Information Retrieval: Still Butting Heads with Natural Language Processing ?

Alan F. Smeaton

Dublin City University, Glasnevin, Dublin 9, IRELAND
asmeaton@CompApp.dcu.ie

Abstract. Information retrieval (IR) is about finding documents which may be of relevance to a user's query, from within a corpus or collection of texts. While apparently a simple task at first glance, IR is in fact a hard problem because of the subtleties introduced by the use of natural language in both documents and in queries. The automatic processing of natural language clearly represents significant potential for improving information retrieval tasks because of the dominance of the natural language medium on the whole IR task. Information extraction is also fundamentally about dealing with natural language albeit for a different function. It is thus of interest to the IE community to see how a related task, perhaps the most-related task, IR, has managed to use the same NLP base technology in its development so far. This is an especially valid comparison to make since IR has been the subject of research and development and has been delivering working solutions for many decades whereas IE is a more recent and emerging technology.

1 Information Retrieval and Information Extraction

1.1 Infomration Retrieval

Information Retrieval (IR) is the task of finding relevant documents from a text corpus or collection in response to a user's information need. As a discipline and as an application, IR has grown enormously in importance in the last few years. Part of this has been due to the increased availability of information in digital form and part also is due to the increased inter-connection between people via computers. As more and more information has been made available in electronic format, especially text information, the increased volume of information available to us has meant that our requirement for effective search has also grown. Since the mid-1980s we have also seen the emergence of personal computing allowing people direct access to data and information. In previous IR applications people searched for information via an intermediary or expert searcher who acted as a conduit between users and their information needs, and the information itself. The combination of an increased amount of information and the direct access by endusers to that information has meant that IR is now large-scale, common and cheap whereas it used to be specialist and expensive.

Despite the fact that IR is now a commonplace operation with many tens of millions of searches on the WWW for example run daily, the techniques available and used in mainstream IR applications are the same or similar to the technologies used by IR specialists years ago. There has been little transfer of technologies from specialist retrieval to the untrained users of today with their 1-, 2- and 3-word queries to the world wide web (WWW). An exception to this is the LiveTopics feature introduced in early 1997 by Alta Vista[1]. This is a feature whereby once a user's initial search has been run and has generated a ranking of some hundreds of web pages (documents), index terms from those ranked documents are then processed to determine the most commonly occurring and offered to the user as potential extra search terms for that search. Furthermore, index terms from throughout the document corpus which have co-occurred a significant number of times with those index terms suggested for query expansion are also listed and the whole collection of candidate extra search terms is presented graphically.

While AltaVista's LiveTopics may sound useful, and indeed it is innovative and an improvement in WWW search engines' functionality, to the naive enduser the fact that he is swamped with extra candidate terms to choose from with no guidance and no help and no expertise available, is as much a hindrance as a useful feature. The real measure in choosing terms for query expansion involves examining term frequencies and co-occurrence frequencies but the point here is that the LiveTopics feature is useful to a user only if he knows how to use it, and most endusers of IR systems, including WWW search engines, do not know what they are doing. Thus we find that contemporary information retrieval as used by the vast majority of users, is inadequate.

Within the evolution of IR there have been several significant developments over the most recent years worth mentioning. The first is that IR is becoming a basic technology underlying other applications. Whereas we will always have a requirement to search through information directly in its own right we are beginning to see IR embedded in other applications for example in agents that autonomously perform tasks that require a search component or as part of mail applications that require search through email archives. A second development is the emergence of other IR applications besides basic searching. The central procedure in an information retrieval operation is the matching operation between text and query but this basic function can be used in tasks related to information retrieval like the following:

- *categorisation* is a task wherein documents in a collection are assigned one or more static pre-defined categories from a closed set of such categories.
- *filtering* involves streaming documents against one or more user profiles, each reflecting a fixed information need from a user.
- *routing* also requires the distribution of incoming documents to groups or individuals based on content.
- *clustering* is where similar documents are grouped together for subsequent browsing or retrieval.

What all these tasks have in common is that they have at their core the matching between documents and some representation of a user's information

[1] http://www.altavista.digital.com

need, where that need is either static or dynamic [1].

The final recent development in IR worth mentioning is the application of IR tasks to other media besides text. As more and more information has become available in media other than text, e.g. image, audio, etc., we are seeing techniques for content-based retrieval being used for these media, though they have some progress to make before reaching the same level of effectiveness as obtainable with text [14].

These major developments are important to information retrieval but they are really only contemporary information retrieval approaches wrapped around new applications or scenarios. In this chapter we are concerned with more fundamental issues relevant to information retrieval.

At a very abstract level, information retrieval appears to be a simple technology. Users indicate what information they seek and the system retrieves documents about that topic. In practice it is much more complicated than that. There are so many degrees of freedom and this is not just to do with the retrieval operation. For the most part, users of an IR system do not know exactly what they are looking for. If they did know then a database or other exact-match system would suffice but as users search they are exploring the document space and their own information needs also. There is also another degree of freedom at the other end of the IR process in that even if a user did manage to come up with the ideal query which fully encapsulates their information requirements it is unlikely that a document author would have used the exact same terminology to express the same content [7] thus matching query terms against document terms must be fuzzy. Furthermore, because documents are about many things a user may be interested in only a subset of a document's content so the query-document matching must account for this.

By far the biggest issue to complicate the information retrieval operation is the fact that we are dealing with natural language, often both in documents and in user queries and it is the properties of natural language that make IR so non-trivial. Text[2] consists of word tokens from a surprisingly small lexicon or dictionary, each of which can independently and in isolation convey some meaning. The morphology of words can be changed as they are concatenated into units called sentences for which there is a grammar of allowable syntactic combinations to which sentences must conform. The grammar specifying the legitimate combinations of word tokens affects the order of tokens and their morphology but this is not always strictly enforced, in user queries, for example. Sentences are in turn concatenated to make prose which makes up documents. When documents reach a large enough size they may be structurally organised, typically into a hierarchy, to ease navigation through the document, so we can have chapters, sections, sub-sections, paragraphs etc. A document corpus can be one single, large structured document like the technical documentation for a computer system, or a collection of many independent documents as in a newspaper archive.

[2] We cover written text here but many of these features are also applicable to spoken dialogue.

Word tokens do not have unique spelling and many common word tokens are polysemous in their base form like *bar*, *table* and *pen*. Many words can also be polysemous in their declined form like "lights" which can be a plural noun or third person singular present tense of the verb "to light", or "drove" which can have the same set of word classes.

