Language engineering for the recovery of requirements from legacy documents

Paul Rayson, Roger Garside and Pete Sawyer
Computing Department
Lancaster University
Lancaster
UK. LA1 4YR
Tel: +44 1524 593780
Fax: +44 1524 593608
{paul, rgg, sawyer}@comp.lancs.ac.uk

Legacy documents, such as requirements documents or manuals of business procedures, can sometimes offer an important resource for informing what features of legacy software are redundant, need to be retained or can be reused. This situation is particularly acute where business change has resulted in the dissipation of human knowledge through staff turnover or redeployment. Exploiting legacy documents poses formidable problems, however, since they are often incomplete, poorly structured, poorly maintained and voluminous. This report proposes that *language engineering* using tools that exploit probabilistic natural language processing (NLP) techniques offer the potential to ease these problems. Such tools are available, mature and have been proven in other domains. The document provides a review of NLP and a discussion of the components of probabilistic NLP techniques and their potential for requirements recovery from legacy documents. The report concludes with a summary of the preliminary results of the adaptation and application of these techniques in the REVERE project.

1. Introduction

Many organisations react to changes to their business environment by changing their strategic business goals and reengineering their organisational structures. This often dramatically changes the requirements of the socio-technical systems used at the operational level to implement the business processes. A precondition for understanding the implications of the changed requirements is an understanding of the systems' original requirements. Business reorganisation often means that this understanding is hard to acquire because continuity of experience has been lost. This paper reports the preliminary results of the REVERE[†] project where we are concerned with helping organisations to cope with this. We are investigating the recovery of requirements from documentation which frequently comprises an important element of the remaining organisational memory.

There are many types of legacy and classifications of change (e.g. [Lam 98]). We use that derived by Alderson and Shah [Alderson 99]:

- *Strategic*: the boardroom-level view of how events in the business environment affect the business. Consequent change to the business might be imposed (e.g. by changed legislation) or it might be opportunistic (e.g. new business opportunities).
- *Organisational*: how the business structure supports the strategy. Changes to the strategy may necessitate reengineering the business. This may be horizontal, where roles and responsibilities are redefined, or vertical where support for new business sectors are introduced and redundant business areas are stripped away.
- *Operational*: the socio-technical systems that put the business processes into operation. If business change degrades their support for the business processes in terms of function, throughput, reliability, compliance with regulation, etc. they must be adapted accordingly.
- *Developmental*: the development and maintenance of the hardware and software that implement the automated parts of the socio-technical systems. Legacy may be a consequence of changes at the levels above but, it may also be independent of these; for example because of a critical shortage of COBOL programmers needed to maintain the software.

[†] REVerse Engineering of REquirements. EPSRC Systems Engineering for Business Process Change (SEBPC) programme project number GR/MO4846. Further details can be found at: http://www.comp.lancs.ac.uk/computing/research/cseg/projects/revere/. We are grateful for support from our industrial partner, Adelard, who provided us with motivation, data and advice.

Our work is motivated by our industrial partner's experience of tackling organisational change that has *already occurred* (figure 1). In a typical scenario, change to the business strategy has led to both vertical and horizontal reengineering of the organisation and its business processes. The pace and/or scale of change has inhibited adaptation of the operational systems. Consequently, the organisation finds itself in a position where its new business processes are inadequately supported by the legacy software. This is illustrated by the experience of UK clearing banks. After many decades of relative stability, recent changes to the global financial services market have caused their core business to change from administering accounts to selling financial products [Blythin 97]. As a result, they have a legacy of systems they cannot do without, but which inadequately support their new business. They still need to manage customers' accounts but they also need to support marketing requirements by, for example, building customer profiles from account data.

In a situation such as this, the legacy software needs to be adapted by, (e.g.) evolution or replacement. However, this change must be informed not only by the requirements of the changed business but also by the requirements that originally motivated the legacy systems. In Figure 1, these are depicted by the grey block arrows bearing onto the existing operational software. These requirements may be derived from many levels; from the end-users who enact the business processes to strategic business goals. It is often tacitly assumed that while user requirements are relatively volatile, requirements that are consequent on the business strategy are relatively stable. However, changes to the business environment in the form of new legislation, globalisation and introduction of the Euro (among many others) show this assumption to be unsafe.

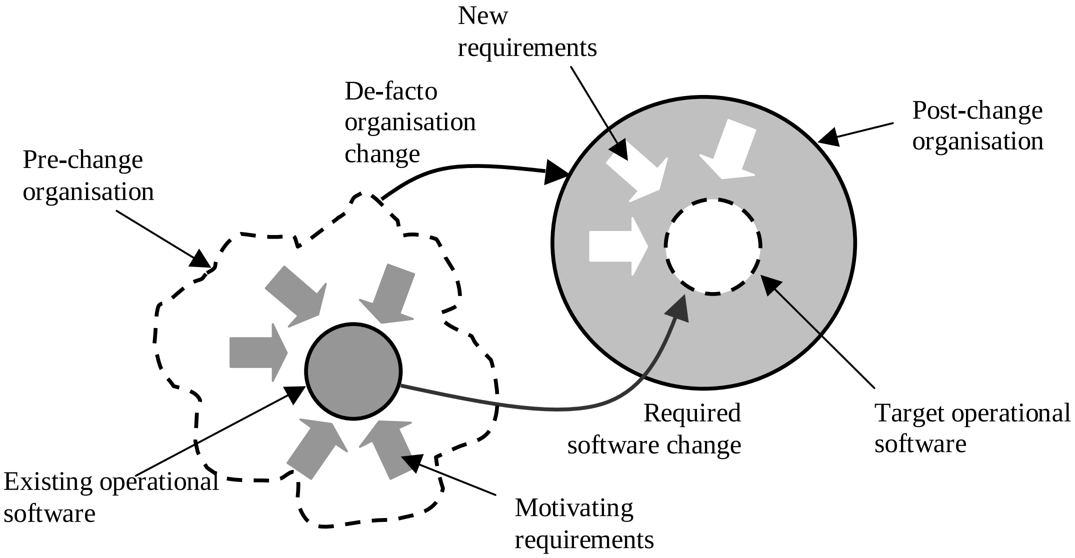


Figure 1. Legacy software and organisational change

Failure to understand the legacy software requirements and their motivation means that rational choices about how to adapt the software cannot be made. It will be uncertain what features of the software are redundant, need to be retained or can be reused. This uncertainty can lead to throwing the baby out with the bath water; solutions that support new business processes but fail to support key requirements that persist from the old business. More often, organisations dare not risk this happening and adopt costly solutions that retain functionality or data that is redundant. The uncertainty that leads to this is often a consequence of the loss of experienced people able to answer questions like "why does the system keep this data?".

This places a premium on the experience that is available and on other sources of the information. These are typically the legacy software itself and documents such as requirements specifications, operating procedures, regulatory standards, etc. In most cases these sources will be incomplete but complementary and the analyst will need to use each to construct partial models that can be verified against each other. The REVERE project is specifically concerned with legacy requirements recovery from documents. Because these documents are, like the software itself, a legacy of an earlier state of the organisation, we use the term *legacy documents* to refer to them.

2. Requirements recovery from documents

The process of recovering the requirements for existing software systems from fragmentary sources of information is analogous to following multiple audit trails that lead from requirements inferred from the domain or business into technical documentation and through to the software. The analyst must use whatever information resources are available to construct conceptual models of the pre-change organisation and its business processes and from these derive the requirements of the legacy software [Butler 99]. This typically entails an iterative process of inferring stakeholders, roles, tasks and business objects and verifying these against the structure and behaviour of the in-service software.

This information has to be gathered from many different sources, both human and documentary. The elicitation of information from human stakeholders has received a great deal of attention elsewhere and there has also been some work on the reverse engineering of source code. The retrieval of requirements information from free text documents has, by comparison, been neglected.

