Natural Language Processing & Information Retrieval

Alan F. Smeaton

School of Computer Applications
Dublin City University
Glasnevin, Dublin 9

asmeaton@CompApp.DCU.IE

http://www.compapp.dcu.ie/~asmeaton/asmeaton.html

... a tutorial presented at the
Second European Summer School in Information Retrieval
(ESSIR'95)
Glasgow, Scotland, September 1995

1. Introduction

IR is an old, mature area of research in computing / information science / library science ... it is not massivley popular like graphics or databases (based on counts at conferences) ... a homely bunch of individuals !

It is based around a technology which delivers solutions to a market which has been in place for decades ... not great solutions, but ones which work.

This is primarily boolean queries with operators like ADJ and word distances as enhancements though as this summer school shows there are alternatives which are much more attractive.

Originally and for a long time, the IR market was

- libraries on dial-up lines
- patent application offices
- legal and para-legal offices

Boolean IR was attractive because of its efficient implementation using inverted files but

- the difficulties of manipulating boolean logic,
- the comparative complexity of search strategies for the untrained
- the monetary costs associated with using computers in the early days

... led to the emergence of the trained intermediary / librarian as a *go between* bridging the user and the IR system.

Naturally, this was/is expensive and time-consuming

Then the following developments happened:

- The PC and networking came, bringing distributed processing to the desktop ... users used tools themselves, **directly**, users got access to data themselves and started/wanted to do IR, users got comfortable with direct access to powerful tools and dispensed with intermediaries, and now demanded more from IR
- The volume of data, machine-readable text information, has increased staggeringly ... every newspaper, book, technical document, office letter and memo, and newswire.

The combination of these two means many users are looking at IR as a basic technology for underlying applications ... the numbers at our conferences are starting to grow ... SIGIR and TREC and SDAIR and HIM and IR is q component in Hypermedia, DL, others ...

... funding in our area is starting to flow ...

- US Digital Libraries includes IR
- DARPA TREC and to a lesser extent MUC, TIPSTER
- CEC 4FP has Information Engineering and Language Engineering as well as LIBRARIES in the Telematics Programme ... in the 3FP there was LRE ... prior to that IR was banished to ESPRIT to compete with everyone else in the “leftovers” bracket

NLP has, like IR, had a long history but whereas IR has always been smaller but constant, NLP has had many more ups and downs.

The **ups** started with the hype of being able to do machine translation and intelligent IR in the 1960s ... remember the computing power available in those days ? First attempts, and all that was computable for volumes of text at that time, were simple dictionary lookup and even simpler rules.

Translation by literal word transformation is ... bad ... *time flies like an arrow* etc ...

... the initial **up** was hammered by the US ALPEC report in 1965 which stated MT impossible and NLP and AI in general received massive cuts in research funding which continued for many years.

Slowly, AI, or aspects of AI, pulled out of these doldrums and AI as a single field split in all kinds of directions.

... we have seen the rise and 'fall' of expert systems or rule-based systems
... we are seeing the rise of neural nets / connectionism
... etc

The history of NLP is tied very much to the history of AI as NLP was seen as the earliest AI application.

After ALPEC, NLP went into decline in terms of funding, but there was still interest and as computing moved from processing numeric data to processing more and more text in applications like WP, NLP became fashionable again.

Now, NLP is a very large and strong field bridging computer science, linguistics, philosophy, psychology, metaphysics and software engineering.

In February 1992 NSF organised a workshop of 23 invited specialists (IEEE Trans KDE, Feb'93) to identify near-term (5- years) prospects and needs in *Speech and Natural Language Processing* ... top of the list was the *Electronic Library and Librarian* which would use IR technology

... by 2000 technology will allow access to US Library of Congress sized volumes of data though WW has accelerated this even moreso

... how can we retrieve effectively from that scale ... it is going to need to go beyond the current full-text retrieval systems and handle heterogeneous collections, multimedia, etc and statistical approaches alone may be inadequate for this.

An Overview ...

1. Introduction ... this is it !

2. Overview of IR and IR processes ... yeah, you've heqrd this in other tutorials but not my version ... this is about users and authors and information needs and where an IR process fits into the scheme ... the nature of text ... the inexact and imprecise nature of information retrieval ... string searching vs using surrogates ... standard indexing by a bag of words ... desirable features of retrieval ... overview of standard matching techniques **Overview of NLP** ... what is NLP ... stages of NLP, lexical, syntactic,

3. semantic, pragmatic and discourse levels ... NLP applications

4. Applications of NLP in IR ... indexing by base forms of words, by word senses and word sense disambiguation ... indexing by phrases or coordinated terms ... handling ambiguity in noun phrases ... query expansion via linguistic structures ... knowledge representation formalisms like frames and conceptual information retrieval

5. Role of NLP in IR ... a generalisation of what NLP techniques can offer IR and what they cannot and an almost philosophical discussion of the limitations of current NLP

6. Prospects ... for future development

In 3 hours we won't get much done, certainly we won't cover all of the significant efforts in the field but only a representative sample

If I was to condense this tutorial into 1 sentence it would be ...