Within society there are many sub-languages, each with their own sub-grammars. For example technical documentation, electronic mail messages, weather forecasts, legal documents, newspaper articles, fault reports, all are written in different styles with slightly different grammars. Technical documentation is terse, tight prose consisting of complex phrases and complex individual sentences are needed because we are usually conveying complex information, whereas electronic mail messages are often ungrammatical, short, full of abbreviations with local dialects and slang and tend to use simple grammatical constructs. Newspaper texts would be grammatically uncomplicated and short.

Text has an intricacy and a complexity and it is filled with synonymy and ambiguity, variations in capitalisation and spelling, syntax, grammar and the use of different word forms and we do not realise this until we try to process language computationally. Given that this natural language is the medium that matters in IR and IR applications it is no surprise to find IR a difficult task.

1.2 Information Extraction

Information extraction (IE) is a task which is related to IR and is well-described elsewhere in this volume. Whereas IR retrieves texts for users in response to their queries, IE processes texts into fixed format unambiguous data, perhaps for indexing documents as a component part of an overall IR process. IE can also exist on the same level as an IR system being used to automatically extract data on some topic of interest to a user from a corpus of texts and use this structured data as input to a spreadsheet or database.

In [4] information retrieval and information extraction are compared with the following highlightss:

- IE systems are more difficult and knowledge intensive to build;
- IE systems are more tied to given domains and scenarios;
- IE systems are more computationally intensive than IR;
- IE systems generally have higher precision than IR and so are potentially more efficient to use because of the possibility of reduced user requirements to read fewer non-relevant texts;
- IE lends itself to cross-lingual operations;

The last point is less of an issue now because cross-lingual information retrieval (CLIR) is currently generating huge interest in IR research with workshops, papers and a special track in TREC-6 all devoted to CLIR.

The relationship between IE and IR is quite multi-faceted as the two operations can potentially be combined in a number of ways such as the following:

1. IR could be embedded within IE to pre-process a large document collection to a manageable subset on which the more computationally expensive IE techniques can be applied;
2. IE can bs used as a component within IR where IE document analysis is used to identify index terms for document representation. For example, named entity recognition [4] is an important part of IE and would certainly help IR;
3. IR combined with IE wherein as a user browses an information space with a combination of IR search and subsequent browsing[3], IE can be used to summarise search output into some coherent whole;

The contribution of the present chapter to this volume is to present an overview of information retrieval from a technical perspective, concentrating particularly on the role of NLP in IR. The automatic processing of natural language clearly represents significant potential for improving information retrieval tasks. As we have seen, IR is fundamentally about dealing with language, language in documents and language in user queries and we will see that many of the difficulties with IR which make it a non-trivial task are due to phenomena of natural language. Information extraction is also fundamentally about dealing with natural language albeit for a different function. It is thus of interest to the IE community to see how a related task, perhaps the most-related task, IR, has managed to use the same NLP base technology so far. This is an especially valid comparison to make since IR has been the subject of research and development and has been delivering working solutions for many decades whereas IE is a more recent and emerging technology.

In this section we have shown how IR is indeed a non-trivial task due to its many degrees of freedom. In the next section we will generalise the features of current operational IR systems, not research prototypes or ideas but operational systems which have been shown to handle of the order of hundreds or megabytes or gigabytes of text. Following that we then discuss the various ways in which we can represent documents and queries and we look at some of the techniques which can be incorporated into the retrieval or matching operation.

The main contribution of this chapter is the discussion of the role of NLP in IR and so in section 5 we give a brief overview of the levels of NLP that concern IR, namely lexical, syntactic and semantic processing and we follow that, also in section 5, with a review of NLP as actually used in operational IR. Having covered the present contribution of NLP to operational IR we then discuss the prospects for the development of the relationship between NLP and IR in section 6, which is the final section.

2 Operational IR Systems

As we have seen, information retrieval is concerned with finding information of relevance to user's information needs. Traditionally IR has found application in

[3] This is the current paradigm for the vast majority of users browsing the WWW.

bibliographic searching of libraries, patent offices and legal applications but as more and more information has become available electronically this has led to users performing their own searches directly. The trickle-down phenomenon is now complete with literally millions of untrained users madly seeking information in the largest collection of globally distributed electronic information available, the world wide web, and generally not getting what they want but accepting that.

To implement information retrieval, the general approach taken is to represent both documents and queries as bags of index terms. For simplicity these index terms can be normalised forms of words such as word stems or word base forms, or perhaps the terms may consist of linguistically or statistically motivated phrases. The "matching" criterion is to count the number of terms in common between query and document and normalise this count by some function of the document and/or query lengths, factoring in such things as passage-level evidence, relevance feedback, query expansion, and so on. Clearly this is inadequate and demands that authors and searchers use a common vocabulary, a proposal which is never palatable.

Given that IR is addressing a problem where user's information needs vary over time it is desirable to add some dynamicity into the IR process. This can be manifest as incorporating relevance feedback from the user back into the retrieval process to help retrieval by feeding back to the system which of the retrieved documents are relevant. Alternatively, or in addition, a system should allow a user to explicitly alter their initial query in mid-search by adding or deleting query terms from the working query. Another desirable feature of IR systems is that they provide ranked output of documents rather than sets. With ranked output, ranked according to estimated similarity to the query, a user decides when to stop looking for documents, i.e. the user takes on the responsibility for this; in set-based retrieval as we get with boolean searches, the system presents an unordered set of documents and it behoves the user to find relevant documents from within this unordered set.

Apart from technical issues, the main philosophical problem with operational information retrieval systems is due to the representation of documents and queries as bags of terms which yields information systems which match at the symbolic,or literal level. Implicit in this are the following assumptions

- users know what they want and can articulate it accurately;
- users' information needs stay fixed during a search;
- authors of documents know what they want to say and can also articulate it accurately;
- users know what terms are used in documents;

Clearly in view of what we have seen in the earlier part of this chapter, all of these assumptions are false and so operational information retrieval must fall short as a solution to the specification of what we really want from IR which is to satisfy user's real information needs.

The IR systems used today are based on matching bags of index terms which as we saw is inadequate, yet operational information retrieval does provide functionality and has done so for decades. The principles of document ranking etc. developed 20 years ago or more have filtered into operational IR and both theoretical and empirical research into improving this has been an active area. Sections three and four of this chapter describe just some of the approaches which have tried to represent (or index) documents and queries and to perfect the matching operation.

The evaluation and comparison of approaches has been an important feature of IR research with the widespread benchmarking of new and innovative approaches to IR components being the norm. Such evaluations have normally been carried out under scientific conditions with the use of test collections. These collections of documents, topics or queries, and relevance assessments for each of those topics, have been expensive to construct and up until recently have been small in scale. Since the early 1990s the United States Department of Defence have been part-funding a number of competitive evaluations in human language technologies. These include evaluations in speech processing, in information extraction from messages (MUC) and in information retrieval (TREC).