Good requirements engineering practice [Sommerville 97] recommends that requirements are carefully documented and managed. Hence, it should be straightforward to list the stakeholders' requirements, understand their motivation and consequent trade-offs, and trace them forwards (ultimately) into the operational software. Unfortunately, good requirements practice is rare and systems are still routinely constructed with minimal requirements documentation [Melchisedech 98]. Some domains (e.g. defence) place a premium on documentation and here it is reasonable to expect requirements specifications, operating procedures, safety cases, etc. to be available. In most cases the available legacy documentation will be less comprehensive but if any does exist, it will represent a potentially important resource; particularly where human expertise is patchy. Given the existence of *some* legacy documentation, the real problem is then how to process it. This will be hard if the volume is large; in extreme cases, there may be filing cabinets full. Similarly, variable quality and the structure of the documents will pose problems if, for example, they are heavily cross-referenced and version control has been poor. This is compounded by the linear structure of paper documents. Even if the legacy documents have good tables of contents, have comprehensive indexes and are in, or can be transformed (via scanning and OCR) into, electronic form, the documents' as-written structure inevitably constrains the way in which people can read and interact with them.

Identification and assimilation of the subset of useful information contained in the legacy documents is therefore difficult, costly and error-prone.

Our aim is to develop tools to ease these problems by exploiting mature techniques for natural language processing. Although technical documents often employ special notations, such as object models, workflow diagrams, etc., the bulk of nearly all such documents is comprised of natural language. This has led a number of researchers in requirements engineering to investigate the use of NLP. Ryan [Ryan 93] has sounded a note of caution by noting the limitations of NLP in requirements engineering. In particular, he notes that attempts to "automate" the analysis or generation of natural language requirements are doomed. This is both because of the limitations to natural language understanding that we describe below, and because of the fact that understanding a system has to be gained by consideration of the system in its context. Hence, social, cultural and political factors are all as important for a system's success as technical issues and these factors are seldom documented. Interestingly, however, he also notes some practical potential for the use of NLP in requirements engineering: "tools to scan, search, browse and tag [the] text could assist in developing a full and accurate statement of needs". We accept both parts of Ryan's argument; that the utility of NLP is restricted *but* that it offers some potential for leveraging the free text that forms the bulk of most requirements documents.

However, Ryan's caution is not universally been accepted. Several researchers have used rule-based NLP techniques for synthesing database conceptual schemas [Rolland 92], generating graphical representations of VLSI system requirements [Cyre 97], generating algebraic specifications from NL requirements [Ishihara 93] and automatically abstracting requirements from requirements specifications [Goldin 97]. However, these approaches are all hamstrung by the limitations of the rule-based NLP techniques they employ. Natural language (NL) is invariably so complex that a very large base of grammar rules is required even for small NL subsets. All the techniques above depend for their efficacy on a tightly constrained subset of English.

Analysts of legacy software have no control over the legacy documents that they must analyse so purely rule-based techniques are impractical. Our approach is therefore to exploit *probabilistic* NLP techniques that were pioneered at Lancaster and a number of other sites in the 1980s. These techniques have been proven in other domains, they are robust and they scale. They do not attempt to automate understanding of text but rather they abstract interesting properties that provide the user with a ’quick way in’ to large documents; by providing Ryan's tools to "… scan, search, browse and tag the text…".

In the next section, we place probabilistic NLP techniques in context and describe their underlying principles and their advantages over rule-based NLP.

3. Overview of NLP Techniques

This section discusses the field of Natural Language Processing, the use of computer programs to provide some degree of analysis of natural language texts. This was an early application of computers, with attempts at English-Russian translation in the early 1950s. Grandiose claims were made about the success of this application area, which inevitably foundered on the inadequacies at the time of the hardware (particularly the storage capacity) and software, and the lack of computational linguistic theory and machine readable resources such as dictionaries.

In this section we concentrate on *analysis*; that is, given a natural language text, what information can be extracted from it (with the ultimate goal to extract the "meaning"). Work has also been done on language *generation*; that is, given some information to express (perhaps an explanation of an expert system's recommended action), how it can be turned into a plausible natural language text. Much more work has been done on analysis than generation.

Similarly, we concentrate on English. Much of the early research in computational linguistics was done on English, and it is still the predominant language in NLP. However, increasingly people are working on the main European languages and Japanese, and work has begun on other languages.

3.1 The Problems of NLP Analysis

Since humans have little difficulty extracting meaning from natural language texts, it is at first unclear why computers have such a problem. It is rare that one is consciously aware of syntactic ambiguity in reading or listening to text.

> The Saudi Arabian Government plans to buy 80 or more British Aerospace Hawke ground attack aircraft, Westland helicopters and Vosper Thornycroft minehunters have been hit by cuts in the Saudi military budget.

Figure 2. An example of syntactic ambiguity[1]

A favourite example is shown in Figure 2, where the word *have* is likely to require the reader to restart the sentence and parse *plans* as a plural noun, rather than the third person singular present indicative of a verb. Typically, if we attempt to parse much simpler sentences than this with an automatic procedure we will end up with thousands or even millions of alternative parses. There will be lexical ambiguity (where a single word, such as *round*, has several alternative parts of speech), and structural ambiguity (where a part of the sentence is capable of being interpreted in terms of several different parse structures). This ambiguity is exacerbated by ellipsis, where recoverable parts of a sentence can be omitted (for example, the word *eats* is recoverable in the sentence *I eat fish and Douglas chips*, because of the parallel structure). It is clear that humans make use of real-world knowledge and the discourse context to reduce the search space, a process which it is difficult to mechanise.

Suppose we have automatic techniques to apply various types of linguistic and perhaps other information sources to the problem of analysing a text, then a further issue is how this knowledge is to be obtained in a sufficiently explicit form for use. In other words, how do we train the system?

3.2 Levels of analysis

The problem of automatically extracting the "meaning" from a text read from the page, or transcribed from a speech-recognition system, is too complex for one step. We usually think of several different levels of

[1] From CAAT News, February 1994, p4.

analysis applied to the text, in increasing difficulty, building one upon the other. This also gives us a natural way of breaking down the analysis problem into several separate sub-problems (although it is likely that in human language processing there is feedback from one level to an earlier one). Typical levels are [Leech 1997]:

3.2.1 The lexical or word-tagging level.

Here we are simply attempting to allocate a part of speech to each word in the text, for example see Figure 3.

Sentence:	***time***	***flies***	***like***	***an***	***arrow***
POS:	Singular common noun.	Third person singular present indicative of verb.	Preposition.	Singular article.	Singular common noun.

Figure 3. POS categories for one sentence

Notice that all of the first three words could be allocated different parts of speech in other circumstances, a particularly acute problem with the English language.

This level of analysis has sometimes been considered simply as a sub-part of the syntactic analysis, but it has become clear that this is a simpler problem but still one worth attempting to solve for useful applications. Different degrees of delicacy of analysis can be applied; we might make a simple noun versus verb distinction, or we might attempt to mark a number of subdivisions within the noun class.

3.2.2 The syntactic level.

Here we are attempting to supply a linguistic analysis of the sentence according to a suitable grammar of the language, without at this stage being concerned with the meaning or semantics of the sentence.

[sentence [noun phrase ***the cat***] [verb phrase ***sat*** [preposition phrase ***on*** [noun phrase ***the Persian mat***]]]]

Figure 4. Parsing labels for one sentence

There is an assumption here that a clear distinction can be made between syntax and semantics, as is attempted in compiler theory; there are problems in detail with this borderline, but we can usually make a workable distinction. The result is usually expressed in terms of a parse tree, although there are various other equivalent notations, such as dependency grammar. Again different degrees of delicacy can be envisaged. For example, we could mark noun phrases with their relation (subject, indirect object, etc.), if any, to the main verb. Or we could mark the location of a moved or omitted element. For example consider Figure 5.

(a) I lost the book. You were reading the book.
(b) I lost the book which you were reading.