> *NLP techniques can sometimes be usefully applied to IR tasks but because these NLP techniques were not developed for IR tasks but for tasks like MT and UIFs which are fundamentally different to IR, the eventual 'topping out' of their contribution will not be so great as initially expected, though there is more mileage to be wrung out yet and if we are to use NLP in IR then we need a different kind of NLP, and maybe a different kind of IR.*

2. Overview of IR and IR Processes

We know that ...

types of information ...
Text, Voice, Image, Structured data, Rules, Programs, Animation, Video, etc...
types of information need ...
vague or precise
types of query language ...
ambiguous or exact
types of matching ...
exact or approximate

Putting all combinations together, we only have a subset of all possibilities

Information retrieval is

- text data,
- vague information need,
- imprecise matching,
- exact or an ambiguous query language

But there is more to text management than retrieval ... indexing, routing, classification, extraction & summarisation ... acquisition (OCR), spell checking, critiquing, compression, encryption, editing and formatting ... all are part of text management

It is important to realise that IR is an inexact application ... people tolerate, even expect, to have non-relevant documents retrieved ... this is unlike most other applications of computing ... MT, KBS/expert systems, etc

Indexing and Retrieval, with a bit of clustering perhaps, were the *standard* IR applications for a long time but now the others, routing, classification, abstraction/summarisation, are increasingly important, due to demand.

Routing or information filtering is re-directing a stream of incoming documents in response to a user's information profile

Classification is assigning one or more pre-defined categories to a document and is increasingly being used to reduce search space when searching large or heterogeneous document collections

Application areas for text retrieval ...

- Traditionally in libraries and in legal domain (searching past case histories) and patent applications ... now searching news stories, encyclopedias, office applications, network resource discovery, etc.

Nature of text ...

- Sometimes text documents are strucured into chapters, sections, paragraphs, sentences, clauses, phrases, words, morphemes, letters

- A collection of text, or corpus, can be one single large structured document, or many millions of independent documents ... if they are "connected" or linked that is hypertext and the hypertext/IR bridge is an important development for both fields.

- The ordering of text tokens usually conforms to a known grammar of rules specifying legitimate combinations of such tokens but not true for these notes.

  These notes are in fact in a sub-language for natural language English ... not full sentences, some abbrevs. &c.

  There are many such sub-languages ... technical documentation, email, fault-reports and diagnoses, weather reports, ...

Currently we have (at least) the following forms of written language all possibly emanating from the same person:

- Technical documentation ... terse, tight prose, complex phrases and complex individual sentences needed because conveying complex information ... mostly unambiguous and declarative in nature.

- Journalistic pieces, newspaper articles, short sentences each quite simple, easy to read.

- Storybook prose, as in novels and books. can be complex but such complexity makes it difficult to read. Should be easy to assimilate, reading for entertainment/recreation. Long passages, mixing declarative, quotations, interrogative.

- E-mail messages, ungrammatical, full of abbrevs., dialects and slang, not necessarily full sentences, simple grammatical constructs.

- Office memos, grammatically correct but not as complex as technical documentation.

- Formal language as in deeds, covenants, wills, legal documents, wedding invitations

- ... and others

Usually, written language is more "well dressed" than spoken language, i.e. grammatically sound and well-constructed

Text (and spoken language also) has an intricacy and a complexity as it is filled with synonymity 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 it computationally.

So, how can we do IR ...

- The simplest approach to IR is to do some kind of string searching ... retrieve based on documents containing substrings ... grep ... or more refined “close” matches to substrings via agrep, soundex or string edit distances or even by using n-grams.

  String search techniques catering for approximate matching are now implementable with very fast search times but the problem is encapsulating an information need as a string search.

  In NL, tokens (lexical entries) may modify or be modified depending on their role in the text ... furthermore, because NL text is so complex there are many ways of specifying the same thing.

  As a result, and for other reasons also, simple string searching for word patterns may be efficient but not necessarily effective. They are a poor man’s morphology.

  - What would be ideal would be to have somebody/something read/process the stored information in an intelligent or semi-intelligent way, then read/process our queries and match the two for us but how do we automate this.

  - In order to address the variations within NL, IR systems typically transform an original text into some canonical or intermediate representation (a process called indexing) and the search for a user's query is executed on this.

  - In **INDEXING** the task is to turn text (query a/o document) into a set of terms whose combined semantic meaning is ***equivalent*** in some sense to the content of the original text ... notice that we are looking for a **set** of terms which immediately is a "cop-out" ... information is much more structured and connected than a **set** of concepts but to make it computable and scaleable this is what IR did in the early days.

It is the “bag of words” problem and it applies whether we index by words, phrases, whatever.

A document (or even a query) will have been written by a person, from the population and that person will have had information to convey and have done so by selecting legitimate combinations of words from a vocabulary where the words and word combinations relate to concepts and concept relations unknown to everyone except the mental state of the author at that time. Because of the complexity of NL as a mechanism for encoding information, and the size of the universe and thus the number of possible pieces of information which might be conveyable, it is impossible to determine by simple lookup table, the mapping between all possible legitimate sequences in the vocabulary all abstract concepts and concept relations.

But we still want some kind of information retrieval ...

In order to achieve something computable the search space must somehow be reduced, and the traditional and conventional approach to IR has been to represent concepts and concept relations (i.e. the information being conveyed) as a set of independent ***index terms***, normally the words which have occurred in the text/query;

a	a	a	a	and	and
are	but	combined	computable	concepts	connected
content	cop	document	does	equivalent	for
immediately	indexing	information	into	IR	is
is	is	is	is	it	looking
make	meaning	more	much	notice	of
of	of	of	original	out	query
scaleable	semantic	sense	set	set	set
some	structured	Task	terms	terms	text
text	than	that	the	the	this
to	to	to	turn	we	what
which	whose		in		

Before jumping into specifics of indexing, we note that indexing can be done on several levels which can vary from one extreme to another ...