The annual series of TREC conferences [12, 11] have had a significant impact on experimental information retrieval in the United States and elsewhere [27]. While TREC can be credited with helping to spawn innovations in a number of IR areas such as collection merging and data fusion, its chief contribution is to have forced issues of scale on the IR community. IR experimentation is now regularly reported on collections of documents which are gigabytes in size and much of this is due to TREC. We will now look at how documents and queries can be represented and matched internally within an IR system.

3 Representing and Matching Documents and Queries

3.1 Representing Text

Information retrieval is supposed to be about matching a user's query against a set of documents but in practice it is about matching a *representation* for a user's query against the *representations* for documents. It is clearly evident that the form of representation and the way that that representation is achieved, is fundamental to any IR system.

Broadly and very simplistically speaking, language is about real world concepts and relationships between those concepts, albeit modified by some modalities or otherwise as we shall discuss later. The task we are faced with is to turn this into some representation whose combined semantic meaning is equivalent to the content of the text. The most popular representation for text in an IR system is as a set of index terms and while one could spend a long time pointing out the inadequacies of such an approach the simple fact remains that this is almost a *de facto* representation approach used in operational IR.

The identification of a set of index terms to represent texts can be done on several levels as follows:

- we can identify word-level equivalencies between words in the text to be represented and index terms in the representation. Thus we can determine that words like "vibration" and "undulation" in documents in an aeronautics domain all refer to a concept which can be represented by the term "oscillation". More simply however we can determine that each word in a document, with the exception of some non-content bearing stopwords, represents a single concept and we can represent the document by the set of non-stopwords, normalising different word forms as appropriate (see later).
- we can attempt to identify equivalencies at the conceptual level where occurrences of phrases such as "sonographic detection of fetal ureteral obstruction", "obstetric ultrasound" and "ultrasound in child gestation" all refer to the same concept of *prenatal ultrasonic diagnosis* and wherever such concepts occur in documents as represented by such phrases, we always index by the concept of *prenatal ultrasonic diagnosis*.

Ideally, concept level indexing represents documents by phrases rather than individual words and it is semantically rich but it is costly, laborious, specialised and domain-dependent, and it is manual. In between these two extremes of simplicity and complexity there is somewhat of a middle ground, though it does lean towards the more simplistic of the approaches which is unfortunate. The idea of transforming

$$\text{text word} \rightarrow \text{index term}$$

rather than

$$\text{text word} \rightarrow \text{concept} \rightarrow \text{index term}$$

is more commonplace in information retrieval and we shall now look at the most common technique used in practice, stemming, and at a number of linguistically-motivated approaches later.

The English language, like most others, modifies word forms depending on the role played by that word in the text or dialogue in question. This we normally add -S to the end of a noun to turn it from a singular form into a plural noun, we add endings like -S, -ED and -ING to verbs to conjugate them and we add endings like -LY to a noun to turn it into its adverbial form. These simple rules are compounded by a large number of exceptions and there is great overlap between the intended meanings of a large number of word form occurrences. Because of the various ways of modifying a word form in language, matching one form of a word in a user's query against all possible other forms of that same word can create a problem. In IR this is handled by normalising all word forms in both documents and queries into a single form and the most common approach to doing this is word stemming.

Word stemming is a crude pseudo-linguistic process which removes word suffices to reduce words to their word stem. In an ideal language this would always reduce a word form occurrence to a real word base form but because of the large number of exceptions to word modification rules, words are more often than not reduced to an artificial word. For example, one of the commonly used stemming algorithms is Porter's stemmer [18] for English which turns the ending -IES into the ending -I if the remaining word is long enough and also turns an ending -Y into an -I under the same conditions. This means that word occurrences such as PONY and PONIES are stemmed to PONI which is not a real word in the sense of small horses but in information retrieval it does not matter what the actual word stems mean so long as they are matched. Thus a query for PONIES will match against documents containing PONY, a document on COMPUTING will be matched against a query on COMPUTERS because both will reduce to the artificial word stem COMPUT, and so on.

Word stemming is a simple process, simple conceptually and simple to implement but despite, or perhaps because of, its simplicity it has found widespread application in IR systems. The true effectiveness of using word stemming has been shown by Harman [10] but more recent work [13] has shown that the properties of even this apparently simple aspect of an IR system have yet to be fully understood.

3.2 Retrieving Text

The function of the retrieval operation in an IR system is to compute some degree of overlap between the representation of a document and the representation of a query, for each document in the corpus, and to rank documents by this Retrieval Status Value (RSV) or score. There are some simple metrics for doing this such as Dice's coefficient, or the Cosine coefficient [29] but these heuristic methods generally do not perform well because they do not account for the relative importance of index terms in the collection.

Much work in IR has been devoted to the development of mathematical models for the IR task, in particular for retrieval. The most important of these have been based on probability theory and on vector space modelling respectively. Such models have been extended to incorporate the most desirable features of retrieval, namely query expansion and relevance feedback. There is a wealth of published literature in this area, going back 20 years, but the essential background points related to conventional retrieval are covered in [19]. This article presents many of the aspects incorporated into contemporary term weighting functions wherein an IR system assigns a weight or degree of importance to index terms in a collection depending on their collection frequency (number of documents in a collection containing a term) and within-document frequency (number of times a term occurs within a given document). The popular $tf \times IDF$ weighting function which combines the two term frequencies above, has proved to be particularly robust and is in widespread use.

Besides the common model of matching query terms against sets of document terms there are other possible aspects to the retrieval operation listed below.

1. *Cluster-based retrieval* depends on clustering documents into groups of similar documents, usually done *a priori*, and using these clusters as part of

retrieval. Alternatively a user may be presented with documents clusters as a result of a search and may be encouraged to browse through the clusters.

