terms and this was linked to both LDOCE and the Princeton WordNet,
- The relationship of word sense disambiguation to the task domain was investigated. Several algorithms were implemented and tested,
- A comprehensive lexical database system complete with a sophisticated user interface was built and a variety of other specialised tools were developed.

A key aspect of the lexical work was the extraction of data from LDOCE using semi-automatic techniques. Parse trees for all definitions in the dictionary had been constructed as part of a previous project and these were used to construct the ontology underlying the dictionary. The main types of relation occurring in the taxonomy are hyponomy, meronomy and synonymy. However, each of these relations has been further divided to provide more fine-grained relations which can assist in the accurate assignment of feature weightings within the distributed lexical representations of the SIFT paradigm.

As is the case with most dictionaries, the taxonomy derived from LDOCE is a forest with 99 root nodes, not a tree. This causes problems for any method of semantic comparison, whether it is based on distributed representations or ontological distance, because the semantic relationships between those 99 nodes are effectively unspecified. Thus considerable effort had to be expended on the creation of links between the tree tops in order to create a single unified hierarchy. Separate top ontologies were constructed for verbs, concrete nouns and abstract nouns. The result was a single ontology of 19,049 nodes. This essentially covers all word meanings in the domain which match LDOCE senses, all sub-vocabulary or domain specific words which did not match LDOCE senses, and all intermediate words needed to link these to the hierarchy.

As well as building the ontology itself, work was carried out to facilitate the extraction of semantic representations from the database. One aspect of this was the specification of feature data for all word senses occurring in LDOCE. The features are triplets consisting of the semantic relation and two content words between which the words hold. Semantic case relations between events and entities expressed in the definitions as well as information on the expressed typicality of the data is also incorporated. On all, 135 semantic relations have been used with an average frequency of 70.47 occurrences of each relation in all the relevant senses.

As described earlier in this report, our initial experiments with taxonomic traversal algorithms were based on the assumption that any path between a pair of nodes in the ontology is the same length. Work on LDOCE included the development of more subtle semantic dependencies between words which could be used to inherit features and differentiate their weighting. Furthermore, an alternative way for determining the relevance of features to a concept was also established. This involves computing feature weighting on the basis of the hierarchical differentiation of concepts instead of the number of ontological links traversed. Using this mechanism it is possible to determine the relevance of semantic features which are derived from the general purpose lexicon on the basis of the differentiation of the domain. This means that the semantic representations can, in principle, be automatically customised for any domain.

Another area of interest to the project was to investigate the feasibility of extracting semantic case frames for verbs automatically from the dictionary. An inventory was created of all case relations expressed in definitions of nouns and verbs with meanings relevant to the domain. An approach was also devised for deriving case frames for verb meanings from these relations by combining case information from the definitions with the syntactic complementation specification of the verbs.

A crucial area of the project turned out to be the ability to disambiguate text. In order to extract the meaning from an utterance we need to know the semantic sense of the main verb and head nouns of any following noun phrases and prepositional phrases. The accuracy with which this information can be determined directly affects the ability of a conceptual retrieval engine to work effectively. Such systems are very easily degraded by noise. A comprehensive study of different methods was undertaken, including those of Lesk (1986), Ide and Véronis (1990) and Gale, Church and Yarowsky (1992). An algorithm combining dictionary and corpus techniques was subsequently implemented and tested within the Ami Pro domain, yielding an accuracy of 78%. The conclusion of this work however, was that highly accurate disambiguation relative to dictionary senses is not feasible at present in a practical system. For this reason, a compromise position had to be adopted whereby it was assumed that any word in the manual had the same semantic sense wherever it occurred and a subset of the total vocabulary was assigned a single sense deemed the most likely to be correct. This approach works well for the small class of verbs which occur in imperative constructs (for example within the numbered points of Level 2 sections headed by an infinitive verb phrase such as 'To display the parts of an Ami Pro window') and for multiple word terminology which occurs extremely frequently within the text. While the constituent words making up such compounds (e.g. 'dialog', 'box') are polysemous, the compound itself ('dialog box') has only one sense within the word processing domain.

3.1.3 Results

- A Cache Memory library in C for creating and accessing compact data files (especially lexicons) which can run on a Mac, PC or Unix system,
- A Lexical Database implemented in C and running on Unix and Macintosh machines,
- A terminology analysis module,
- A program called ELF (Extract Logical Form) for converting parse-trees of dictionary definitions into flat representations representing their underlying logical form.