Figure 5. An example of an omitted element

In some grammar theories, the sentence (b) is constructed from the pair of notional sentences (a) by replacing the second occurrence of the phrase *the book* with the word *which*., and then merging the two sentences into one. But the word *which* has been moved to the beginning of the subordinate clause *which you were reading*, rather than be in the normal position where a direct object would be expected, after the verb *to read.*

3.2.3 The semantic level.

Here we are attempting to link words to their counterparts in the "real world". If we distinguished noun *spring* or *table* from verb *spring* or *table* at the syntactic level, we now want to decide whether *spring* is a season, water source or metal coil, and whether *table* is made of lines or made of wood. Obviously context, or topic of discourse, is a helpful signal, though perhaps less so in sentences like the one in Figure 6.

the astronomer married the star

Figure 6. A semantically ambiguous sentence

3.2.4 The pragmatic or discourse level

Here we are attempting to mark the "meaning" elements which are related to the particular context of utterance of the sentence, where the semantic level is more concerned with the fixed world context. Thus *it's hot in here* could, in one context, be a statement about ambient temperature, and in another a request to open a window or turn down the heating. A very significant problem at this level is anaphor resolution; that is, what is the referent of the pronoun *it* in the sentences shown in Figure 7.

the committee condemned the meat because it found evidence of infection
the committee condemned the meat because it contained evidence of infection

Figure 7. Anaphor resolution example

3.2.5 Other possible levels of analysis

For example, we could mark the prosody of a sentence, which might be important in a text-to-speech system in distinguishing statements from questions. We ignore these other levels in this report.

3.3 Rule-based systems

Let us consider the syntactic analysis problem; here is a sentence, what is the corresponding parse tree? The traditional answer to this, corresponding to a mechanism which is very effective for programming languages, is to have a linguist write down a grammar for the language in a suitable notation, and then search for a parse tree labelled with rules from this grammar from which we can derive the given sentence. There are various suggestions for the "suitable notation", but most of these could, at least in theory, be rewritten as a context-free grammar [Winograd 83]. There is an amusing on-going argument about whether the syntax of a natural language is in fact context-free [Pullum 82], but the argument at least shows that any non-context-free part is rather obscure. The notations (such as GPSG [Gazdar 85]) are not usually directly context-free, because of the wish to capture the generalisations which linguists can make, such as that subject and object noun phrases are largely the same (at least, apart from pronouns, in English), and that the larger structure of singular and plural noun phrases is also the same.

What are the problems with this line of approach? Essentially the problem is the size of the rule-base. If we attempt to write a grammar for a large fraction of the English language, taking into account phenomena like ellipsis, we would require a grammar of at least several hundred thousand rules if written in context-free form. Other notations may theoretically be more parsimonious, but they typically require parameterisation by the words of the lexicon, so the dictionary of the words of the language becomes a large and complex rule-base. We may also require a rule-base representing real-world knowledge, probably on a per-domain basis, to attempt to help with the disambiguation. There are two problems with this:

a) building these large rule-bases, presumably by elicitation from a team of linguists and lexicographers.
b) searching the rule-base for the combination (or, more likely, combinations) of rules which will allow the given sentence to be derived.

3.4 Statistically based systems

An alternative way of tackling NLP problems is by probabilistic methods. For a number of years Chomsky and others scorned these techniques, as not a serious competitor to rule-based methods. However, in the mid to late 1980s, at a number of centres including Lancaster University's UCREL (University Centre for Computer Corpus Research on Language), probabilistic methods were used to tackle the lexical or tagging level of assigning a part of speech to each word in the text [Marshall 83] [Church 88]. Success rates with accuracy in the high 90% range were reported, and now nearly all serious word-tagging systems are probabilistic. Similar techniques are being applied to other NLP tasks.

Let us consider the problem of assigning part-of-speech tags to words in a text from a different point of view, considering it as an example of a Hidden Markov Model (HMM) [Jelinek 90]. A Markov model is an entity which changes from state to state according to a set of probabilities. In a hidden Markov model, we cannot directly observe the state the entity is in; instead, the entity generates a sequence of output symbols with a probability which depends on the state that the entity is currently in. Then there are efficient procedures for finding the most likely sequence of hidden states for a given sequence of observed output symbols.

Now we apply this to the word-tagging problem. The hidden states are the possible parts of speech and the observed symbols are the actual words of the sentence. We imagine that the speaker is a Markov model going from one part-of-speech state to another, and generating the words from the states. The transition from the adjective state to the noun state is likely to have a high probability, while the transition from the noun state to the adjective state is likely to have a low probability. Similarly, while in the noun state, some words (*man, woman, dog, car,* etc.) are more likely to be generated than others (for example, *over* except in rather restricted domains). So if we have collected transition probabilities from every possible state to every other possible state, and further probabilities of generating particular words from each part-of-speech state, then the HMM theory gives us a way of calculating the most likely part-of-speech sequence for a given sequence of words. We take this to be the parts of speech to be assigned to the words in that sentence.

This technique works well, with accuracy in the high 90%. It is also robust; that is, it always generates some output for any input, although perhaps with a lower accuracy rate for inappropriate input. This is in contrast with rule-based methods, which fail catastrophically when they fail; if there is no appropriate set of rules, we get no answer at all.

A problem with statistically based techniques is where to get the required probabilities. There are two main methods:

a) We collect a *corpus*, a large representative collection, of texts. We use a team of linguists to annotate each word with the appropriate part of speech (with suitable procedures to maximise consistency and correctness, etc.). Then we use this to extract estimates for the probabilities required. At Lancaster, we have sometimes supplemented this with judgements from linguists where insufficient data has been collected. For example, we might add an entry that the word *over* could be a noun as well as a preposition, but only very rarely (without trying to put an exact numerical value on the rarity) [Garside 96].

b) The HMM theory gives us another mechanism for generating estimates for these probabilities. If we have a large corpus of (un-annotated) text, we can use an iterative algorithm (called the forward-backward algorithm) to generate a sequence of improved estimates from an initial crude guess. In practice, the algorithm is a hill-climbing procedure which searches for the set of probabilities which maximises the likelihood of the text corpus.

Both of these techniques rely on there being a corpus of texts from which we can derive the estimates, so teams involved in statistically-based NLP techniques are, of necessity, involved in building up corpora of suitable texts. We have, of course, to ensure that these corpora are large enough to make the estimates reasonably accurate - perhaps several million words. And they have to be representative of the texts upon which we aim to do the word-tagging.

3.5 Summary

Rule-based NLP systems can produce a deep analysis, according to a set of rules hand-crafted by a linguist or domain expert. But the systems will be fragile, delivering results only on a constrained set of input texts. Statistical NLP systems tend to generate shallower analyses, but are robust - they will generate results for a wide range of authentic texts. It is possible to attempt a hybrid approach; to start with a basically probabilistic mechanism, and then to modify it to apply rules either to extend the analysis or to invoke additional context to improve the accuracy rate. UCREL's current word-tagging system, CLAWS4, uses such a hybrid approach [Fligelstone 96].

4. NLP techniques in UCREL research

UCREL has always concentrated on the application of probabilistic and hybrid approaches in automatic text analysis at various levels. In this section we focus on the methods and tools developed by UCREL.

The research performed by UCREL has focused mainly on annotation of corpora within the corpus linguistics methodology. There is no defined minimum or maximum size for a corpus, or of what it should contain. A corpus could contain the entire works of Shakespeare, sets of instructions from washing powder boxes, or the text of the match-day programmes from Nottingham Forest Football Club in the season they won the League Championship. Corpora need not only contain written language, s*poken corpora* can be built by transcribing the recorded speech from, for example, news broadcasts or conversations of people giving directions in the street. Corpora are usually collected with a particular research project in mind, such as providing frequency information for dictionary entries or advanced language learning of German [Jones 97]. Sometimes corpora are collected without a specific purpose and are made available as a general language resource to linguists, social scientists, language teachers, market researchers and others. In recent years with the advent of CD-ROMs and the World Wide Web (WWW), a corpus can be a *multimedia corpus* which includes still pictures, video and sounds.

The term *corpus linguistics* has been described [McEnery 96] in simple terms as the study of language based on examples of ‘real life’ language use. It has a relatively long history. Corpus linguistics is not a branch of linguistics such as syntax, semantics and pragmatics which concentrate on describing or explaining some aspect of language use. It is a methodology that can be applied to a wide range of linguistics.

[Leech 97] defines *corpus annotation* as the practice of “adding interpretative, linguistic information to an electronic corpus”. It adds value to the corpus by inserting linguistic information into the corpus which aids information retrieval and extraction from the corpus.