2. *Retrieval as a combination of several retrieval strategies* often involving data fusion approaches; this has found particular favour in the TREC experiments and involves a combination of rankings from more than one document ranking approach into one, consolidated ranking. Data fusion may also involve generating more than one version of the same query to run against a document collection and in this case the individual document rankings are combined [2]. Data fusion has consistently been shown to improve overall retrieval effectiveness when the rankings come from retrieval strategies which are conceptually independent of each other.
3. *Latent semantic indexing* is based on the statistical technique of singular value decomposition where an $m \times n$ matrix is reduced to an $m \times \delta n$ matrix where the transformation of $n \rightarrow \delta n$ is a reduction in the number of index terms for the document collection and δn is of the order of 100 to 300. This dimensionality-reduction technique allows term-term dependencies and relationships to be statistically aggregated and incorporated into the reduced term space and although very computationally expensive, especially for very large collections like TREC [6], has been shown to lead to effective information retrieval [8]
4. IR normally delivers entire documents in response to user queries and on those documents users can make relevance judgements but we have also seen the emergence of *passage retrieval*, where the retrieved items are sections within documents [20]. This is known as passage retrieval and is difficult to evaluate, which is something IR has traditionally liked to do, but appears to be worthwhile for handling long documents.
5. An aspect related to passage retrieval is the problem of applying standard IR techniques to *heterogeneous length documents*. One can simply normalise a document's RSV by its length but this pre-supposes that documents are about topics which are treated equally throughout the length of a (long) document. This is not so as long documents are compositional and the topics covered in a text are treated unevenly and in different document segments. Significant improvements in retrieval effectiveness may be obtained by incorporating document length into the document scoring procedure as shown in [23] and [19].

The above list represents just a snapshot of some of the areas of retrieval research commanding lots of attention in the IR research community and illustrates that this is still a very active area.

4 Natural Language Processing and Information Retrieval

Computational linguistics is the study of computer systems for performing automatic natural language processing and like IR, NLP and computational linguistics has had a long history of evolution. Computational linguistics aims to develop systems for processing natural language and aims to handle *most* cases of natural language. NLP systems do not mind occasional failures and are more concerned with getting systems working.

The automatic processing of natural language is often divided into a number of levels or strata which represent the levels at which language exists. NLP as a whole is well-described elsewhere in this volume and a re-description here would be inappropriate. Rather than that we choose to present a summary overview of the levels of language processing which are relevant for information retrieval, namely lexical, syntactic and semantic and we follow that with a review of how those levels of language analysis are used in IR systems. Before we proceed it is worth pointing out that this is a very short and simplified view of a large field which has been active for many decades and such a short review must surely be unable to do justice in terms of coverage. What we do present here is the view of NLP from the information retrieval perspective and all its innate biases.

4.1 Brief Overview of NLP

Lexical Level Language Processing. At a lexical level of language processing we seek to identify words and their grammatical classes. We process each word individually and out of context, we handle word morphology meaning different word endings and so on, and we generally make use of a lexicon or dictionary.

In an ideal language[4] where lexical ambiguity did not exist, we would take each word token in text and look it up in a lexicon to determine its base form and grammatical class. In English this is not possible as we have much lexical ambiguity where nouns can act as verbs, where noun plurals are created by appending -S to the word which is the same way in which the third person singular present tense of a regular verb is constructed. Thus when we encounter the words “leaves” or “covers” we do not know whether these are plural nouns or verb forms. When we also take into account the ambiguities introduced by irregular verb and other word forms we have even more ambiguity. For example, “DROVE” is both a noun and verb form.

It is impossible to resolve the many instances of lexical ambiguity by processing at only the lexical level and it requires higher levels of language analysis to do this.

Syntactic Level Language Processing. Traditionally syntax means the structure of a sentence. It means the parts-of-speech and their set of rules acting on them to determine grammaticality, or put simply, the set of rules which determines legitimate sequences of words in a language. Researchers at the syntactic level of language analysis have primarily been concerned with the construction of wide-coverage grammars and the development of efficient parsing strategies. Grammatical formalisms have also been studied in order to try to capture the vagaries of natural language and these have led to the development of phrase

[4] An ideal language for automatic analysis that is !

structure grammars, context-free grammars, context-sensitive grammars, transformational grammars, definite clause grammars, constraint grammars and many more.

Natural language has proved notoriously difficult to capture in its entirety as a set of rules as there are always exceptional sentences or clauses which make the complexity of grammars huge, hence there is no definitve "grammar for English", or any other natural language.

The aim of syntactic processing is to determine the structure of a sentence but that structure itself can be ambiguous and there is that word "ambiguous" again which causes so many of the problems in NLP. The input to the syntactic analysis process (probably) has lexical ambiguities and structural ambiguity can arise within the resultant syntactic structure itself due sometimes but not always to an underlying lexical analysis. For example, the sentence "I saw her duck" can have two structural interpretations parenthesised as either:

(((I saw) her) duck)
((I saw) (her duck))

In the first case we refer to somebody viewing somebody else's pet water fowl whereas in the second case we refer to somebody viewing somebody else who was crouching perhaps to avoid a low-flying object ! This structural ambiguity is caused by the lexical ambiguity of the word "duck" which can act as either a noun or verb form. The sentence "sheep attacks rocket" is similarly ambiguous with "attacks" being either a noun or verb. On the other hand the classic "I recognised the boy with the telescope" is genuine pure structural ambiguity without any underlying lexical ambiguity.

There are many sources of syntactic ambiguity in English but the three most common sources are caused by prepositional phrase attachment, co-ordination and conjunction and noun compounding.

– **PP Attachment:** Prepositional phrases (PP) are a linguistic feature consisting of a preposition followed by a noun phrase, which can be attached to almost any syntactic category, including itself, in order to act as a modifier. Thus we can have PPs modifying noun phrases, verb phrases, adjectival phrases, and so on. The following sentence, although somewhat contrived, has a total of 13 PPs:

  Example 1. "I broke the seal from the fuel pump with the red top to the right of the engine in the car with the dent in the back from a crash on the road to Dublin during the icy spell of weather in 1988".

  In this case, as in all PPs, each prepositional phrase is used to modify some other single construct. So, *fuel pump* is modified by the PP *with the red top*; *car* is modified by *with the dent*, and so on. The problem with using PPs is in finding out to what they should be attached as modifiers. For example, consider the following:

  "Remove the bolt with the square head"
  "Remove the bolt with the square wrench"

Both sentences are identical except for the last words, *head* and *wrench*, both of which can be either a noun or a verb, yet there is a syntactic ambiguity in that we do not know, syntactically, what the PP *with the square XXX* is used to modify. From our semantic understanding of the words in the sentences we know that bolts may have square heads but we do not normally have such things as square wrenches and that the removal operation may be completed with a square wrench but not with a square head and it is this higher level semantic processing that disambiguates the syntactic alternatives for us.