The Lexical Database (LDB) is implemented in an object-oriented fashion and allows the user to carry out a number of functions, either from the graphical interface or via the library of C routines:

- Selection of word senses or entries which have certain properties,
- Traversal of semantic relations between word senses in an upward or downward direction, for a given relation type, sense or group of senses,
- The derivation of statistics on such matters as the number of relations per sense and the nature of the hierarchical structures,
- The ability to add, delete, replace, compare or merge lexical entries,
- The facility for the import and export of lexical data in various formats, including ASCII, binary and ltree.

In addition, the system includes both a general data editor for modifying information relating to a word sense, and a special data editor for disambiguating hierarchical relations in order to extend the semantic hierarchy. There is also a set of heuristics for deriving the hierarchical dependencies from the genus words extracted from definitions, together with a set of heuristics for deriving systematised semantic features from parsed definitions.

Lingware produced as part of the project includes:

- A description of the salient words and meanings relevant to the domain in terms of tokenised and lemmatised text, lists of names and name-like words, a list of words matching LDOCE entries, a specification of senses relevant to Ami Pro, word frequency data and concordance data,
- Morphosyntactic information for word meanings matching LDOCE senses including inflectional paradigms, countability of nouns and detailed complementation data for verbs,
- Hierarchical semantic structures for both nouns and verbs, as discussed earlier,
- Systematised semantic features for both nouns and verbs, as discussed earlier.

The complete LDB containing all the above information has a size of 5.7 Mbytes (binary data) and 6.0 Mbytes (ltree data). It is also available as a flat ASCII file of 108.6 Mbyte. The total lexicon contains 2,276 entries derived from LDOCE using 10,332 senses derived from LDOCE.

3.1.4 Findings

The main conclusions are as follows:

- It is possible to derive semantic and syntactic information for general words from an existing electronic dictionary (LDOCE), mainly using automatic techniques and up to a certain level of detail,
- Multiple word terminology can be extracted semi-automatically from a text. In addition heuristics can be used to infer ontological relations between terms. However, the creation of a domain-specific ontology is still largely a manual process,
- In a specific domain such as the Ami Pro manual, a large proportion (80%) of the text appears to consist of general words (about 2000 root forms) which are present in a learner's dictionary,
- General words nevertheless only represent 50% of the root forms which occur (in various inflections) in the manual. The remaining words (about 2000 root forms) are mostly names and single lexeme terms which are not very frequent in the text,
- Linking a domain-specific ontology to a general one is fairly straightforward,
- Present methods for disambiguating word senses are not sufficiently reliable for text retrieval applications. This means that any word in either a query utterance or a document utterance can only be used as part of the conceptual retrieval process if it has previously been linked by hand to a particular LDOCE or WordNet sense. Even this is only possible if the word is largely monosemous within the application domain.

The relationship between vocabulary and terminology in the domain is most interesting. In considering the results in relation to the high proportion of general vocabulary in the text, various points should be borne in mind. The vocabulary of verbs, especially those occurring in imperative constructions, is small, and their semantic senses are indeed fairly close to those found in dictionaries (e.g. 'move', 'select', 'create', 'print', 'type'). However, compound nominals occur very frequently in the text and often comprise common words such as 'file', 'menu' and 'bar' in different permutations and combinations (e.g. 'file menu', 'menu bar'). Such terms have meanings which are very specific to the domain and do not correspond directly to word senses in a general dictionary. It is important to recognise them because they are often the only nominals in an utterance whose semantic sense can be predicted accurately.

3.2 Parsing

3.2.1 Objectives

In SIFT-2 and SIFT-3 the intention was to index the text of the document in terms of a stream of distributed semantic case frames. In order to do this it was necessary to extract syntactic information from utterances. The aims here were therefore to investigate the parsing requirements of the project and then to develop a suitable parsing system to fulfil them.