- just index by the words in the text, as they occur, but this is bad because of word variants, difference between function & content words, semantic word equivalences ...

- identify word level equivalence where, for example, {vibration, undulation, pulsation, swing, rolling} -> oscillation, in an aeronautics domain

- identify concept level equivalence where *"prenatal ultrasonic diagnosis"* indexes:

    * sonographic detection of fetal ureteral obstruction
    * obstetric ultrasound
    * ultrasonics in pregnancy
    * ultrasound in twin gestation
    * midwife's experience with ultrasonic screening

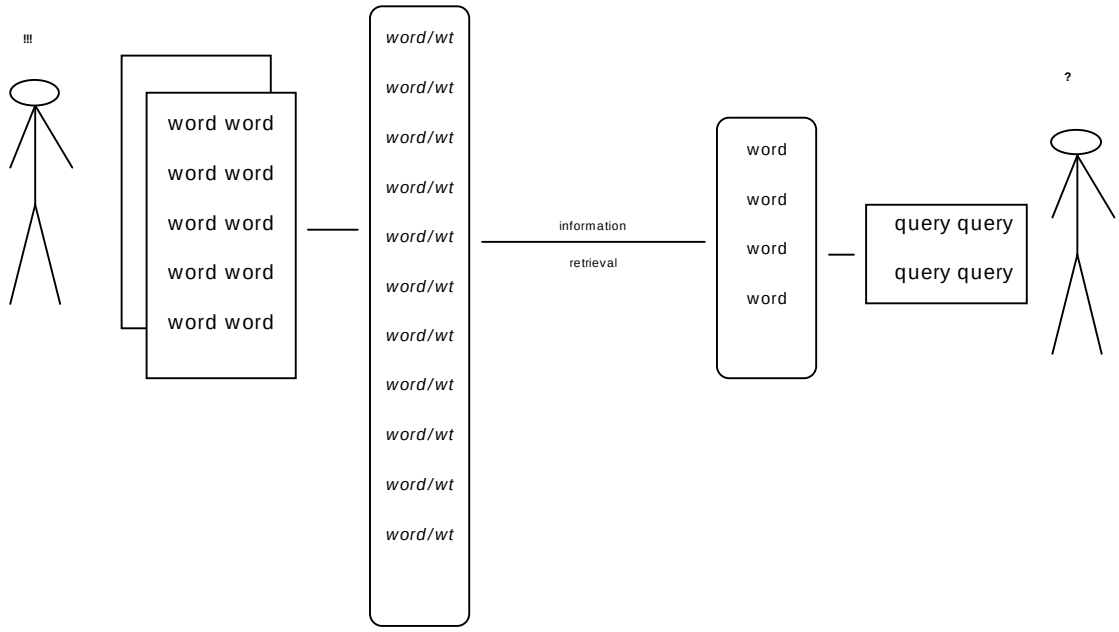
    ... this would be based identifying concepts as they occur in a text stream and indexing by the concepts, independently of the particular word/phrqse combinations used ... such concept level indexing ideally produces phrases as indexing terms, is semantically rich, costly, laborious, specialised and almost entirely manual and is done in some commercial applications

- beyond concept identification lies the possibility of concept and concept relationship identification, but this is really difficult to do for IR.

The simplest approach to indexing is to index directly by the words that occur in the text

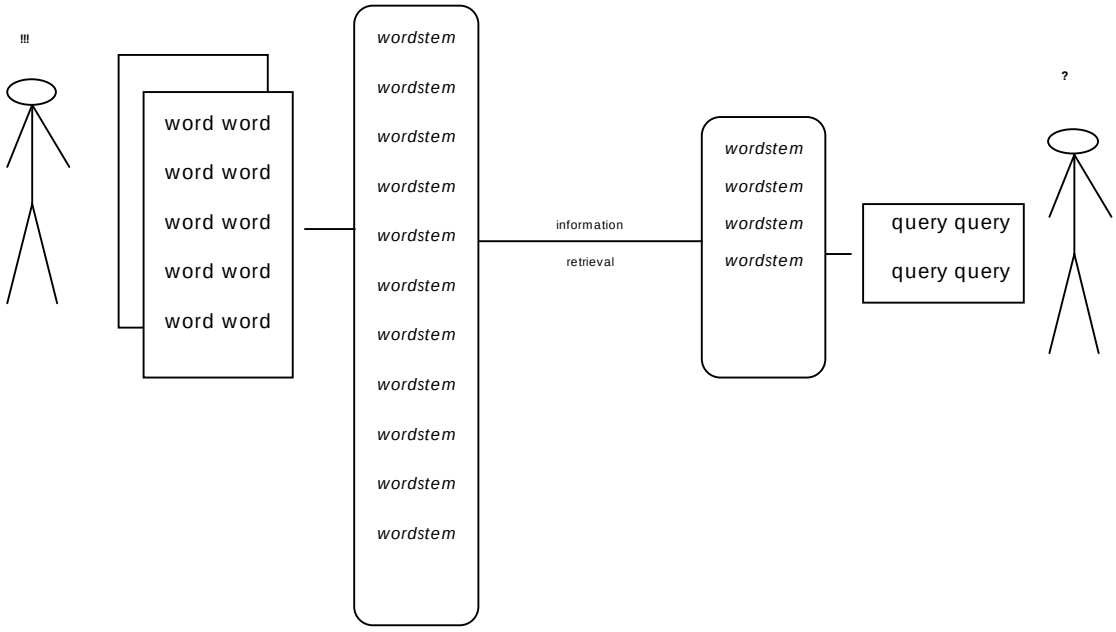
* most frequent words are function words
* least frequent words are obscure
* mid-range words are content-bearing

... so index by the mid-frequency words. This can be refined by the basic term weighting indexing methods commonly used in IR ... the most well-known and general is ***tf*IDF*** weighting and there are many variations on the basic formula.