As described in section 3.4, probabilistic NLP techniques must be trained by annotating a corpus or corpora either completely manually or semi-automatically using the forward-backward algorithm. Such probabilistic algorithms have achieved high success rates when applied to part-of-speech tagging (see below). However, the amount of data needed to train more accurate models increases exponentially. Therefore, as a complement, and occasionally as an alternative, to probabilistic methods, UCREL increasingly employs *template analysis* techniques [Fligelstone 96, Fligelstone 97] to reduce errors and/or ambiguity. Some examples of the template approach can be seen below.

The analysis undertaken at Lancaster can be divided into six levels, each of which builds on the level preceding it, and in some cases acts as a corrective to the analysis at the preceding level.

1. Prosodic annotation
2. Morphological analysis
3. Grammatical tagging
4. Syntactic annotation
5. Lexical semantic annotation (word-sense)
6. Discourse annotation

For spoken corpora, *prosodic annotation* is applied after the digitisation of the waveform from recorded speech and includes phonemic transcription of the data [Knowles 96]. This data is useful in speech recognition systems.

Morphological analysis can be carried out on a written corpus or on a spoken corpus once it has been transcribed. It deals with the internal structure of words and enables the separation of the base form of a word from any inflections. A program (LEMMINGS) employs an automatic process to derive the ‘stem’ (equivalent to the head word in a dictionary) of each word in running text. This automatic process is sometimes called *lemmatisation* [Beale 87].

Grammatical tagging or part-of-speech (POS) analysis is the main focus of UCREL’s work. Here, a label or tag is attached to each word in the corpus to show which grammatical class it belongs to. Both morphological and grammatical analysis are carried out within the CLAWS4 program [Garside 97] which has been developed continually since the early 1980s. CLAWS4 is a hybrid tagger using a statistical HMM technique and a rule-based component. In a fully automatic procedure, CLAWS4 assigns POS tags with 97-98% accuracy. Other POS taggers using various tagging methods quote similar success rates, such as the rule-based taggers Brill’s [Brill 92] and ENGCG [Karlsson 95], memory-based learning taggers [Daelemans 98] and the statistical Xerox tagger [Cutting 92]. CLAWS4's advantage is that it is a robust tool having been trained and tested over a large amount of data, most recently the one hundred million words of the British National Corpus (BNC) [Leech 94]. Figure 8 shows an example of the CLAWS4 c7 tagset. The first letter of each tag shows the major word class: A for article, D for determiner, I for preposition, J for adjective, M for number, N for noun, P for pronoun, R for adverb, and V for verb. TO is a special tag for the infinitive marker, XX for ‘not’ and punctuation is tagged as itself.

The_AT **lovers**_NN2 **,**_, **whose**_DDQGE **chief**_JJ **scene**_NN1 **was**_VBDZ **cut**_VVN **at**_II **the**_AT **last**_MD **moment**_NN1 **,**_, **had**_VHD **comparatively**_RR **little**_DA1 **to**_TO **sing**_VVI **.**_.

Figure 8. An example of CLAWS4 POS tagging

Part-of-speech tagging is often seen as the first stage of a more comprehensive *syntactic annotation*, which assigns a phrase marker, or labelled bracketing, to each sentence of the corpus, in the manner of a phrase structure grammar. The resulting parsed corpora are known, for obvious reasons, as ‘treebanks’. Currently, UCREL employs a technique known as skeleton parsing. This simplified grammatical analysis uses a small set of grammatical categories. Texts are parsed by hand using a program called EPICS [Garside 93]. Figure 9 shows a small section from a parsed corpus. The bracket labels shown are Fr for relative clause, J for adjective phrase, N for noun phrase, P for prepositional phrase, S for sentence, and V for verb phrase. The manual production of large treebanks allows training and testing of automatic parsing tools [Church 88].

[S[N **Nemo**_NP1 **,**_, [N **the**_AT **killer**_NN1 **whale**_NN1 N] **,**_, [Fr[N **who**_PNQS N][V **’d**_VHD **grown**_VVN [J **too**_RG **big**_JJ [P **for**_IF [N **his**_APP$ **pool**_NN1 [P **on**_II [N **Clacton**_NP1 **Pier**_NNL1 N]P]N]P]J]V]Fr]N] **,**_, [V **has**_VHZ **arrived**_VVN **safely**_RR [P **at**_II [N **his**_APP$ **new**_JJ **home**_NN1 [P **in**_II [N **Windsor**_NP1 [**safari**_NN1 **park**_NNL1]N]P]N]P]V] **.**_. S]

Figure 9. An example of skeleton parsing

Beyond grammatical annotations, *semantic annotation* is an obvious next step. For example, semantic word tagging can be designed with the limited (though ambitious enough) goal of distinguishing the lexicographic senses of the same word: a procedure also known as ‘sense resolution’.

The UCREL word sense tagging system [Rayson 96] accepts as input text which has been tagged for part of speech using CLAWS4 POS tagging. The tagged text is fed into the main semantic analysis program (SEMTAG), which assigns semantic tags representing the general sense field of words from a lexicon of single words and an idiom list of multi-word combinations (e.g. ‘as a rule’), which are updated as new texts are analysed. Currently, the lexicon contains nearly 37,000 words and the idiom list contains over 16,000 multi-word units. (Items not contained in the lexicon or idiom list are assigned a special tag, Z99.)

The tags for each entry in the lexicon and idiom list are arranged in general rank frequency order for the language. The text is manually pre-scanned to determine which semantic domains are dominant; the codes for these major domains are promoted to maximum frequency in the tag lists for each word where present. This combination of general frequency data and promotion by domain, together with heuristics for identifying auxiliary verbs, considerably reduces mistagging of ambiguous words. (Future work will attempt to develop more sophisticated probabilistic methods for disambiguation.) After automatic tag assignment has been carried out, manual postediting takes place, if desired, to ensure that each word and idiom carries the correct semantic classification. A program using template analysis techniques then marks key lexical relations (e.g. negation, modifier + adjective, and adjective + noun combinations). Figure 10 is an example of semantic word tagging, taken from a library system requirements definition document.

It_Z8 **is**_Z5 **anticipated**_X2.6+ **that**_Z5 **the**_Z5 **system**_X4.2 **will**_T1.1.3 **be**_Z5 **administered**_A9- **by**_Z5 **the**_Z5 **Library**_Q4.1/H1 **,**_PUNC **but**_Z5 **this**_Z8 **will**_T1.1.3 **not**_Z6 **always**_N6+++ **be**_A5.2+[i9.3.1 **the**_A5.2+[i9.3.2 **case**_A5.2+[i9.3.3 ._PUNC

Figure 10. An example of lexical semantic tagging

The semantic tags are composed of:

- an upper case letter indicating general discourse field.
- a digit indicating a first subdivision of the field.
- (optionally) a decimal point followed by a further digit to indicate a finer subdivision.
- (optionally) one or more ‘pluses’ or ‘minuses’ to indicate a positive or negative position on a semantic scale.
- (optionally) a slash followed by a second tag to indicate clear double membership of categories.
- (optionally) a left square bracket followed by ‘i’ to indicate a semantic idiom (multi-word unit).

For example, A5.2+ indicates a word in the category ‘general and abstract words’ (A), the subcategory ‘evaluation’ (A5), the sub-subcategory ‘true and false’ (A5.2), and ‘true’ as opposed to ‘false’ (A5.2+). Likewise, Q4.1/H1 belongs to the category ‘communication’ (Q), subcategory ‘the media’ (Q4), and refers to ‘books’ (Q4.1), as well as ‘kinds of houses and buildings’ (H1).

The semantic annotation is designed to apply to open-class or ‘content’ words. Words belonging to closed classes (such as prepositions, conjunctions, and pronouns), as well as proper nouns, are marked by a tag with an initial Z, and usually set aside from any statistical analysis.