3.2.2 Work Undertaken

Research in this area can be divided into several categories:

- A detailed study of the grammatical constructions occurring in the Ami Pro domain,
- The development of a syntagmatic grammar for PLAIN, based on the results of the study,
- Conversion of the PLAIN software from a prototype system written in Pascal and running on a mainframe to a development system operating under Unix,
- The investigation of other parsing systems including the Link Parser of Sleator and Temperley (Sleator and Temperley, 1991),
- The organisation of a workshop on the parsing of technical manuals,
- The implementation of a simple but fast predicate-argument extraction engine based on an input already tagged for part-of-speech.

In order to use the PLAIN system for the parsing of texts in English, the existing grammar for English had to be refined or replaced. As a basis for the necessary enhancements, the texts to be analysed had to be examined and the grammatical constructions occurring therein had to be identified. To accomplish this goal, randomly chosen paragraphs from the Ami Pro manual were manually analysed and the results were assessed with regard to the complexity and diversity of the grammatical constructions they contained. It turned out that the Ami Pro manual contains a wide range of grammatical constructions and that the grammar would be a very large subset of the full grammar for English.

Based upon the results obtained in the first analysis phase, it was decided that a complete grammar of English would be implemented in order to facilitate the re-usability of the lingware and to allow for further extensions of the object domain of SIFT.

The development process of the grammar was envisaged as an iterative process. In a first step, an existing grammar of English (Quirke and Greenbaum, 1973) was transformed into a PLAIN grammar. Using this initial grammar, a first attempt at parsing the manual was made in order to identify which parts of it could be successfully analysed. Based on these results, the existing grammar was augmented by additional valency templates to increase the percentage of successfully analysed sentences. By repeating these steps, the coverage of the PLAIN grammar was gradually extended.

Two features of the PLAIN grammar formalism made this approach feasible:

- It has been developed in such a way that traditional grammatical descriptions can be easily converted into PLAIN grammars,
- Adding new valency templates to an existing grammar does not effect the analysis obtainable by the original set of templates.

Nevertheless, the Ami Pro manual presents a number of serious grammatical problems to any parsing system which aims at complete analysis of the input sentences. Apart from the well-known problems related to co-ordination, elliptic constructions and prepositional phrase attachment, the usage of ordinary English words in terminological contexts leads to major difficulties. Domain dependent usages are quite different grammatically from domain independent ones, but at the same time can not be distinguished lexically. This means that the parser must account for a large number of special cases by means of additional valency templates which in turn can lead to a combinatorial explosion of parsing results.

The PLAIN system was originally written in PL/I and implemented on mainframe platforms. During the Translator's WorkBench Project, PLAIN was re-implemented in C on Unix systems. In the course of SIFT, the implementation was refined and certain enhancements were introduced, including the addition of a new stage to the parsing algorithm. The resulting system has been separated into a server component and a client portion that can communicate by various means of Internet Procedure Call (IPC) facilities.

Recently, the server module has been separated into a library with a well-documented Application Programmer Interface (API), and a server application that is linked with the PLAIN library. By using this approach, the re-usability of the software was ensured since it is now possible to embed PLAIN into various applications by linking with the PLAIN library. Several small applications (e.g. a stand-alone lemmatiser and a corpus tagger) have been developed to demonstrate the capabilities of the embeddable PLAIN library.

As part of SIFT we carried out an investigation into the capabilities of the Link Parser (LP). Like PLAIN, LP operates bottom-up via a lexicon of syntagmatic templates. The result of grammatical analysis is a *linkage*. This is effectively a set of binary links joining lexemes within the input utterance into a connected graph. We were originally attracted to LP because of its sophistication in handling and/or and list co-ordination which occur very frequently in Ami Pro. For example, a typical utterance is: 'this section describes the hardware, system software, memory, and disk space requirements for using Ami Pro on a stand-alone computer or on a network'. An initial study was carried out (Sutcliffe, Brehony and McElligott, 1994) and this led to the idea of a workshop in which different teams would establish the performance of their parsers when applied to three standard texts derived from technical computer manuals.
A workshop was run at Limerick in 1995 and the results are to appear as a book (Sutcliffe, Koch and McElligott, 1996).

As SIFT developed it became apparent that we needed to be able to extract predicate-argument information from a text but that at the same time a complete analysis of each utterance was not feasible. The main difficulty in the latter case was resolving the usual ambiguities of prepositional phrase attachment and general co-ordination which can not be accomplished automatically with sufficient reliability. However the comparative studies mentioned above suggested that there was a class of constructs and co-ordinations which could be analysed accurately. As an experiment, therefore, a new algorithm was developed which used an input already tagged for part-of-speech and which operated via a series of scans. The idea was that each scan would only look for certain kinds of construct but could use information extracted by previous scans. The result was the Robust Parser. We have carried out some experiments with document indexing using this parser but results are still at a preliminary stage.