Rather than index by words alone, we can refine this by Stemming and Conflation were the indexing terms are word stems, not words ... a simple and crude linguistic process which is OK if used consistently for both documents and queries to cause a query-document match.

Usually more effective than using raw word forms as stems normalise morphological variants, albeit in a crude manner.



There are other approaches to indexing into phrases, into word senses, into more structured representations, etc., but that is enough to give the basics of indexing as it is conventionally done.

When we look at retrieval we see there are a number of desirable features we would like:

- ranked output rather than sets
- relevance feedback from user back into the retrieval process, used to help retrieval ... learn or adapt the strategy
- query modification/expansion during retrieval as users become clearer on their own information needs.

There are several metrics or association measures between objects to be classified which could be used as retrieval functions ... Dice, Cosine, etc.

These heuristic methods from other fields are a useful starting point. Furthermore, they can be used in conjunction with weighted indexing of texts and/or of user queries and are applicable to a number of internal representations ... words, stems, word senses, etc.

To make progress on simply *grabbing heuristics*, several approaches to formally modelling the retrieval process have been made using different mathematical formalisms, and the most successful approaches have been based on probabilistic and Vector Space theories

These statistical methods in retrieval produce a ranking of documents based on estimated probability of relevance to a query using evidence like the number of documents containing query terms and number of occurrences of terms in documents.

There are a number of other important aspects to text retrieval as follows:

- Cluster based retrieval ... depends on pre-clustering document collection into cliques of similar documents, possibly generating a centroid

- Extended Boolean Retrieval ... a combination of boolean and ranked retrieval by weighting the strength of interpretation of the boolean connectives ... more effective than boolean and addresses the mid-point between ranking and boolean IR but never took off because of the complexity of understanding weighted boolean operators.

- Retrieval as a combination of several retrieval strategies ... data fusion ...in experiments on TREC collection (see later) and in our own experiments on structured documents it has been found that a combination of rankings from several different approaches can actually bootstrap to an even higher level of effectiveness.

- Relevance Feedback ... a good thing, used in probabilistic retrieval and also there are formulae to re-weight query terms based on their (non-)occurrences in known relevant texts

- Query Expansion can be a follow-on or derivative of relevance feedback if one selects index terms (whatever they are) from known relevant documents, manually, though there are a variety of formulae for ranking candidate additional query terms ... I did one in

1983 ! Query expansion can also be from a static structure like a thesaurus, but that is really query formulation.

- Latent Semantic Indexing ... based on the statistical technique of Singular Valued Decomposition where an n x t matrix is reduced to an n x δt matrix, statistically, effectively dimensionality-reduction to c.100 to 300 dimensions (index terms) which incorporate term-term dependency relationships ... and it is computationally expensive ... but it works.

- Some computing isevolving towards distributed, co-operative processing ... distributed text retrieval is big due to large collections being inherently distributed and the increasing growth of internet ... people want to be able to search +1 text database with one single search ... this is distributed text retrieval which led to the emergence of WAIS from TMC et al., and the emergence of Z39.50

- IR delivers **documents** in response to user queries and on these users make relevance judgements, but what if documents are not abstracts but full text ... hence the emergence of passage retrieval where **places within documents** are retrieved in response to a query ... this is difficult to evaluate (in terms of P-R) which is something IR likes to do ... not known how to handle.

- An aspect related to passage retrieval is the problem of applying standard IR techniques to heterogeneous lengths documents ... with relatively minor variations one can normalise by document length but this pre-supposes documents are about topics treated equally throughout a (long) document ... not so ... alternative is text-tiling, chopping documents up into “pages” of approx same length using crude or more sophisticated techniques.

- Document texts can be many homogeneous independent documents or few (one ?) large, **structured** document ... IR techniques can take advantage of the structural relationships between segments of text ... grammars for structured documents, markup languages like SGML, etc.

- Efficiency aspects ... some people work in an area trying to deliver faster implementations of current IR indexing and retrieval

techniques using new data structures or organisations, or taking advantage of new, mostly parallel, hardware.

Most of IR research and certainly IR implementations, are based around the above areas of work and have varying levels of sophistication but (almost) all rely on keyword techniques which operate at a symbolic, literal, text-matching level.

Additionally, they assume ...

- the user knows what words/terms are used in documents
- how those words are used in different contexts
- how to spell those words
- the user has a precisely formed goal and can articulate that goal into an accurate representation as a query

Most of these assumptions are false. Peter Ingwersen will cover (tomorrow) the fact that searches can be for known documents (verificative) or searches may be exploratory in an unknown (to the user) domain.

What all this means is that the IR process, and research in IR to improve that process, is far short of making IR a *solved problem.*

Natural language processing may offer IR something more than another refinement of the same ... before we look at that though, a brief overview of NLP.