The UCREL *anaphoric (discourse) annotation* scheme [Fligelstone 92] co-indexes pronouns and noun phrases within the broad framework of cohesion such as is described by [Halliday 76]. As with syntactic annotation, it is a manual annotation process which aims to build linguistic resources as training material for future automatic procedures. The semantic tagging system described above includes a template analysis module aimed at automatically resolving the antecedents of 3rd person pronouns (it, she, he, him, her, they, and them). This was partly trained on the anaphoric treebank, a section of which appears in Figure 11. The numbered items show words or phrases linked by anaphoric reference.

```
(0) The state Supreme Court has refused to release {1 [2 Rahway State Prison 2] inmate 1}}
(1 James Scott 1) on bail .
(1 The fighter 1) is serving 30-40 years for a 1975 armed robbery conviction .
(1 Scott 1) had asked for freedom while <1 he waits for an appeal decision .
Meanwhile , [3 <1 his promoter 3] , {{3 Murad Muhammed 3} , said Wednesday <3 he
netted only $15,250 for (4 [1 Scott 1] 's nationally televised light heavyweight fight against
{5 ranking contender 5}} (5 Yaqui Lopez 5) last Saturday 4) .
(4 The fight , in which [1 Scott 1] won a unanimous decision over (5 Lopez 5) 4) , grossed
$135,000 for [6 [3 Muhammed 3] 's firm 6], {{6 Triangle Productions of Newark 6} , <3 he
said .
```

Figure 11. The anaphoric annotation scheme

5. Information retrieval and statistical analysis

Given that probabilistic NLP techniques can classify words and word sequences, the next concern is how this information can be used to reveal interesting properties of a body of text. As a first step in explaining this, this section summarises the various information retrieval and extraction techniques that can be applied to a corpus once it has been annotated.

Word frequency profiling is a standard operation that can be applied to a corpus. The frequency list records the number of times each word occurs in the text. It can be arranged in order of first occurrence, alphabetically or in frequency order. First occurrence order serves as a quick guide to the distribution of words in a text, an alphabetic listing is built mainly for reference, but a frequency listing can provide interesting information about the words which appear (or do not appear) in a text. For example, Juilland produced a series of frequency dictionaries [Juilland 64, Juilland 65, Juilland 70]. More traditional dictionaries can result from the frequency dictionaries. The American Heritage Word Frequency Book [Carroll 71] formed the citation base for the American Heritage School Dictionary. [Francis 82] takes the simple word frequency list one stage further by reporting *grammatical word* frequencies. This gives frequencies of words with their associated POS tags in the Brown corpus. The frequency profile for a given text can be compared to that of other similar texts or to that of large bodies of text. Significant changes to the ordering of the words in the frequency list can flag items of interest to the researcher since the high frequency items tend to have a stable distribution [Sinclair 91]. Such techniques can be carried out manually for a small corpus but otherwise we need the aid of a computer program. Although the computer saves us time with its processing of the texts into frequency lists, it presents us with so much information that we need a filtering mechanism to pick out significant items before the analysis can proceed. [Hofland 82] use Yule’s K statistic and the chi-squared goodness-of-fit test in their comparison to pick out statistically significant different word frequencies across British and American English. Various formulae can be applied to adjust the raw frequencies for the distribution of words within a text, or to describe the dispersion of frequencies in subsections of a corpus.

Recently, with so much work being done on the analysis of corpora, it is seen as essential to annotate a corpus with the results of the research. Obviously, this can act as a bootstrap for an increasingly detailed and accurate analysis at the same linguistic level or for the next level of research. As shown in the previous section, we can build a hierarchy of analyses from POS tagging, parsing, semantic tagging to discourse analysis. This enables us to revisit the results obtained from word frequency analysis and obtain frequency profiles for POS tags, semantic categories and so on. By applying the same significance testing methods we can extend and refine our analysis based on more precise linguistic categories. We can perform a statistical comparison of annotated corpora and obtain results at each level of annotation contained within the corpus.

Another type of corpus comparison views one of the corpora as a *normative corpus*. In this analysis the normative corpus should be representative of general English or a sublanguage of English that we are interested in studying. A sublanguage has been defined as the semantically highly restricted subset of language used by a particular group of people who share a common interest or are employed in a specialised occupation [Bross 72]. More practically, it can be seen as the language used in a body of texts dealing with a particular subject area in which certain vocabulary and grammar are used in common [Hirschman 82]. We can build such a representative corpus and use it in a comparison with a smaller text to highlight the unusual content, vocabulary and grammar which distinguishes that text from more general usage.

XMATRIX is a tool developed within UCREL to perform this kind of analysis. It is able to produce frequency profiles for annotated corpora at the word, POS and semantic level. XMATRIX allows us to carry out the comparison of frequency profiles across two or more corpora to determine statistically significant differences at the word, POS or semantic level. It has been used for the study of content analysis (statistical analysis of primarily the semantic features of text) [Thomas 96], social differentiation in the use of English vocabulary [Rayson 97] and automatic profiling of learner English [Granger 98].

6. Tagging techniques for Requirements recovery

Early natural language processing techniques applied to retrieval of information [Salton 83] were disappointing due to lack of robustness and applicability across domains. As observed in section 2, attempts to adapt such rule-based NLP techniques to the analysis of requirements documents are hamstrung by these limitations. We believe that these attempts have been too ambitious. Instead of attempting to automatically process documents by, for example, generating system models from NL requirements, greater leverage can be obtained by using corpus linguistic techniques to reveal potentially interesting properties of requirements information to a human analyst. We characterise this as providing tools to give analysts a 'quick way in' to large, complex and poorly structured documents. These should help analysts impose a structure on the information, to provide selective views on the text and to point to clues to crucial information about actors, roles, capabilities and constraints within the system context. In developing this theory, we have extrapolated from our knowledge of the practical problems posed by 'legacy' document to analysts and from the track-record of probabilistic NLP techniques in other domains. For example, semantic annotation has already been shown to improve retrieval in general American English [Gonzalo 98].

We propose to use the existing UCREL tools and apply them to all classes of legacy documents; requirement documents, operating procedures and any other documents that describe, or otherwise provide context for, system, software or stakeholder requirements. We will evaluate the use of tools for lemmatisation, POS tagging, semantic tagging and anaphoric reference on the specific document genres.

In order to perform natural language analysis on the legacy documents, we need to obtain a corpus from which we can build a language model. Logically, the best corpus for this purpose would be a large collection of requirement specifications, operating manuals and so on. Unfortunately these are not easy to acquire for reasons of commercial confidence. To compensate for this restriction, we are building a corpus containing what we anticipate to be similar vocabulary distribution from various sources:

1. requirements documents and operating manuals to which we have access
2. IBM manuals corpus (800K words) [Black 93]
3. Subcorpus of the BNC (extracted from the pure and applied science section which is 11 million words)
4. CSEG technical reports (from the Computing Department at Lancaster University)
5. Transcripts of ethnographic studies of technical workplaces (e.g. Air Traffic Control)
6. Public domain IT standards documents

The purpose of this corpus is twofold. It will act as a baseline (or normative) corpus in our statistical profiling analysis. This corpus will also be used as a testbed for retraining the existing UCREL tools. We hope to determine whether the grammar used in requirements documents is similar to that in the IBM manuals corpus. The IBM corpus featured more uses of the imperative verb form, had a lower than average number of types (unique grammatical words) and showed a shorter than average sentence length [McEnery 96]. The corpus may be used to retrain the probability matrix within CLAWS4, and update the associated linguistic resources such as the lexicon and idiom list with new vocabulary particular to these genres. Items 1 and 5 will be specifically used for this purpose as they are more domain specific. Once the corpus has been collected, we will need to balance the quantity of each type of document so that the corpus as a whole is more representative of the domain we are modelling. This may mean, for example, removing or sampling from the large domain specific texts we obtain.

As mentioned in section 5, XMATRIX is used to perform frequency profiling. At the most basic, this allows us to produce a simple concordance of individual words in context. However, POS and semantic tagging allows more useful concordances to be formed. For example, XMATRIX can show the frequency of occurrence of different parts of speech. An obvious example of how this can benefit the identification of requirements is the extraction of all the occurrences of modal verbs (‘shall’, ‘must’, ‘will’, ‘should’, etc.). Expressions of need, desire, etc., consistent with user or system requirements can therefore be located in a document very easily and without the need to construct complex regular expressions or search templates. Even this basic level of analysis goes beyond what the current generation of commercial requirements and document management tools allow.