3.2.3 Results

- The PLAIN parser system working under UNIX and incorporating a sophisticated syntagmatic grammar of English,
- The Robust Parser, which can extract case frames from free text with reasonable reliability and high efficiency,
- An International Workshop on the Industrial Parsing of Software Manuals (IPSM'95), held at Limerick in May 1995, together with a book describing the results,
- Several detailed studies on parsing text in the technical domain.

3.2.4 Findings

- Accurate and complete parsing of technical text remains a difficult task.
- The main difficulties are co-ordination and prepositional phrase attachment.
- Certain types of grammatical information can, however, by extracted.
- It is of enormous help in parsing if the text is accurately tagged for part-of-speech.

3.3 Text Retrieval

3.3.1 Objectives

The main aim in the text retrieval area was to built a series of prototypes which would enable us to determine the efficacy of the SIFT tools. In particular we were interested in establishing whether the paradigm of distributed representations could be used for information retrieval.

3.3.2 Work Undertaken

The main work carried out was as follows:

- Four sets of queries relating to the Ami Pro manual were collected,
- Correct answers to these queries were determined by experts and encoded in an evaluation database,
- Three SIFT prototypes were built,
- The prototypes were evaluated.

The four query sets are termed the *Schmidt Queries, Hyland Queries, Orion Queries* and *Designer Queries.* The Schmidt Queries were noted by someone as they learned Ami Pro. The Hyland and Orion Queries were collected from volunteer Ami Pro users in Ireland and the United States. The Designer Queries were artificial queries specifically designed by an expert

Ami Pro user to test a conceptual retrieval system by avoiding the use of well-known keywords.

A set of 'correct' answers to each query was established by determining which Level Two sections within the Ami Pro manual could be said to contain information needed to answer the query. Each such set was converted into a list of URLs and stored in a query database.

The three retrieval prototypes combined SIFT components in different ways. In SIFT-1 the manual was indexed via the meanings of atomic concepts while in SIFT-2 and SIFT-3 semantic case frames were used. In all cases indexing information was stored in the RDAFT database. During retrieval via SIFT-1, atomic concepts are extracted from the input query and portions of the manual alluding to similar concepts are searched for in RDAFT. With SIFT-2/3 retrieval is the same except that one or more semantic case frames are extracted from the input query.

The principle behind the evaluation of SIFT was that it should perform better than a tf*idf keyword search engine. In order to test this hypothesis, an interface to the WAIS public domain text retrieval system was created and the Ami Pro manual was converted to allow its use within WAIS in a manner entirely compatible with its treatment within SIFT. This effectively involved treating each Level Two section as a separate document.

3.3.3 Results

- Four evaluation databases each consisting of a set of user queries together with a set of correct answers for each,
- Three SIFT systems, one working with atomic concepts and two working with semantic case frames,
- An interface to the WAIS public domain search engine.

3.3.4 Findings

Evaluation of the SIFT systems is only preliminary because the prototypes themselves require refinement in a number of directions. However, results so far can be summarised as follows:

- In a SIFT system involving only concepts, WAIS tends to perform better. This is because only certain concepts can in fact be indexed due to problems with word sense disambiguation. In the meantime, WAIS can often achieve matches via keywords,
- However, there are a large number of responses which WAIS can not retrieve and which SIFT has the potential to achieve (recall with WAIS is normally about 0.4),
- It seems likely that concept based approaches need to be combined with keyword methods in order to get the best results. This is difficult to accomplish within the present RDAFT retrieval paradigm.

4. Key Findings of the Project

4.1 Concept Retrieval vs. Keyword Retrieval

- Concept based retrieval methods must be combined with keyword techniques if good results are to be obtained. This is because many concepts alluded to in the text can not be identified with sufficient certainty.

4.2 Identifying Concepts Accurately

- Word sense disambiguation can not be performed with sufficient accuracy for concept-based retrieval using current techniques,
- However, technical terms (particularly multiple word terms) tend to be monosemous within the domain. This means that concept based retrieval can be performed using such terminology without solving the problem of disambiguation, providing that it has already been analysed and linked to the core ontology.