3. Overview of NLP

NLP is (roughly) divided into computational and theoretical linguistics

The aims of ***computational linguistics*** are to develop systems for processing natural language ... they aim to handle *most* cases of NL and can cope with approximations or inexact solutions ... don't mind occasional failures ... more concerned with getting systems working

... whereas ...

theoretical linguistics is concerned primarily with things like grammatical coverage, principles of grammar, grammatical formalisms, deermining the single universal and ultimate grammar or fragments of a grammar to handle certain linguistic constructs ...

Theoretical linguistics feeds into computational linguistics.

CL is an engineering rather than scientific discipline.

Goals of CL are to develop systems for processing natural language for applications like:

- Machine Translation from one NL to another; early 1960's saw lots of work in the US but realised that for progress to be made then fundamental problems of knowledge representation would have to be addressed, so funding cut; nearly end of AI; Now MT is big in Europe (Eurotra, METAL) and in Japan.

- Information Retrieval or conceptual IR used to be an early goal which had little early success for same reasons as above but now IR is not a named goal of CL.

- Man-machine Interfaces, arguably a natural language interface is more ideal than an artificial language but certainly NL interface has a role in some element of MMI.

NLP research currently supports two schools ...

1. Symbolic, grammar-based approach, rule-based, rules to detect NPs etc
2. Statistical, probabilistic approach using observed probabilities of linguistic features and based on corpus evidence to find most likely analyses

Because the former is more mature, it has been used most in IR, but the greater potential is for the latter as current IR processes and corpus linguistics have the same underlying philosophy.

In order to build complex systems to process NL the task is usually divided into sub-tasks with an increasingly blurred distinction between them ... originally these levels were independent but cooperating ... for the purposes of overviewing NLP and for highlighting NL problems, I will stick with the categories

For IR, the levels of interest are lexical, syntactic, semantic and discourse ... we ignore phonology and bundle morphology in with lexical processing.

3.1 Lexical level

In order to process a sentence of a language, the elements/tokens must be identified. For NLP the lexical level processing involves identifying words and determining their grammatical class so that syntactic level processing can use this.

Lexical processing operates at the single word level, independent of context.

To do this we need a dictionary or vocabulary of known words for a domain. (Unknown words tend to be proper nouns).

Words in language can have affixes and suffixes attached to them depending on their function in language. English mostly has suffixes, a small number of them and not too complex compared to German, or even worse, Finnish !

Thus the verb "to cover" can appear as:
cover, covers, covering, covered, uncover, etc.

Not all words are regular though:
blow, blows, blowing, blew
am, is, is, was
...

In NLP, morphological language analysis involves breaking down a word into its morphological components or constituents, as in

"uncovering" -> "un-cover-ing".

This is a word-level process and is usually followed by lexical lookup, i.e. in a lexicon or dictionary, which sounds simple but is not because of the presence of lexical ambiguity.

Arguably the single most important problem in NLP is handling ambiguity, which arises at all levels of language processing, including lexical.

Lexical lookup ideally determines one base form of a word, and one syntactic label, but not always so as

- many nouns can act as verbs
- most noun plurals are created by adding -s, so also the 3rd person singular form of verbs

"covers" could be a noun or verb:

The *covers* were blown off the jam jars.
He *covers* the jam jars with lids.

"leaves" could be a form of the verb to leave, or the plural of the noun leaf:
He *leaves* behind a great legacy.
The *leaves* blew in the Autumn wind.

It is impossible to resolve the many instances of lexical ambiguity at this level and it is the task of higher levels of language processing to distinguish between them.

Advantages of this level of language processing for IR:

- Efficient;
- Lexicons are available (more later);

but

- Doesn't give much on its own.

NB We are not concerned at this level with different senses of words.

3.2 Syntacic level

Traditionally syntax has been

- the structure of a sentence with semantics meaning the actual content
- parts-of-speech and the set of rules acting on them determining grammaticality
- set of rules determining which ordering of words are allowed

Researchers at this level have been *primarily* concerned with the construction of wide-coverage grammars and the development of efficient parsing strategies.

Grammar formalisms have also been studied, phrase structure grammars, context-free grammars, context-sensitive grammars, transformational grammars, definite clause grammars, constraint grammars, and many more in order to try to capture vagaries of language.

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

The aim of syntactic processing is to determine the structure of a sentence but that structure can be ambiguous ... there is that word again !

The input to this process (probably) has lexical ambiguities and structural ambiguity can arise in syntactic structure itself, due sometimes but not always to lexical ambiguity.

- "I saw her duck"

... did you see her dive down to avoid a low-flying object, or did she show you her feathered friend. This structural ambiguity is caused by lexical ambiguity in "duck".

- "Sheep attacks rocket"

... same story with lexical ambiguity of "attacks" and "rocket".

But,

- "I recognised the boy with the telescope"

... who had the telescope, you or the boy. This is pure structural ambiguity without any lexical ambiguity.

Three common sources of pure structural ambiguity in English are PP attachment, coordination and conjunction, and noun compounds.

3.2.1 PP Attachment:

PPs can be attached to almost any syntactic category like verb phrases, noun phrases and adjectival phrases, in order to act as modifiers.

"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" - 13 PPs!

The problem with PPs is in finding out to what they should be attached:

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

are both lexically identical but in the former one removes bolts **which have** square heads and in the latter one removes bolts **using** a wrench.

In general, higher levels of language processing (semantics) are needed to try to resolve problems of PP attachment, and even this sometimes cannot be done.