7. Summary of preliminary results

The first part of a normative corpus for REVERE has been built. We have selected 135 files from the pure and applied science section of the BNC which were related to Information Technology (IT). Collectively, the files form a corpus of 1.7 million words, of which about 60% are news stories relating to IT. We expect the building and tuning of the corpus to be an on-going task and to tag new documents as we identify them. For example, we are currently acquiring a number of IEEE and ISO/IEC standards to tag.

In parallel with corpus annotation, we have ported some of the UCREL tools from UNIX to Linux and NT to make them more widely usable. An implication of this work is that some evolution of the XMATRIX is likely be necessary as we evaluate it and as requirements for the recovery of requirements from legacy documents are refined during the course of our experiments.

Comparison with the normative corpus allows information to be extracted from a document by searching for statistically significant deviations from the frequency norm suggested by the corpus. One of our first experiments has been with the user requirements definition of a library information system. When we sorted the semantically tagged text by deviation from the norm, among the most over-represented semantic categories (under-represented categories can also be interesting) were those shown in Table 1.

Semantic tag	Semantic category	Example items	LIBSYS relative frequency	Log-likelihood	BNC IT relative frequency
Q1.2	Paper documents and writing	documents, records, prints	4.74	717.5	1.02
T1.1.3	Time future	will, shall	3.70	483.8	0.91
A1.5.1	Using	user, end-user	2.64	260.1	0.81
I2.1	Business	agents, commercial	0.12	208.8	1.44
S7.1+	Power, organising	administrator, management, order	2.31	159.5	0.89

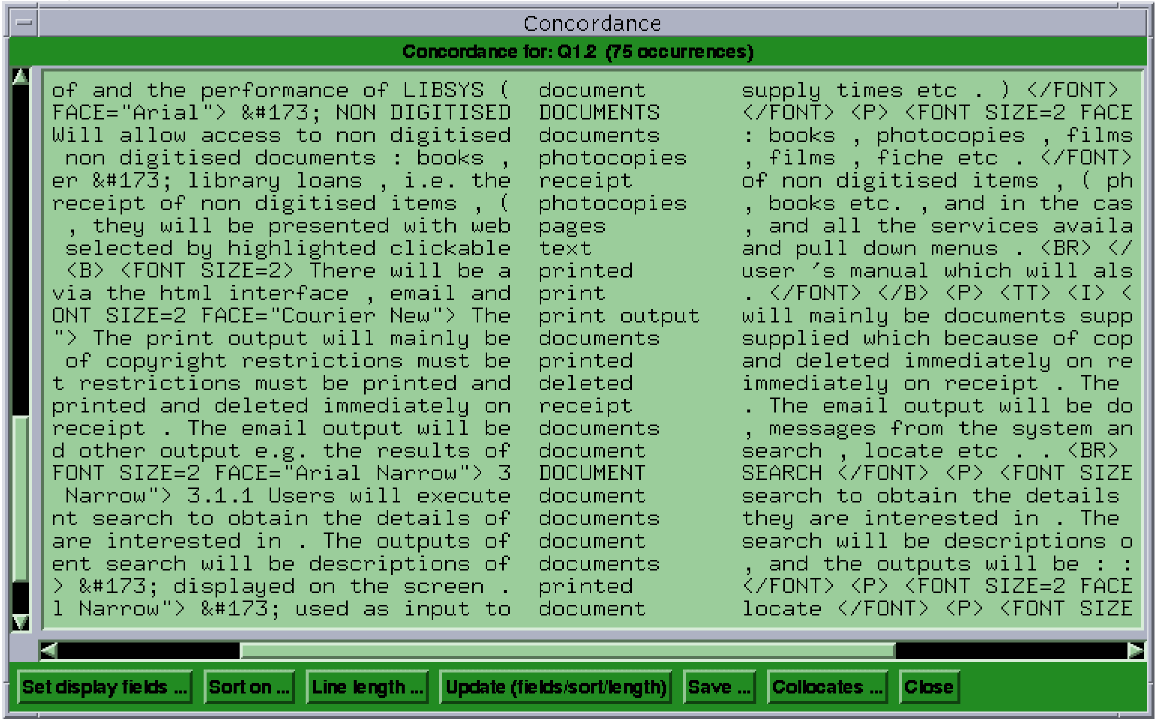
Q4.1	The media	author, catalogues, librarian	0.98	144.6	0.22
X9.1+	Ability, intelligence	be-able-to	0.75	129.9	0.14
X2.4	Investigate	search	0.04	119.0	0.74

Table 1 Semantic tag comparison of LIBSYS and BNC IT corpus

The figures in the 5th column represent the semantic categories' log-likelihood (LL) figure. This statistical significance figure shows the degree of deviation from the norm. An LL greater than 6.63 indicates that there is a 99% confidence in the result's accuracy. Hence, the above 8 semantic categories stand out as highly unusual deviations from normal English usage which suggest that it would be rewarding to investigate the usage of the corresponding words or idioms within the document.

In this case, for example, 'author', 'catalogue' and 'search' emerge as candidate roles, objects and tasks respectively. To further investigate these, and validate whether these are really candidates for components of a conceptual model, XMATRIX can be used to display a concordance for each semantic category (Figure 12). This shows words and idioms of the semantic category *paper and document writing* (tag Q1.2) in the context in which the words appear in the document.

From this, the analyst can infer that significant objects within the library application include 'document', 'web page' and 'photocopy'. Similarly, 'print' and 'receive' are candidates for operations on the objects.



Concordance for: Q1.2 (75 occurrences)

```
of and the performance of LIBSYS (   document        supply times etc . ) </FONT>
FACE="Arial"> &#173; NON DIGITISED   DOCUMENTS       </FONT> <P> <FONT SIZE=2 FACE
Will allow access to non digitised   documents       : books , photocopies , films
 non digitised documents : books ,   photocopies     , films , fiche etc . </FONT>
er &#173; library loans , i.e. the   receipt         of non digitised items , ( ph
receipt of non digitised items , (   photocopies     , books etc. , and in the cas
 , they will be presented with web   pages           , and all the services availa
 selected by highlighted clickable   text            and pull down menus . <BR> </
 <B> <FONT SIZE=2> There will be a   printed         user 's manual which will als
via the html interface , email and   print           . </FONT> </B> <P> <TT> <I> <
ONT SIZE=2 FACE="Courier New"> The   print output    will mainly be documents supp
"> The print output will mainly be   documents       supplied which because of cop
 of copyright restrictions must be   printed         and deleted immediately on re
t restrictions must be printed and   deleted         immediately on receipt . The
printed and deleted immediately on   receipt         . The email output will be do
receipt . The email output will be   documents       , messages from the system an
d other output e.g. the results of   document        search , locate etc . . <BR>
FONT SIZE=2 FACE="Arial Narrow"> 3   DOCUMENT        SEARCH </FONT> <P> <FONT SIZE
 Narrow"> 3.1.1 Users will execute   document        search to obtain the details
nt search to obtain the details of   documents       they are interested in . The
are interested in . The outputs of   document        search will be descriptions o
ent search will be descriptions of   documents       , and the outputs will be : :
> &#173; displayed on the screen .   printed         </FONT> <P> <FONT SIZE=2 FACE
l Narrow"> &#173; used as input to   document        locate </FONT> <P> <FONT SIZE
```

Figure 12. Concordance of semantic tag Q1.2

In addition, just as simple frequency profiling on parts of speech adds value to information retrieval when compared to using regular expressions, so using deviation from the norm of semantic categories adds further value. In our example, the multi-word combination 'be able to' occurred in one of the requirements without an accompanying modal verb. This requirement was not retrievable using POS tagging only. This suggests that NLP tools can also help detect inconsistencies and help accommodate quality problems in the documentation.

It should be noted that the experiment described above tends to flatter our approach because the document it applied to conforms to good requirements definition practice, is relatively small (34 pages of text) and we were familiar with it. Information is unlikely to emerge so easily from less well-documented or poorly maintained requirements documents. Nor will relevant information be so easily extracted from legacy documents of, for example, business procedures because such documents are unlikely to embody information about the role, scope or context of the legacy software. However, the experiment demonstrates the tools' potential. This will be verified and during the next stage of the project when we plan to investigate tuning the tools for requirements recovery and how they can be integrated in an analyst's toolkit.