- **Co-ordination and conjunction:** Conjunction or co-ordination is one of the most frequently used constructions in natural language but the scope of the conjunctions, i.e. what is being conjoined, can almost always be ambiguous. For example, we can have conjunction among the heads of a noun phrases as in:

  "Inspect the bearing cups and cones"
  "Inspect the hub and bearing components"

  In the first case it is not clear whether we are to inspect bearing cones as well as bearing cups and in the second case it is not clear whether we are to inspect hub components. Conjunctions can appear in natural language almost anywhere, among modifiers, among PPs, among heads, among clauses, and they are used to make language more concise, but at the cost of increased ambiguity.
- **Noun compounding:** Noun or nominal compounds occur when a noun or nouns are used as a modifier of another noun, making a compound structure as in:

  "computer performance evaluation"

  In this case we have *performance*, a noun, modifies *evaluation*, another noun. *Computer*, another noun, modifies what .. *performance evaluation* or just *performance* ? We genuinely don't know from simple syntactic analysis, hence the ambiguity.

  Another feature of nominal compounding is the ambiguity caused by the kind of relationship might exist between the nouns being compounded [9]. A *fighter plane* is a plane made for fighting, a *garden party* is a party held in a garden and a *timber house* is a house made from timber.

The final problem with syntactic ambiguities is that they are potentially multiplicative rather than additive, so long and complex sentences, as are found normally in technical documentation will be likely to have much ambiguity at this level. On the other hand the advantages of syntactic level processing are that can be reasonably efficient and the rules of syntax are general and concepts like word classes are abstract, meaning that the syntactic analysis process is fairly domain dependent once the word tokens are in the lexicon.

Semantic Level Language Processing. Semantic level language analysis is concerned with context-independent meaning, taking one sentence at a time, independently of its more global context in the text or discourse. It focuses on broad questions such as what type of knowledge representation formalism to use and how to interpret things like

"John only introduced Mary to Sue"

which could actually mean any of the following:

"John did nothing else with respect to Mary"
"John introduced Mary to Sue but to no one else"
"John introduced Mary and no one else to Sue"

Generally, semantic level NLP involves defining a formal language into which natural language can be processed. The earliest attempts at understanding meaning used various forms of logic but more recently artificial intelligence represents knowledge by specifying primitive or simple concepts and then combining or structuring them in some way to define complex, real-life concepts. These can be used to capture permanent universal objects like physical objects and relationships between them but natural language discusses more than concepts and relationships between them. NL involves notions of modality (possibility, necessity), of belief and of time among others and it is necessary or at the very worst case desirable for any semantic representation of language to capture these elements of natural language. This is non-trivial and there is no universally-agreed knowledge representation formalism which does this.

Semantic level language analysis should be able to analyse grammatically parsed text into a knowledge representation format and should also be able to "parse" the semantics of the input, to note and to respond to nonsense or violations of real-world constraints or axioms. The reason for wanting to do this is that a sentence may have a number of semantic interpretations, possibly arising from a number of syntactic interpretations, and we want to eliminate as many of these as possible, especially those that would not make common sense. The sentence

"I noticed a man on the road wearing a hat"

has (at least) two syntactic interpretations with the participal phrase "wearing a hat" modifying either the man or the road. Semantic level interpretation should tell us that hats are word by animate objects such as men and donkeys and not by roads, and thus the latter interpretation should be discarded. In order to do this, however, a huge amount of domain knowledge is needed, even for restricted domain applications, and this is generally not available.

4.2 The Role of NLP in IR

So far in this chapter we have looked at information retrieval and at how it is generally implemented by representing documents as bags of terms and we have looked at NLP techniques noting the problems of ambiguity at different levels of processing. The conventional approaches to IR based on bags of terms and their statistical distributions will always have inherent limitations and possibilities for text retrieval because of the following:

1. We can have different words used to convey the same meaning as in *throttle* equals *accelerator* when referring to automobiles; a *stomach pain after eating* is a *belly-ache*, and we can have differing perspectives and thus different language used to refer to the same concept as in *the accident* vs. *the unfortunate incident* depending on whether you are for the prosecution or defence in a court case.
2. We can also have the same words used to convey different meanings. A *blind Venetian* is not a *venetian blind* and a *juvenile victim of crime* is not the same as a *victim of juvenile crime*.
3. We can also have a single word or phrase have different meaning depending on the domain; *sharp* may refer to a measure of pain intensity in medicine or the quality of a cutting tool in a gardening handbook.

Restrictions like these which intuitively set an upperbound on the processing capabilities of the contemporary approaches to IR provide a simple motivation and a justification for attempting to use NLP within IR. Large-scale applications of NLP tend to be domain-dependent and this arena of large-scale is the one in which information retrieval systems must operate. Natural language processing, excluding the simple NLP techniques used in spelling error detection and correction [3] also require much coding of knowledge bases, simple but large lexicons in the case of syntactic analysis but much more complex structures in the case of anything more sophisticated. Because this semantic-level processing requires such specialist resources it is clear that we are not going to get fully interactive, domain-independent language processing of large text bases for retrieval but the question must be asked whether we need such processing or not for IR ?

It is believed by many that the problems NLP wrestles with, especially at the semantic level, are not important for information retrieval which as we have seen already has so much vagueness and imprecision inherent. Thus IR has a great tolerance for “noise” in its processing. If a user wants to retrieve some documents about apples or about elephants then an IR system does not need to know what an apple or an elephant is, or what the difference between them might be, it just needs to find areas of its corpus which **might** be about apples or about elephants because in an IR system the decision on relevance is something that is loaded upon the user as a responsibility. In IR we do not need to comprehend or to wrestle with meaning at all, we only need to distinguish texts from each other in the context of a specific query. Perhaps these might be sub-texts and perhaps we have to generate a ranking of texts, but nonetheless information retrieval is unlike information extraction in that it does not do any more processing on documents other than deliver them to the user. Systems which do more than this are not information retrieval systems.

The lessening of ambition with respect to what IR sets out to do may be seen as a “cop out” but if this is so it is because information retrieval must deliver operational systems which scale up to large volumes and have rapid response times and it is this over-riding constraint of operation that determines this field. Given that this is so, what can NLP offer to information retrieval. The following are possibilities:

- NLP in document (and query) indexing, perhaps as a way to identify co-ordinated terms or good phrases or something else to be used as content representatives. This leads to an alternative to the "bag of words", namely the "bag of phrases".
- NLP in query formulation where NLP analysis of a user's query dialogue to support information seeking may help a user to more accurately formulate and refine a query.
- NLP in the matching operation where matching a query with a document may incorporate dynamic, on-the-fly NLP analysis of documents, perhaps involving inference on the document content from such an analysis.
- others ?

In practice it is document indexing, and by implication retrieval, which has received the most attention in applying NLP to IR and it raises the fundamental question of what should we replace the bag of words with ? Although for the remainder of this chapter we look at indexing using NLP, the retrieval operation which would have to follow normally defaults to being statistically-based retrieval as the greatest impact of NLP on IR processes has been to try to improve the quality and range of the internal representation of document and query, and the retrieval operation simply follows this.