3.2.2 Coordination & Conjunction:

Conjunction or coordination is one of the most frequently used constructions in natural language but the scope of conjunctions, i.e. what is being conjoined, can almost always be ambiguous.

Example, conjunction among heads of a NP:

- "Inspect the *bearing cups and cones*" ... bearing cones ?
- "Inspect the *hub and bearing components*" ... hub components?

Conjunctions can occur almost anywhere, among modifiers, among PPs, among heads, among clauses, ... and are used to make language more concise.

However, the price for this is ambiguity, which is usually resolved at higher levels of language analysis.

3.2.3 Noun Compounds:

Noun (nominal) compounds occur when a noun (or nouns) is used as a modifier of another noun, making a compound structure as in

"computer performance evaluation".

Performance, a noun, modifies evaluation, another noun. Computer, a noun, modifies ... performance evaluation or just performance ? We don't know, hence the ambiguity.

Also, what kind of relationship exists between nouns in a compound ?

- Fighter plane ... made for
- Garden party ... held in a
- Timber house ... made from

Noun-noun compounding is very common in formal and in technical English as a nominal compound is expressing something that is too complex to be expressed in a single word in the language (until one single word descriptor is invented).

In addition to the 3 above sources of ambiguity, another, less common cause of structural ambiguity is adverbial ambiguity, which like PPs is due to attachment. Adverbial phrases are not as frequent as PPs, so less of a problem.

Not all adverbs are ambiguous when resolved by higher level processing:

- "The robber *quickly* ran into the woods"

dealing with fast-running robbers.

But some are:

- "The robber *probably* ran into the woods"

Did the robber run or walk, or did he run into the woods or run elsewhere ? Even we cannot know from this sentence alone and thus must resolve using the context in which the sentence appears.

The final problem with ambiguities is that they are potentially multiplicative rather than additive, so long and complex sentences, as in technical and formal writing, will be likely to have much ambiguity.

The main advantages of syntactic level processing for IR:

- It gives more than lexical processing;
  ... it determines sentence structure as well and sentence structure can be indicative of content.
- It can be made efficient;
  ... much work has been done on developing efficient parsing strategies in NLP and in processing of artificial ‘programming) languages and the mechanical process of parsing is now reasonably well understood.
- The rules of syntax are general and concepts like word classes are abstract;
  ... this means that the process is domain-independent, except for the lexical input, so a syntactic analyser developed for one domain could be ported to another.

but

- There are many ambiguitites it cannot handle and it needs higher level analysis to do this;

- Is not inherently robust at handling ill-formed input. If a sentence is not legal according to the grammar, it fails, but parsing can be made to handle this.

3.3 Semantic Level Language Processing

concerned with context-independent meaning, taking one sentence at a time, independent of its more global context in the text/discourse.

Focusing on broad questions like what type of KR formalism to use and how to interpret things like:

John only introduced Mary to Sue

which could actually mean ...

- 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 NL can be processed which should:

- be unambiguous
- have simple rules of interpretation and inference
- have a logical structure
- facilitate hierarchies to define sub- and super-types of concepts, so concept-relationships can be made explicit; eg Toyota and Ford are sub-types of cars, and Corolla and Carina are sub-types of Toyota
- allow role structures to define components of entities, for example in a physical injury there are 2 important roles: the injured and the injurer; as both may be the same, we distinguish by giving each a name and assign the name to a role or slot.

In AI, the earliest attempts at understanding meaning used various forms of logic but more recently, AI represents knowledge by specifying primitive or simple concepts and then combining or structuring them in some way to define complex, real-life concepts.

These, in all their flavours, capture permanent, universal objects and their relationships quite well but there are other aspects of natural language which need to be addressed.

NL discusses notions of modality (possibility, necessity), belief and time, and it is essential/desirable/necessary for any semantic representation to capture these elements of NL as NL can be so succinct.

Capturing and reasoning about these aspects of language is non-trivial and there is no universally-agreed KR formalism which does this.

Semantic level NLP should be able to analyse grammatically parsed text into a KR format and should also be able to "parse" the semantics of input, to note and 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.

I noticed a man on the road wearing a hat

leads to two syntactic interpretations with the participial phrase "wearing a hat" modifying the man or the road ... semantic level interpretation should tell us that hats are worn by animate objects (men, donkeys, etc) and this the latter interpretation should be discarded.

- *"Freedom is dark green"*
- *"My closet is well behaved"*

This assumes that all input is supposed to make sense, which is reasonable.

However, in order to perform this kind of reasoning, an enormous amount of domain knowledge is needed for all words in the vocabulary.

We need to know the properties of all objects and we need to know the legitimate arguments of all verbs, and building a KB to support semantic level processing is a huge task.

Advantages of semantic level processing for IR:

- It gives the meaning;

but

- No best KR formalism;
- it requires huge domain knowledge;

3.4 Discourse level language processing

concerned with the study of context-dependent meaning, the meaning of an entire conversation or text, taking all parts into consideration, knowledge of the world, who is writing and reading, etc.

Wrestles with problems at the text/discourse level including things like presuppositions:

- "The king of America is at this tutorial"

presupposes a king of America exists.

Indirect speech acts:

- "Can you sit up ?"

could be interpreted as a yes/no question by a hospital visitor asking about a patient's health or it could really be a request from a visiting doctor.

These are the subtle hidden meanings in spoken and in written text.