4.3 Parsing

- Lotus manuals are written according to a controlled (though complex) grammar. This makes fairly accurate predicate argument extraction a viable proposition. Certain complex constructions (e.g. co-ordination) can also be handled because they are so regular. Complete analysis is not possible, however.

4.4 Interface Design

- The conventional information retrieval processing paradigm, in which a query is input and an ordered list of 'hits' is returned, is not very appropriate to the task domain being studied. More convenient interface paradigms are required.

4.5 Efficiency

- Distributed utterance representations are large and the current comparison methods are too slow.
- The RDAFT retrieval paradigm is too cumbersome to be viable. An approach in which the current representations are combined with the advantages of inverted indexing is required.

5. Demonstrations

5.1 Lexical Database System

The LDB is available from the University of Amsterdam. Figures 3 to 6 give an idea of the system in operation when being used via its graphical user interface.

5.2 SIFT Demonstration Suite

The SIFT demonstration suite is accessible over the World Wide Web at URL nlp01.cs.ul.ie/sift.html. At present the following demonstrations are available for you to try:

5.2.1 SIFT

This is a demonstration version of SIFT which does not perform searches of the entire manual but only the Level 2 sections (see Figures 7 and 8).

5.2.2 Term Recogniser 1

The first recogniser highlights terminology in the input based on regular expression matches on sequences of part-of-speech tags.

5.2.3 Term Recogniser 2

The second recogniser highlights terminology by using a large database of Ami Pro technical terms (see Figures 9 and 10).

5.2.4 Brill Tagger

This is an interface to the Brill Tagger (Brill, 1992).

5.2.5 Robust Parser

As described above, the Robust Parser recognises prepositional phrases, noun phrases and verb groups, including those involving complex co-ordination, using a robust and efficient multiple scan algorithm (see Figures 11 and 12).

5.2.6 Case Frame Extractor

The Case Frame Extractor uses the output of the Robust Parser and extracts semantic case frames for each predicate-argument grouping based on a case lexicon developed for the project, together with a system of defaults used for unknown verbs (see Figures 13 and 14).

5.2.7 Link Parser

This is an interface to the Link Parser (Sleator and Temperley, 1991). The parser was evaluated as part of the project and was used within the IPSM'95 parsing workshop.

5.2.8 Word Stemmer

The Word Stemmer is an interface to the Wide Coverage Morphological Analyser (Karp, Schabes, Zaidel and Egedi, 1992).

5.2.9 Lexical Analyser

The Lexical Analyser is a simple tool which can be programmed to tokenise an input according to different criteria.

6. Next Steps

6.1 Inverted Indexing

Our research indicates that for best results keyword and concept methods need to be combined. The present retrieval paradigm via RDAFT does not allow this. However, we have devised an inverted indexing scheme which allows concepts to be indexed alongside keywords. This is currently being implemented.

6.2 Interface Design

The present user interface is very simple. However, during the project a number of ideas have emerged in relation to alternative approaches which are particularly relevant to the computer manual domain. In particular the concept of retrieval needs to be combined with the notion of document navigation in a more felicitous manner. At the moment the user has no idea where they are in the manual or where else they might be.

6.3 New Retrieval Paradigms

At present the method of retrieval being used is directly derived from the traditional information retrieval domain based on the notion of input queries and large heterogeneous document collections. We plan to experiment with other paradigms such as those based on the relationship between tasks defined in the manual and the operations needed to accomplish them.

7. 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, ANLP, Trento, Italy, 1992.*

Gale, W. A., Church, K. W., & Jarowsky, D. (1992). A Method for Disambiguating Word Senses in a Large Corpus. *Computers and the Humanities,* **26**(5-6), 415-439.

Ide, N. M., & Véronis, J. (1990). Very Large Neural Networks for Word Sense Disambiguation. *Proceedings of the European Conference on Artificial Intelligence, ECAI'90, August 1990, Stockholm, Sweden.*

Karp, D., Schabes, Y., Zaidel, M., & Egedi, D. (1992). A Freely Available Wide Coverage Morphological Analyzer for English. *Proceedings of the 14th International Conference on Computational Linguistics (COLING'92)*, 950-955.