8. Future work

Our next task is to devise a large-scale experiment with our project partners and refine the semantic categories and lexicon of words and idioms (originally defined by professional linguists for general English). At present, the lexicon contains a number of categorisations that, in an IT context, appear anomalous. For example, while 'browse' would be tagged with the semantic category *investigate* along with 'search' and 'look for', 'query' would be tagged as a *speech act*. We expect that it will be necessary to refine the lexicon to derive an IT-oriented lexicon and semantic classification.

Our expectation is this would underpin a core toolkit that can be tailored for particular application domains. Hence, for example, an analyst would be able to add new semantic categories for banking or railway signalling. A capability already exists to refine the lexicon by reclassifying words or idioms. This will have to be extended with the additional abilities to import and classify new words or idioms, and to define new semantic categories.

To further extend the utility of the NLP tools, we plan to develop a framework of guidance for constructing scenarios and conceptual models (possibly with some automated support [Burg 95]) from the information. The NLP tools will be integrated with a tool (JPREview) that implements the PREview method [Sommerville 98, Sommerville 99] for modelling systems and processes using viewpoints. This will support an iterative investigative process where the analyst posits a set of stakeholder types and iteratively refines this set, confirming or discounting the posited viewpoints and extending the set as new ones are inferred by the text analysis (Figure 13).

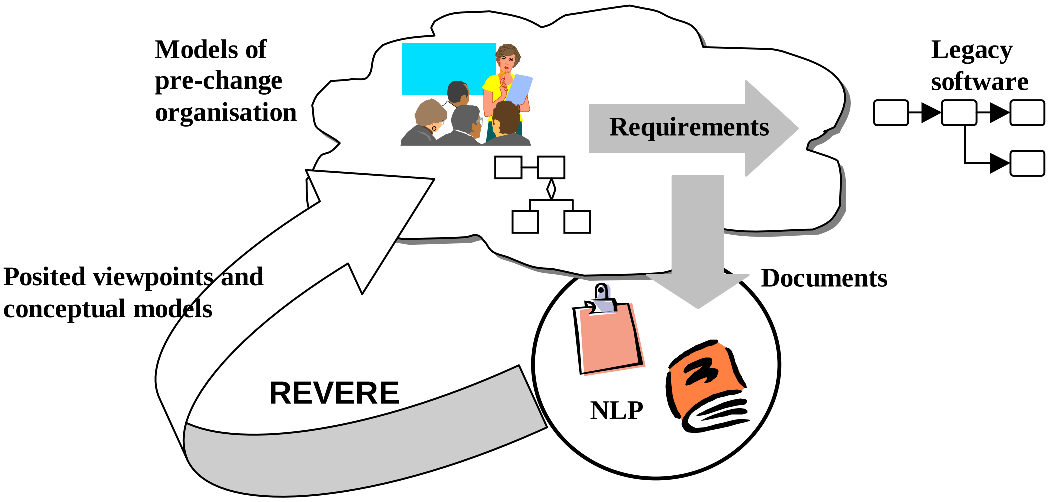

Models of pre-change organisation
Legacy software
Requirements
Documents
Posited viewpoints and conceptual models
REVERE
NLP

Figure 13. Iterative NLP - informed modelling

9. Conclusions

This document is about the application of *language engineering* to legacy systems. It has described the preliminary results of the REVERE project investigating support for the recovery of legacy software requirements from documents. The motivation for this is that:

- Decisions about how to adapt legacy software must be properly informed by an understanding of what is redundant, what must be retained and what can be reused. A precondition for this is that the requirements that motivated the legacy software before the organisational change occurred must be understood.

- Documents often form an important, and sometimes the primary, source of information about the legacy software and the pre-change organisation. These documents are often large and of variable quality so extracting information from them can be costly. They are usually written in natural language.

We plan to integrate a number of techniques to provide a set of tools to help analysts explore the documentation, and reconstruct models of the business that motivated the software. At the core of this toolset are probabilistic NLP tools to provide a ’quick way in’ to large, complex and imperfectly structured documents. At present, little effective support is available for the analysis of such documents. Probabilistic NLP offers the potential to save much painstaking and error-prone manual effort. A crucial requirement of our work is that it must scale in a way that the manual analysis of documents does not. The probabilistic NLP tools that we have chosen have been proven in other domains to do so. They are also mature and can tolerate variation in the use of language contained in documents. Hence, in contrast to other attempts at applying NLP techniques to the analysis of technical documents, they are not restricted to a well-defined subset of English. The main part of this document is concerned with a review of NLP and of probabilistic NLP techniques in particular.

The results of an initial experiment on an English language requirements definition document for a library system are presented. This experiment is used to illustrate the use of frequency profiling, part-of-speech tagging and semantic tagging to reveal interesting properties of the text. For example, certain semantic categories of words appear in the document with a frequency that significantly differs from a norm suggested by a corpus of English text. Examining occurrences of words in these categories reveals words that the analyst can infer to represent objects, roles and tasks.

We extrapolate from this to suggest that useful results may also be possible from less structured documents and documents not specifically about the software (e.g. documented business procedures). Confirmation of this must await a forthcoming large-scale experiment but initial results are promising.

We should also note that probabilistic NLP techniques may have wider potential for systems, requirements and software engineering. The potential exists for such tools to assist requirements validation by, for example, checking for document quality and consistency. Similarly, interview transcripts and ethnographic study field reports used in requirements elicitation or system evaluation currently pose severe problems for analysts who have to process them and extract key information. Probabilistic NLP techniques have the potential to assist these tasks by the imposition of structure and the abstraction of interesting features.

10. References

[Alderson 99] Alderson, A., Shah, H.: Viewpoints on Legacy systems, *Communications of the ACM*. (in press)

[Beale 87] Beale, A. D.: Towards a Distributional Lexicon, In: R. Garside, G. Leech and G. Sampson (eds.), *The Computational Analysis of English: A Corpus-based Approach*. Longman, 1987, pp. 149 - 162.

[Black 93] Black, E., Garside, R., Leech, G. (eds.).: *Statistically-driven computer grammars of English: The IBM/Lancaster approach*. Amsterdam, Rodopi, 1993.

[Blythin 97] Blythin, S., Rouncefield, M., Hughes, J.: Never Mind The Ethno Stuff - What Does All This Mean and What Do We Do Now?, *ACM Interactions*, 4 (3), 1997.

[Brill 92] Brill, E.: A simple rule-based part-of-speech tagger, *Proc. Third Conference on Applied Natural Language Processing*, Trento, Italy, 1992.

[Bross 72] Bross, I., Shapiro, P., Anderson, B.: How information is carried in scientific sub-languages, *Science*, 176, 1972, pp. 1303 - 1307.

[Burg 95] Burg, J., van de Riet, R.: COLOR-X: Object Modeling profits from Linguistics, *Proc. Second International Conference on Building and Sharing of Very Large-Scale Knowledge Bases (KB&KS’95)*, Enschede, The Netherlands, 1995.

[Butler 99] Butler, K., Esposito, C., Hebron, R.: Connecting the Design of Software to the Design of Work, *Communications of the ACM*. 42 (1), 1999.

[Carroll 71] Carroll, J., Davies, P., and Richman, B.: *The American Heritage Word Frequency Book*, Houghton Mifflin, Boston, 1971.

[Church 88] Church, K.: A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text, *Proc. Second Conference on Applied Natural Language Processing*, Austin, Texas, 1988, pp. 136 – 143.

[Cutting 92] Cutting, D., Kupiec, J., Pederson, J., Sibun, P.: A practical part-of-speech tagger, *Proc. Third Conference on Applied Natural Language Processing*, Trento, Italy, 1992.

[Cyre 97] Cyre, W., Thakar, A.:Generating Validation Feedback for Automatic Interpretation of Informal Requirements, in *Formal Methods in System Design*, Kluwer, 1997.

[Daelmans 98] Daelemans, W., van den Bosch, A., Zavrel, J., Veenstra, J., Buchholz, S., Busser, B.: Rapid development of NLP modules with memory-based learning, *Proc. ELSNET in Wonderland (ELSNET98)*, Utrecht, 1998, pp. 105 – 113.