An example of a discourse phenomena is anaphora, a phenomenon of abbreviated subsequent reference, eg using pronouns, a technique for referring back to an entity introduced with more descriptive phrasing earlier, by using a lexically and semantically attenuated or abbreviated form.

It is used orally and in written texts to avoid repetition and improve cohesion by eliminating unnecessary re-descriptions.

Anaphora reminds the reader/listener of something and the more "distant" the anaphoric reference from the target, the more detail is needed in the reference:

"*Computers* are often mixed up with questions about *their* impact on the ability to learn" (7 words)

"*Computer systems*, on the other hand, can undergo many changes. Every time a new program is added to *such a system* ..." (16 words)

Detecting anaphora and resolving the reference would improve our understanding of a text or discourse but even detection is difficult as there are no indicator terms.

In IR it may be of interest to identify anaphoric constructs as they may be hiding the real distribution of statistics on concept appearance in texts ... most extensive studies on anaphora in (traditional) IR on document abstracts found:

- Anaphora in abstracts are used to refer to integral rather than peripheral concepts

- Manual analyses show there are an average of 12 potential anaphors per abstract with an actual use of 3.67 (Av) ... so there are red herrings !

- A simple resolution of replacing each potential anaphoric word occurrence by the nearest preceding word matching in gender and number would resolve 70% of potential anaphora, of which 60% would be correct.

  This was tried on CACM and CISI and others -> marginal improvement in retrieval effectiveness.

- Manually and correctly resolving anaphors in texts and performing retrieval provided mixed results, some queries were improved, others worse ... another strange result.

Resolving anaphora would seem to be (intuitively) a good thing to do, but we don't know how to do it properly and reliably, and we don't know what to do with it when we do resolve it.

Consensus is that anaphora resolution should be treated with other discourse level phenomena and should form part of an overall semantically-based NLP on text.

3.5 Applications of NLP

... the successful ones of course !

I. Machine Translation:

METAL, developed by Siemens in Münich is arguably the most technologically advanced MT system in commercial use.

Supporting German-English, English-German, German-Spanish, French-Dutch, Dutch-French, and probably more.

METAL has extensive lexicons and parses input text into a tree structure for an entire sentence. It then applies a translation rule base assigning probabilities to each structural interpretation. From that the tree is transformed into a predicate and argument construct, like a case grammar, from which a translation is generated.

METAL translation runs on a SYMBOLICS Lisp machine, consists of 100,000 lines of LISP code plus lexical database plus grammar rules, runs as a translation server on a network, in batch mode, translating 200 pages (40,000 words) in 8 hours (1992 figures)

It runs as a machine-aided translation system, as most MT does and is marketed by Sietec http://www.sietec.de/

SYSTRAN was initially developed for USAF in 1964 to translate Russian -> English and sold in 1970's to C.E.C. who developed it further and use it still with 27 language pairs available.

CEC use it, not by translators but by Eurocrates in need of a quick rough translation of official documentation (CEC translation budget is 450MECU p.a.) and this will get worse with Swedish and Finnish.

As well as the mainframe version there is now a PC version for Windows with 9 language pairs and costing US$1500 for q single user ... it translates on a 486/DX with 16Mb, 1 x A4 page in 5 seconds ... info@systrqnmt.com

LOGOS are at http://islander.whidbey.net/~logos/

and there is a translation experts page http://www.net-shopper.co.uk/softwqre/ibm/trans/index.htm

II Natural Language Interfaces:

NL-IF technology is mainly the same as MT, but translation into a formal language like SQL.

For example, Loqui is a NL interface system developed by BIM in Belgium from an ESPRIT project started in 1983; Loqui is available since 1990.

Written as 35,000 lines of BIM-PROLOG, Loqui has interfaces to most major Relational DBMS, but like all NL-IFs, must be tailored to new domains.

A criticism of NL-IFs are that they are too verbose but that ignores the important dimension of natural language dialogue. Sentence by sentence, NL-IFs are slow, but dialogue-processing features make NL concise, short follow-up questions about a topic, ellipses, anaphora. Conversation/dialogue modelling is the approach.

So there are a couple of examples ofishere NLP is used but can NLP be performed efficiently ? Yes ... I could quote figures but they would be out of date ... suffice to say that NLP analysis is fast enough to operate at indexing time, though not dynamic NL analysis at retrieval.

4. Applications of NLP in IR

Traditional keyword based approaches to text retrieval (statistical, VSM, probabilistic) involving statistics on word occurrences will always have inherent limitations and possibilities for text manipulation.

For example, keyword based retrieval cannot handle things like ...

1. Different words, same meaning:
   Stomach pain after eating = Post-prandial abdominal discomfort == belly-ache
   Throttle == Accelerator
   ... the latter is a case of straightforward synonymity but not the first case

2. Same words, different meaning:
   Venetian blinds v blind venetians
   Juvenile victims of crime v victims of juvenile crime

3. Differing perspectives on single concept:
   "The accident" v "the unfortunate incident",
   prosecution and defense in court

4. Different meanings for the same words in different domains:
   "Sharp" can be a measure of pain intensity in medicine or the quality of a cutting tool in a gardening book (or in medicine also!).

Restrictions like these provide the simple motivation and justification for attempting to use NLP in IR ... what is being wrestled with is language and it follows that language processing rather than string manipulation, is more likely to yield better IR

Previously I have presented indexing and then retrieval, in the context of NLP but here I will bundle the two together.