4.3 Simple NLP as Used for Information Retrieval

Indexing by Base Forms If we process the text in documents (and queries) simply at the word level then we may try to use morphological analysis of word tokens as they occur in text to equate variations of word forms due to pluralisation and different verb forms. This would mean taking a word occurrence and determining the base form of the word from a lexical lookup and some processing of the word form. In theory this sounds attractive but there are a few significant hurdles to this becoming commonplace. In the first case, all potential words in documents and queries must be in the lexicon and building and maintaining this is likely to be expensive and incomplete. Many words in text are proper names, the names of people, places, companies, etc. and identifying these in text is difficult, in fact it is a significant part of text pre-processing for the information extraction application [4]. A second reason while indexing into base forms is not practical is that processing text at the word-only level leaves us with lexical ambiguity which as we saw earlier is only resolved with higher levels of language analysis.

The final and most significant reason why representing text by word base forms is unattractive is that for all the effort in lexical lookup and word processing, it can only be slightly better than simple, domain-independent word stemming. Word stemming is simple and it is crude and it makes mistakes, but it makes the same mistakes for stemming query words as it does for stemming document texts and the resultant incorrect word stems, even though they my not be linguistically accurate, do match against each other thus supporting retrieval. In experiments we have consistently seen that indexing by stemming and by true word base forms are approximately equal in terms of overall retrieval effectiveness [25] so overall it is not worth the computational effort compared to simple word stemming.

Indexing by Word Senses A very active area in information retrieval research in the 1990s has been the application of the recently available machine readable dictionaries (MRDs) to a variety of tasks, including text indexing for information retrieval. For each word an MRD will contain a short textual description of each valid sense of that word and so for polysemous words with more than one semantic meaning there will be more than one entry. For example, the word "ball" will have entries for the senses referring to a solid or hollow sphere, an accoutrement used in games, a missile fired from a cannon, part of the base of the foot, a type of bone joint, a social gathering or assembly, a verb meaning to squeeze material into a ball shape, and so on. It would be clearly desirable to index texts and queries accurately not by words but by the word **senses** intended.

If word sense indexing could be done accurately and with reasonable efficiency then one of the big problems with the "bag of words" approach to information retrieval, word mis-matching, would be solved ! It is thus no surprise to find that using MRDs to perform word sense indexing has attracted so much interest since the increased availability of MRDs over the last few years.

Word sense disambiguation (WSD) is a linguistic problem that has many applications in NLP besides information retrieval and as a problem in its own right attracts researchers. The most common approach to WSD is based on matching each textual sense description from the MRD against the local context or window in which a polysemous word occurs in text. Thus if some of the sense descriptions of the word "ball" include

1. a missile fired from a cannon
2. a joint between bones
3. a social gathering of people
4. a hollow sphere used in playing various sports and games
5. . . . others

and we have the following extracts of text in documents

> "The General ordered the cannons to be fired
> and the cannon balls breached the barracade"
> "The centre forward put the ball in the back of
> the net and the game was over"
> "The ladies dressed for the ball that evening"

In the first example we take the local context for the word "ball", i.e. the words surrounding it and search for occurrences of those words in the sense descriptions from the dictionary. We find the word "cannon" occurring adjacent to the word in question and that same word "cannon" appearing in a sense description. We also find the word "fired" occurring 3 words before the word being disambiguated and also in that same sense description and so deduce that the sense intended is the first one listed.

For the second sentence we find the word "game" occurring 9 words after the word being disambiguated (ball) and also occurring in sense description 4. Finally for the last of the sentences we have no words[5] immediately surrounding the word being disambiguated so we have to resort to a second-level process and look up the sense definitions of all words in the input text surrounding the word "ball" until we eventually get connections between words surrounding the word being disambiguated, and the various sense definitions on offer from the MRD.

These simple canned illustrations illustrate the mechanism behind a simple form of word sense disambiguation as a problem and illustrate the potential it has for information retrieval. If we could correctly index documents (and queries) by word senses then we would never retrieve documents about cannon projectiles when seeking information about a part of the foot nor would we receive documents about social gatherings of people when seeking information about an implement used in playing games. Unfortunately, in practice, automatic word sense disambiguation is a notoriously difficult problem and approaches based on word experts, neural networks, spreading activation and dictionary lookup have all been tried. While the accuracy of a WSD problem is dependent on the number of sense definitions in the MRD, figures of about 70% accurate recognition of the unique word sense among all words in input texts seems to be about the best.

In a series of information retrieval simulations in which he used the concept of a pseudo-word to incorporate a synthetic value for the accuracy of a word sense disambiguator, Sanderson [21] showed that a WSD accuracy of at least 90% is required before the payback of WSD begins to have a positive effect on the effectiveness of subsequent information retrieval. With performance figures less than that the amount of noise introduced by incorrect disambiguation choices brings down the relative performance of retrieval when compared to indexing and retrieval based on simple word stems. A series of subsequent experiments [22] confirmed earlier results as well as exploring a number of other WSD approaches for an information retrieval application.

The current thinking among those who work on using WSD for IR seems to be to use multiple sources of evidence in indexing by word senses by using parts-of-speech, word morphology, etc., and instead of trying to uniquely identify **the** correct sense of a word occurrence to rule out the incorrect senses and to weight the likely senses highly. This is progress in representation when compared to encoding a word occurrence as all senses equally likely which is what indexing by words or word stems, but the idea has to be explored much further.

[5] Excluding stopwords of course !

Another school of thought is not to try WSD into dictionary senses of words which may be archaic and are certainly static, but to dynamically define word senses from within a whole corpus and to index into only those word senses which exist in a corpus. At present this idea is exploratory and requires much work in corpus linguistics but it is intuitively attractive and removes the dependency on machine readable dictionaries.

4.4 Using NLP to Index by Phrases

We now turn our attention to indexing text into units which are larger and more complex than single words, stems or word senses, i.e. phrases. In information retrieval a phrase is a concatenation of two or more word forms and the set of possible phrases for a corpus is larger and richer than the set of word stems or senses. Phrases are generally more content-bearing than their individual components (words) used in isolation and so if we can successfully index text into phrases then we have a richer representation of text and hence the possibility of getting more effective retrieval.

It has been assumed by researchers in IR that in text it is the noun phrases that are the content-bearing elements and certainly noun phrases are more content-bearing than single words but phrases are not a full representation of meaning, just better content indicators than single words and in IR that is all we want.