Lesk, M. (1986). Automated Word Sense Disambiguation Using Machine-Readable Dictionaries: How to tell a Pine Cone from an Ice Cream Cone. *Proceedings of the ACM SIGDOC Conference, June 1986, Toronto, Ontario.*

Proctor, P. (Ed.) (1978). *The Longman Dictionary of Contemporary English (LDOCE).* London, UK: Longman.

Quirk, R., & Greenbaum, S. (1973). *A University Grammar of English.* Harlow, UK: Longman.

Rada, R., Mili, H., Bicknell, E., & Blettner, M. (1989). Development and Application of a Metric on Semantic Nets. *IEEE Transactions on Systems, Man, and Cybernetics,* **19**(1), 17-30.

Resnik, P. S. (1994). Selection and Information: A Class-Based Approach to Lexical Relationships (IRCS Report 93-42). Philadelphia, PA: University of Pennsylvania, Institute for Research in Cognitive Science.

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

Salton, G. (Ed.) (1971). *The SMART Retrieval System - Experiments in Automatic Document Processing*. Englewood Cliffs, NJ: Prentice-Hall.

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

Sleator, D. D. K., & Temperley, D. (1991). *Parsing English with a Link Grammar* (Technical Report CMU-CS-91-196). Pittsburgh, PA: Carnegie Mellon University, School of Computer Science.

Sussna, M. (1993). Word Sense Disambiguation for Free-Text Indexing Using a Massive Semantic Network. *Proceedings of the Second International Conference on Information and Knowledge Management, 1993, Arlington, VA.*

Sutcliffe, R. F. E., Brehony, T., & McElligott, A. (1994). The Grammatical Analysis of Technical Texts using a Link Parser. In *Proceedings of the Second Conference of the Pacific Association for Computational Linguistics, PACLING'95, 19-22 April 1995, The University of Queensland, Brisbane, Queensland, Australia*.

Sutcliffe, R. F. E., Koch, H.-D., & McElligott, A. (1996). *Industrial Parsing of Software Manuals*. Amsterdam, The Netherlands: Rodopi.

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.

8. SIFT Bibliography

The following list comprises published articles and publicly available deliverables.

Boersma, P. (1996). *The Sift Lexical DataBase* (SIFT deliverable D10b, D21a, LRE 62030, June 1996). Amsterdam, The Netherlands: University of Amsterdam, Computer Centrum Letteren.

Donker, T., Serail, I., & Vossen, P. (1994). *Salient Words and Phrases* (SIFT deliverable D5, LRE 62030, March 1994). Amsterdam, The Netherlands: University of Amsterdam, Computer Centrum Letteren.

Donker, T., & Vossen, P. (1994). *The SIFT Syntactic Lexicon* (SIFT deliverable D21, LRE 62030, December 1994). Amsterdam, The Netherlands: University of Amsterdam, Computer Centrum Letteren.

Masereeuw, P. (1994). *Ltree and CM: The SIFT LDB developer's libraries* (SIFT deliverable D10, LRE 62030). Amsterdam, The Netherlands: University of Amsterdam, Computer Centrum Letteren.

O'Sullivan, D., McElligott, A. & Sutcliffe, R. F. E. (1995). Augmenting the Princeton WordNet with a Domain Specific Ontology. *Proceedings of the IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing, 19-21 August, 1995, Montreal, Canada.*

O'Sullivan, D., Sheahan, L., McElligott, A., & Sutcliffe, R. F. E. (1995). Concept-Based Searching of Technical Documents (Extended Abstract). In I. Richardson and N. Power (Eds.) *Proceedings of the Second Annual Computer Science and Information Systems Research Conference, University of Limerick, Tuesday 12 September, 1995.*

Sutcliffe, R. F. E. (1994). *Proposal for a Software Standard for the SIFT Project* (Technical Note, 31 January 1994). Limerick, Ireland: University of Limerick, Department of Computer Science and Information Systems.

Sutcliffe, R. F. E. (1994). *A Study of Syntactic Structure and Semantic Case in Chapter 3 of the Lotus Ami Pro User's Guide Release 3* (Technical Report UL-CSIS-94-6, April 1994). Limerick, Ireland: University of Limerick, Department of Computer Science and Information Systems.

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.