[Fligelstone 92] Fligelstone, S.: Developing a Scheme for Annotating Text to Show Anaphoric Relations, In: G. Leitner (ed.), *New Directions in Corpus Linguistics*, Mouton de Gruyter, 1992, pp. 153 – 170.

[Fligelstone 96] Fligelstone, S., Rayson, P., Smith, N.: Template analysis: bridging the gap between grammar and the lexicon, In Thomas, J., and Short, M. (eds.), *Using corpora for language research: Studies in the Honour of Geoffrey Leech*. Longman, 1996, pp. 181 – 207.

[Fligelstone 97] Fligelstone, S., Pacey, M., Rayson, P.: How to generalise the task of annotation,. In. R. Garside, G. Leech, and A. McEnery (eds.) *Corpus Annotation: Linguistic Information from Computer Text Corpora*, Longman, 1997, pp. 122 – 136.

[Francis 82] Francis, W., Kučera, H.: *Frequency Analysis of English Usage: Lexicon and Grammar*, Houghton Mifflin, 1982.

[Garside 93] Garside, R., McEnery, A.: Treebanking: The compilation of a corpus of skeleton parsed sentences, In Black, E., Garside, R., Leech, G. (eds.) *Statistically-driven computer grammars of English: The IBM/Lancaster approach*, Rodopi, 1993, pp. 17 – 35.

[Garside 96] Garside, R.: The robust tagging of unrestricted text: the BNC experience, In Thomas, J., and Short, M. (eds.) *Using corpora for language research*, Longman, 1996, pp. 167 – 180.

[Garside 97] Garside, R., Smith, N.: A hybrid grammatical tagger: CLAWS4, In Garside, R., Leech, G., and McEnery, A. (eds.) *Corpus annotation: linguistic information from computer text corpora*, Longman, 1997, pp. 102 – 121.

[Gazdar 85] Gazdar, G., Klein, E., Pullum, G., Sag, I.: *Generalised Phrase Structure Grammar*. Blackwell, 1985.

[Goldin 97] Goldin, L., Berry, D.: AbstFinder, A Prototype Natural Language Text Abstraction Finder for Use in Requirements Elicitation, *Automated Software Engineering*, 4, 1997.

[Gonzalo 98] Gonzalo, J., Verdejo, F., Chugur, I., Cigarran, J.: Indexing with WordNet synsets can improve Text Retrieval, *Proc. COLING/ACL'98 Workshop on Usage of WordNet in NLP systems*, Montreal, 1998.

[Granger 98] Granger, S., Rayson, P.: Automatic profiling of learner texts, In S. Granger (ed.) *Learner English on Computer*. Longman, 1998, pp. 119 – 131.

[Halliday 76] Halliday, M. Hasan, R.: *Cohesion in English*, Longman, 1976.

[Hirschman 82] Hirschman, L. Sager. N.: Automatic information formatting of a medical sublanguage, In Kittredge, R. and Lehrberger, J. (eds.) *Sublanguage: studies of language in restricted semantic domains*, Walter de Gruyter, 1982, pp. 27 – 80.

[Hofland 82] Hofland, K. Johansson, S.: *Word frequencies in British and American English*, Longman, 1982.

[Ishihara 93] Ishihara, Y., Seki, H., Kasami, T.: A Translation Method from Natural Language Specifications into Formal Specifications Using Contextual Dependencies, *Proc. IEEE International Symposium on Requirements Engineering*, San Diego, January 1993.

[Jelinek 90] Jelinek, F.: Self-organised language modeling for speech recognition, In Waibel, A, and Lee, K-F. (eds.) *Readings in speech recognition*, Morgan Kaufman, 1990, pp. 450 – 506.

[Jones 97] Jones, R.: Creating and using a corpus of spoken German, In Wichmann, A., Fligelstone, S., McEnery, T. and Knowles, G. (eds.) *Teaching and language corpora*. Longman, 1997, pp. 146 – 156.

[Juilland 64] Juilland, A.,Chang-Rodriguez, E.: *Frequency dictionary of Spanish words*, Mouton & Co., 1964.

[Juilland 65] Juilland, A. et al.: *Frequency dictionary of Rumanian words*,. Mouton & Co., 1965.

[Juilland 70] Juilland, A., Brodin, D., Davidovitch, C.: *Frequency dictionary of French words*, Mouton & Co., 1970.

[Karlsson 95] Karlsson, R., Voutilainen, A., Heikkila, J., Anttila, A. (eds.): *Constraint Grammar, a language-independent system for parsing unrestricted text*, Mouton de Gruyter, 1995.

[Knowles 96] Knowles, G., Williams, B., and Taylor, L. (eds.) *A Corpus of Formal British English Speech: The Lancaster/IBM Spoken English Corpus*. Longman, 1996.

[Lam 98] Lam, W., Loomes, M.: Requirements Evolution in the Midst of Environmental Change, *Proc. Second Euromicro Conference on Software Maintenance and Reengineering, Florence*, 1998.

[Leech 94] Leech, G., Garside, R., Bryant, M.: CLAWS4: The tagging of the British National Corpus, *Proc. 15th International Conference on Computational Linguistics (COLING94),* Kyoto, Japan, 1994, pp. 622 – 628.

[Leech 97] Leech, G.: Introducing corpus annotation, In Garside, R., Leech, G., and McEnery, A. (eds.) *Corpus annotation: linguistic information from computer text corpora*. Longman, 1997, pp. 1 – 18.

[Marshall 83] Marshall, I.: Choice of grammatical word-class without global syntactic analysis: tagging words in the LOB corpus, In *Computers and the Humanities*, **17**, 1983, pp. 139 – 150.

[McEnery 96] McEnery, A., Wilson, A.: *Corpus Linguistics*, Edinburgh University Press, 1996.

[Melchisedech 98] Melchisedech, R.: Investigation of Requirements Documents Written in Natural Language, *Requirements Engineering*, 3 (2), 1998.

[Pullum 82] Pullum, G. Gazdar. G.: Natural Languages and Context-Free Languages, In *Linguistics & Philosophy*, **4**, pp. 471 – 504, 1982.

[Rayson 96] Rayson, P., Wilson, A.: The ACAMRIT semantic tagging system: progress report, *Proc. Language Engineering for Document Analysis and Recognition (LEDAR),* Brighton, UK, 1996, pp. 13 – 20.

[Rayson 97] Rayson, P., Leech, G., Hodges, M.: Social differentiation in the use of English vocabulary: some analyses of the conversational component of the British National Corpus,. *Int. J. of Corpus Linguistics*, **2** (1), 1997, pp. 133 – 152.

[Rolland 92] Rolland, C., Proix, C.: A Natural Language Approach for Requirements Engineering, *Lecture Notes in Computer Science*, Vol. 593, 1992.

[Ryan 93] Ryan, K.: The Role of Natural Language in Requirements Engineering, *Proc. IEEE International Symposium on Requirements Engineering*, San Diego, January 1993.

[Salton 83] Salton, G., McGill, M.: *Introduction to modern information retrieval*, McGraw-Hill, 1983.

[Sinclair 91] Sinclair, J.: *Corpus, concordance, collocation*, Oxford University Press, 1991.

[Sommerville 97] Sommerville, I., Sawyer, P.: *Requirements Engineering - A Good Practice Guide*, John Wiley, 1997.

[Sommerville 98] Sommerville, I., Sawyer, P., Viller, S.: Viewpoints for Requirements Elicitation: a Practical Approach, *Proc. Third IEEE International Conference on Requirements Engineering (ICRE 98),* April 1998.

[Sommerville 99] Sommerville, I., Sawyer, P., Viller, S.: Managing Process Inconsistency using Viewpoints, *IEEE Transactions on Software Engineering* (to appear).

[Thomas 96] Thomas, J., Wilson, A.: Methodologies for studying a corpus of doctor-patient interaction, In J. Thomas and M. Short (eds.) *Using corpora for language research,* Longman, 1996, pp. 92 – 109.

[Winograd 83] Winograd, T.: *Language as a Cognitive Process, Volume 1: Syntax*, Addison-Wesley, 1983.