There are two approaches to phrase identification in text termed “statistical” and “linguistic”. Statistical phrase indexing approaches are simple and crude and very common among the IR systems benchmarked in TREC [12]. Typically they pre-process a sample of text in order to determine a phrasal lexicon, effectively a list of phrases that occur with reasonable frequency in the corpus sample. Subsequent indexing of document texts is based on identifying potential phrases on the fly at indexing time and looking up such phrases in this phrasal lexicon to see if the phrase occurs with any reasonable frequency in the domain and if so then the document in question is indexed by that phrase. For example, in [15] we have used a sample text size of 20 Mbytes and if a phrase occurs more than 25 times in this sample then it is entered in the phrasal lexicon. Using these figures our phrasal lexicon is normally around 200,000 to 250,000 entries.

The technique used to identify phrases in text, both at pre-processing time to determine the phrasal lexicon and at document indexing time is normally very straightforward. Typically it involves stemming words and determining phrases to be sequences (pairs or triples) of non-stopwords that do not span sentence or paragraph boundaries, or sequences of capitalised words to identify person, place or company names (named entity recognition [4]).

Statistical phrase indexing has really crept into IR systems in recent years and the true contribution of statistical phrases has been difficult to isolate since systems reported in TREC and elsewhere normally incorporate a large number of techniques besides phrase indexing to improve retrieval effectiveness. In a recent series of experiments the effect of phrase indexing on retrieval performance has been isolated and evaluated and it has been shown that even a simple approach to statistical phrase identification is a profitable feature to incorporate into an IR system, without great computational or resource overheads [15].

NLP techniques have also been incorporated into phrase indexing yielding "linguistic phrases". Normally these are based on a syntactic analysis of document texts and such analysis can be used to determine the boundaries of noun phrases in texts and in queries. The problem with automatically indexing by noun phrases is the variety of ways of representing a concept which is so complex that it needs a noun phrase to represent it. This can lead to the kind of linguistic problems discussed earlier in the section on the role of NLP in IR and in IR we need to represent documents and queries by some normalised derivative of the occurrence form in order to address these variances. For single words the normalisation is normally done by word stemming; for phrases three approaches have been identified previously [24]

1. **Ignore:** This approach allows text to be indexed directly by phrases as they occur in documents and depends on the matching or retrieval to do something about the problems of ambiguity, different ways of expressing the same concept, etc. Generally this can only work if the phrases are limited to 2 words in length and the user is forced to enter variants of their search phrases to ensure coverage.
2. **Normalise the indexing phrases:** Here we index texts by some processed version of sets of words as they have occurred in document texts. The advantage with this approach is that it does yield a smaller vocabulary and makes subsequent retrieval less complex as syntactic variants in texts and in queries will always be normalised to the same form and the retrieval process can default to the techniques used to match single word terms.
   An example of such an approach is the CLARIT system from CLARITECH Corp. [5]. This system operates in a similar way to using statistical phrases in that a phrasal lexicon, called a first order thesaurus, is created by linguistically analysing a sample of the document corpus. This analysis is based on identifying phrases from a linguistic motivation and not just from word adjacency and frequency of co-occurrence. For example, a headnoun and its noun pre-modifier is normally a good candidate for a phrase, as is a headnoun and its modifying adjective. Once the sample text has been pre-processed to generate the phrase list, document indexing can commence by linguistically analysing document texts to identify good candidate phrases. As these are identified they are searched for in the phrasal lexicon and if they exist literally with an exact match then the phrase is used to index the document. If document phrases do not exist in the lexicon but a phrase or phrases in the lexicon are found as constituents of candidate document phrases then the document is indexed by the constituent phrases. Finally, if there are any leftovers then these require special processing. For example if we have the terms *"autonomous robot"* and *"robot navigation"* in the lexicon and we encounter a document with the text:

   > "...and the autonomous robot navigation system is designed to ..."

then the text will be indexed by the phrases *"autonomous robot"* and *"robot navigation"* even though a linguistic analysis of the document suggests the phrase *"autonomous robot navigation system"* as a good indexing phrase. The key thing about CLARIT is that it indexes documents only by phrases already existing in the phrasal lexicon and in this way it achieves phrase normalisation.

3. **Index by structures to capture linguistic ambiguities:** The third approach to handling phrasal variation in matching query and document terms is to represent either the document or query phrases as a structured representation from which all semantic interpretations can be derived and to allow the retrieval or matching operation to handle the ambiguity automatically. As reported in [24] this is attractive in theory but in practice has not yielded any fruitful developments in retrieval performance.

An alternative to identifying phrases in text for use as indexing units is taken by Strzalkowsky [28] where a linguistic analysis (parse) of document texts is performed in order to identify pairs of words which are used as indexing units. Conceptually this lies somewhere between indexing by single word forms and indexing by phrases as they occur in texts. The word pairs identified by Strzalkowsky correspond to linguistic dependencies such as a headnoun and its modifier. In indexing by word pairs this approach circumvents the problems of matching phrases as they occur in queries and in documents by indexing by word pairs which are another form of normalised simple phrases. The effectiveness of this approach in experimental evaluation in TREC has been impressive and represents another of the few examples where NLP techniques have been shown to improve document retrieval.

4.5 Using Linguistic Resources in IR

The development of natural language processing has yielded and continues to produce resources which may be of use to information retrieval. We have already seen how machine readable dictionaries can be used in indexing by word senses. The other great resource from NLP, knowledge bases, may also be of use. In terms of large-scale, domain-independent knowledge bases there are really only two possibilities at present, CYC [16] and WordNet [17].

The problem with using CYC at present is that it is still under development and is unlikely to be made available for general use. WordNet on the other hand, is freely available and has been used in IR applications to varying degrees of success, and failure. The principal task that using something like WordNet does is to address the automatic handling of related terms which is an inadequacy of keyword based IR and in the present author's own work we have used WordNet as a basis for computing word-word semantic distances as a basis for IR [26] which is ongoing research. It remains to be seen whether linguistic resources such as thesauri and knowledge bases can be exploited really effectively in information retrieval tasks, something we would expect to happen but which has not been delivered upon yet.

5 Prospects for NLP in IR

In discussing the prospects for NLP in information retrieval tasks we must define what we mean by NLP. If NLP includes such low level processes such as spelling error correction [3] then there is a lot of NLP already in IR but this is all quite straightforward stuff. As we saw, NLP exists at many different levels and if we exclude stemming and word normalisation we find that there is not too much work to report on. The reasons for this used to be because of processing costs and the poor availability of NLP techniques and resources but these reasons no longer appear to be valid. What does seem to be important is that IR and NLP are inherently different processes. IR is inexact and vague NLP has been developed for applications like machine translation and user interfaces where vagueness and imprecision are simply not tolerated. These fundamental differences in approach seem to point to an uncomfortable alliance, notwithstanding the systems mentioned earlier which are exceptions.