Sutcliffe, R. F. E., & Slater, B. E. A. (1994). The Use of Syntactic Tagging in Word Sense Disambiguation: Two Methods and their Comparison. *Proceedings of the 1994 Joint Conference of the 8th Asian Conference on Language, Information and Computation and the 2nd Pacific Asia Conference on Formal and Computational Linguistics, (ACLIC94/PacFoCoL94), 10-11 August, 1994, Shiran Kaikan, Kyoto, Japan.*

Sutcliffe, R. F. E., O'Sullivan, D., Sharkey, N. E., Vossen, P., Slator, B. E. A., McElligott, A., & Bennis, L. (1994). A Psychometric Performance Metric for Semantic Lexicons. *Proceedings of International Workshop On Directions Of Lexical Research, 15-17th of August, 1994, Beijing, China.*

Sutcliffe, R. F. E., O'Sullivan, D., & Hellwig, P. (1994). The Representation of Nouns by Distributed Patterns Constructed via Taxonomic Traversal. In *Proceedings of 2. Konferenz ``Verarbeitung natuerlicher Sprache" (KONVENS94), 28-30 September, 1994, University of Vienna, Austria.*

Sutcliffe, R. F. E., & Slater, B. E. A. (1994). Word Sense Disambiguation of Text by Association Methods: A Comparative Study. In *Actas Del X Congreso, Sociedad Española para el Procesamiento del Lenguaje Natural (10th Annual Meeting of The Spanish Association For Natural Language Processing) (SEPLN'94), 20-22 July, 1994, Córdoba, Spain*.

Sutcliffe, R. F. E., O'Sullivan, D., Slater, B. E. A., & Brehony, T. (1994). Traversing WordNet to Create Optimised Semantic Lexical Representations. In *Proceedings of the Seventh Annual Irish Conference on Artificial Intelligence and Cognitive Science (AICS'94), Trinity College Dublin, 8-9 September, 1994.*

Sutcliffe, R. F. E., & Slater, B. E. A. (1994). The Assessment of Two Algorithms for Disambiguation by Association. In *Proceedings of the Seventh Annual Irish Conference on Artificial Intelligence and Cognitive Science (AICS'94), Trinity College Dublin, 8-9 September, 1994.*

Sutcliffe, R. F. E., O'Sullivan, D., & McElligott, A. (1994). Creating a Large Semantic Lexicon for Nouns. *Proceedings of the Second International Conference on Quantitative Linguistics (QUALICO'94) Moscow, Russia, 20-24 September 1994.*

Sutcliffe, R. F. E., & Slater, B. E. A. (1994). Disambiguation by Association: Two Algorithms and their Assessment. *Proceedings of the Second International Conference on Quantitative Linguistics (QUALICO'94), Moscow, Russia, 20-24 September 1994.*

Sutcliffe, R. F. E., O'Sullivan, D., & McElligott, A. (1995). The Creation of a Semantic Lexicon by Traversal of a Machine Tractable Concept Taxonomy. *Journal of Quantitative Linguistics*, **2**(1), 33-42.

Sutcliffe, R. F. E., & Slater, B. E. A. (1995). Disambiguation by Association as a Practical Method: Experiments and Findings. *Journal of Quantitative Linguistics*, **2**(1), 43-52.

Sutcliffe, R. F. E., Vossen, P., Hellwig, P., McElligott, A., O'Sullivan, D., Relihan, L., Sheahan, L., & Slater, B. (1994). The Tractable Representation of Utterance Meanings for Information Retrieval. Abstract in *Proceedings of the Fifth Computational Linguistics in the Netherlands Meeting (CLIN'94), Wednesday November 23, 1994, University of Twente, The Netherlands.*

Sutcliffe, R. F. E., Hellwig, P., Vossen, P., O'Sullivan, D., Relihan, L., & McElligott, A. (1994). SIFT, a Hybrid Retrieval Engine for Providing Help from Technical Computer Manuals. *Proceedings of the 17th Annual Colloquium of the British Computer Society Information Retrieval Special Interest Group, IRSG'95, Manchester Metropolitan University, 4-5 April 1995.*

Sutcliffe, R. F. E., O'Sullivan, D., Sheahan, L., & McElligott, A. (1994). The Automatic Acquisition of a Broad-Coverage Semantic Lexicon for use in Information Retrieval. *Proceedings of the AAAI Spring Symposium `Representation and Acquisition of Lexical Knowledge: Polysemy, Ambiguity and Generativity', Stanford University, California, 27-29 March, 1995.*

Sutcliffe, R. F. E., Brehony, T., & McElligott, A. (1994). The Grammatical Analysis of Technical Texts using a Link Parser. In *Proceedings of the Second Conference of the Pacific Association for Computational Linguistics, PACLING'95, 19-22 April 1995, The University of Queensland, Brisbane, Queensland, Australia.*

Sutcliffe, R. F. E., Vossen, P., Serail, I., Masereeuw, P., Hellwig, P., Boersma, P., Bon, A., McElligott, A., O'Sullivan, D., & Sheahan, L. (1994). From SIFT Lexical Knowledge Base to SIFT Lexical Data Base: Creating a Repository for Lexicological Research and Development. Abstract in *Proceedings of the Workshop on Linguistic Databases, LINGDB'95, 23-24 March 1995, University of Groningen, Groningen, The Netherlands.*

Sutcliffe, R. F. E., Brehony, T., & McElligott, A. (1994). Link Grammars and Structural Ambiguity: A Study of Technical Text. Technical Note UL-CSIS-94-15, Department of Computer Science and Information Systems, University of Limerick, December 1994.

Sutcliffe, R. F. E., Hellwig, P., Vossen, P., O'Sullivan, D., Relihan, L., McElligott, A., Slater, B., Brehony, T., & Sheahan, L. (1994). *Retrieval from Software Instruction Manuals by Semantic Skimming* (Technical Report UL-CSIS-94-16, December 1994). Limerick, Ireland: University of Limerick, Department of Computer Science and Information Systems.

Sutcliffe, R. F. E., Boersma, P., Bon, A., Donker, T., Ferris, M. C., Hellwig, P., Hyland, P., Koch, H.-D., Masereeuw, P., McElligott, A., O'Sullivan, D., Relihan, L., Serail, I., Schmidt, I., Sheahan, L., Slater, B., Visser, H., & Vossen, P. (1995). Beyond Keywords: Accurate Retrieval from Full Text Documents. *Proceedings of the 2nd Language Engineering Convention, Queen Elizabeth II Conference Centre, London, UK, 16-18 October 1995.*

Sutcliffe, R. F. E., Koch, H.-D., & McElligott, A. (1995). *Proceedings of the International Workshop on Industrial Parsing of Software Manuals, 4-5 May 1995, University of Limerick, Ireland.*

Sutcliffe, R. F. E., & McElligott, A. (1995). Using the Link Parser of Sleator and Temperley to Analyse a Software Manual Corpus. In *Proceedings of the International Workshop on Industrial Parsing of Software Manuals, 4-5 May 1995, University of Limerick, Ireland.*

Sutcliffe, R. F. E., & O'Sullivan, D. (1995). Sifting the Contents of Technical Documents (Abstract). *The Irish Scientist*, **3**, 64-64.

Sutcliffe, R. F. E., Koch, H.-D., & McElligott, A. (1996). *Industrial Parsing of Technical Manuals*. Amsterdam, The Netherlands: Rodopi.

Vossen, P. (1996). Semantic cases in the Sift Lexicon (SIFT deliverable D31 and D33a, LRE 62030). Amsterdam, The Netherlands: University of Amsterdam, Computer Centrum Letteren.

Vossen, P., Boersma, P., Bon, A., & Donker, T. (1995). A flexible semantic database for information retrieval task*s. Proceedings of the AI'95, June 26-30, Montpellier, France.*

Vossen, P., & Bon, A. (1996). *Building a semantic hierarchy for the SIFT project* (SIFT deliverable D20a, LRE 62030). Amsterdam, The Netherlands: University of Amsterdam, Computer Centrum Letteren.

Vossen, P., Bon, A., & Donker, Ton (1996). *The semantic features in the SIFT Lexicon* (SIFT deliverable D20b, LRE 62030). Amsterdam, The Netherlands: University of Amsterdam, Computer Centrum Letteren.

Vossen, P., & Donker, T. (1996). *The Subvocabulary in the SIFT project* (SIFT deliverable D47, D47a and D51, LRE 62030). Amsterdam, The Netherlands: University of Amsterdam, Computer Centrum Letteren.