If we accept the evidence that current approaches to NLP and to IR are not going to lead to a groundswell in the application of NLP to information retrieval then what are the prospects for the relationship between IR and NLP. Current approaches to what we call natural language processing cannot help to progress what we currently call information retrieval and clearly the "butting of heads" which we see at present with IR attempting to cherry-pick any appropriate techniques from NLP, is not going to have any long-term impact. On the other hand, as we saw earlier, NLP and IR both deal with language and there should be some overlap in what the respective disciplines attempt to do.

It is this author's belief that what we currently call information retrieval systems, which rank documents in response to a set of user's query terms, will be enhanced by the kinds of functionality hinted at earlier in this chapter. Long documents, or even short ones, should be abstracted dynamically in the context of a given user query and these summaries presented to users; a single summary of all top-scored documents should be generated for users to browse; user's should be presented with clusters of related documents, not a single list; query formulation should be a process of dialogue and browsing the information or term space, not just a simple query input dialogue box. All of these techniques will require some elements of NLP and will deliver systems for retrieving information which go much further to satisfying a user's information seeking task than what we currently have. Such systems may use what we currently call information retrieval as a base technology underlying their operation and historical inertia may dictate this to be so but these systems will integrate techniques used in IR, information extraction and NLP. How far they go to satisfying our information seeking requirements remains to be seen but for certain they should be an improvement over what we currently call information retrieval.

References

1. Belkin, N.J. and Croft, W.B. : Information Filtering and Information retrieval: Two Sides of the Same Coin ? **Communications of the ACM**, 35(12), 29-38, 1992.

2. Belkin, N.J., Kantor, P., Fox, E.A., and Shaw, J.A. : Combining the Evidence of Multiple Query Representations for Information Retrieval. **Information Processing & Management**, 31(3), 431-448, 1995.
3. Church, K.W. and Rau, L : Commercial Applications of Natural Language Processing **Communications of the ACM**, 38(11), 71-79, 1995.
4. Cunningham, H.: Information Extraction —A User Guide. Department of Computer Science, University of Sheffield Research Memo **CS-97-02**, January 1997.
5. Evans, D.A., Milić-Frayling, Lefferts, R.G. : CLARIT TREC-4 Experiments. in [11].
6. Foltz, P.W. and Dumais, S.T. : Personalised information Delivery: An Analysis of Information Filtering Methods. **Communications of the ACM**, 35(12), 51-60, 1992.
7. Furnas, G.W., Landauer, T.K., Gomez, L.M. and Dumais, S.T. : Analysis of the Potential Performance of Keyword Information systems. **Bell Systems Technical Journal**, 62(6), 1753-1806, 1983.
8. Furnas, G.W., Deerwester, S., Dumais, S.T., Landauer, T.K., Harshman, R.A., Streeter, R.A., and Lochbaum, K.E. : Information Retrieval Using a Singular Value decomposition Model of Latent Semantic Indexing. in Proceedings of the 11th International Conference on Research and Development in Information Retrieval, Grenoble, France, ACM Press, 465-480, 1988.
9. Gay, L.S. and Croft, W.B. : Interpreting Nominal Compounds for Information Retrieval. **Information Processing & Management**, 26(1), 21-38, 1990.
10. Harman, D. : How Effective is Suffixing ? **Journal of the American Society for Information Science**, 42(1), 7-15, 1991.
11. Harman, D.H. (Ed.) : The Fourth Text Retrieval Conference. NIST Special Publication 500-236, 1996.
12. Harman, D.H. (Ed.) : The Fifth Text Retrieval Conference. NIST Special Publication (in press), 1997.
13. Hull, D. : Stemming Algorithms – A Case Study for Detailed Evaluation. **Journal of the American Society for Information Science**, 47(1), 1996.
14. Finding the Right Image: Content-Based Image Retrieval Systems. Special issue of **IEEE Computer**, V.N. Gudivada and V.V. Raghavan (Eds.), 28(9), 1995.
15. Kelledy, F. and Smeaton, A.F. : Phrase Indexing for Information Retrieval. In. **Information retrieval Research, Aberdeen, 1997: Proceedings of the 19th Annual BCS-IRSG Colloquium on IR Research**, London: Springer-Verlag, in press, 1997.
16. Lenat, D.B. : CYC: A Large-Scale Investment in Knowledge Infrastructure. **Communications of the ACM**, 38(11), 33-38, 1995
17. Miller, G.A. : WordNet: A Lexical database for English. **Communications of the ACM**, 38(11), 39-41, 1995
18. Porter, M.F. : An Algorithm for Suffix Stripping. **PROGRAM**, 14, 130-137, 1980.
19. Robertson, S.E. and Sparck Jones, K. : Simple, Proven Approaches to Text Retrieval. Technical Report 356, **University of Cambridge Computer Laboratory**, 1996.
20. Salton, G. : Approaches to Passage retrieval in Full Text information Systems. in: **Proceeedings of the 16th ACM-SIGIR Conference**, Pittsburgh, 1993, 49-58, ACM Press.
21. Sanderson, M. : Word Sense Disambiguation and Information Retrieval. in **Proc. 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval**, Dublin, Ireland, 142-151, Springer-Verlag, 1994.

22. Sanderson, M. : Word Sense Disambiguation and Information Retrieval. PhD thesis, **Department of Computing Science, University of Glasgow**, Scotland, 1997.
23. Singhal, A., Buckley, C. and Mitra, M. : Pivoted Document Length Normalization. in **Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval**, Zürich, Switzerland, 21-29, ACM Press, 1996.
24. Smeaton, A.F. : Progress in the Application of NLP to Information Retrieval Tasks. The Computer Journal, 26(3), 268-278, 1992.
25. Smeaton, A.F. TREC-4 Experiments at Dublin City University: Thresholding Posting Lists, Query Expansion with WordNet and POS Tagging of Spanish. in [11].
26. Smeaton, A.F. : Using NLP or NLP Resources for Information Retrieval Tasks. in **Natural Language Information Retrieval** T. Strzalkowski (Ed.), Kluwer Academic Publishers, (in press), 1997.
27. Smeaton, A.F. and Harman, D.H. : TREC and its Impact on Europe ? **Journal of Information Science**, (in press) 1997.
28. Strzalkowski, T. and Carbello, J.P. : Natural Language Information Retrieval: TREC-4 Report. in [11].
29. van Rijsbergen, C.J. : Information Retrieval (2nd Edition). Butterworths, 1979.

This article was processed using the LaTeX macro package with LLNCS style