A. Indexing by Base Forms:

Simplest attempts at using NLP have been at the word level, indexing texts by normalised or derived form of individual word occurrences, possibly based on **word base forms** rather than word stems, however this has not really be explored as:

1. All potential words must be in the lexicon, building this is expensive ... unknown words are proper names or proper nouns ... proper name recognition is an active area ...
2. Lexical analysis can lead to ambiguity which is only resolved at higher levels of NLP
3. It can only be slightly better than mechanical stemming.

More important than all that however is the fact that if one has gone to that much trouble to look up in a lexicon then not much further effort is required to apply some higher level language analysis.

Interestingly, exptl. results have consistently shown stemming algorithms and true base forms of words to be approx. equal in overall, retrieval effectiveness.

B. Indexing by Word Senses:

As an enhancement to indexing by potentially ambiguous base forms of words, the potential of indexing by **word senses** was explored. Here, each document/query is indexed by the non-stopwords which occur but also by which *sense* of each word is intended.

One of the significant developments in lexical level NLP for IR has been the availability of conventional dictionaries in machine-readable form (MRDs) over the last few years.

Originally the publisher's typesetting tapes were made available, now they have been cleaned up and made easier to manipulate.

Available MRDs include:

- Longmans Dictionary of Contemporary English (LDOCE)
- Websters 7th Collegiate (W7)
- Oxford English Dictionary (OED)
- Collins dictionary of English
- Oxford Advanced Learners Dictionary

... and others

Sometimes use of these is restricted with a licence fee, sometimes it is completely free.

I'm not sure why they have been made available, possibly because publishers feel they can be made use of, moving towards electronic publishing, but these are publishers who make money out of selling books and now are moving towards electronic journals ...

Formats of dictionaries vary from MRD to MRD but include a definition for each semantic sense or interpretation of a word, each of which has:

- Syntactic class of word, parts of speech
- Short and concise textual description of meaning
- Morphology
- Semantic restriction information, constraints on verb arguments
- Subject classification, circuit -> engineering

In my concise OED (paper) the word BAR has the following entries:

(n) long piece of metal
strip of silver below clasp of medal as additional distinction, a band of colour
rod or pole to fasten or confine on a window
immaterial restriction
place for prisoner
rail dividing off space
pub counter
place for refreshments
(v) to fasten with a bar
(n) large Mediterranean fish
(n) unit of pressure, $10^5 N/m^2$
(prep) except, as in racing.

... and there could be more ! The bar is a legal exam in the U.S. and it is also a bank of sand or mud under water.

MRDs could be used to help index texts and queries by word senses.

Indexing by word senses is intuitively more pleasing than indexing by words or word stems as a word sense is a more accurate description of a concept however, it does not yield a structured or semantic representation of text.

Ambiguity of grammatical categories can be handled by parsing, sometimes, but word sense disambiguation is more difficult, though not the same as or as difficult as, semantic interpretation of language ... a kind of intermediate. It ‘wsd) is a task which must be completed in all NLU applications yet at present seems driven by IR as an application.

Furthermore, it is possible that statistical approaches to retrieval (and indexing) could be used on top of word sense indexing and thus word sense indexing could be plugged into existing retrieval strategies, even existing systems.

With these goals in mind, researchers set out to investigate and much work has been reported in recent IR literature, but the experiments to date have shown mixed results ... this kind of work has only been possible recently ‘cos of the availability of MRDs.

Krovetz and Croft (TOIS’92) reported extensive research on word sense ambiguity using CACM and TIME test collections where the sense disambiguation was done manually and they found that sense mismatches occurred when documents were not relevant to queries. Their results showed WS ambiguity causes surprisingly little degradation in IR and for those corpus it seemed perfect WSD would yield only 2% improvement

Voorhees built an automatic sense disambiguator based on WordNet and tried it on a variety of standard test collections (SIGIR93) but got no improvement in IR performance ... this was borne out by subsequent work by others ... this is surprising and analysis has thrown up the evaluation of wsd as an unknown quantity ... manual checking is too costly.

An approach of artificially introducing sense ambiguity into texts based on Yarowsky's pseudo-words was reported by Sanderson (SIGIR94) on Reuters collection (20 Mb) and a series of experiments run to measure the effects of word sense ambiguity on IR performance ...

... his conclusion was that IR performance is very sensitive to erroneous disambiguation ... say 75% accuracy ... don't do it at all rather than do it incorrectly ... only when it gets to 90% accuracy it is as good as no disambiguation ... beyond that, it yields improvement but only when the query is short.

This really puts it up to those who do wsd ... 90%+ before it is useful ... and it must be fast also, 'cos we deal with large volume texts.

There are some other considerations in these experiments ... Reuters is GP text, so more ambiguities in words whereas CACM is fairly domain-specific and thus wsd is easier or words will always have only 1 sense in such restricted domains ... some lexicon vocabularies have finer-grained senses than others, eg WordNet is notorious for this.

(much of the following discussion is abbreviated from recent postings to the CORPORA mailing list)

On the other hand Schultz and Pedersen found 7% to 14% improvement on a baseline when they added a sense disambiguation module though they did not use a MRD as a reference source for senses. K&C and Sanderson and most others have used an MRD to define the senses of words as reference points but for S&P, their assumption was that the senses of a polysemous word are defined by clusters of common contexts surrounding the word in question, not from a static, predefined, list of possibly incomplete or alternativey too fine, senses.

For example, the word *ball* occurs

- x times in the context of Graf, tennis serve
- y times in the context of rugby, Australia, try
- z times in the context of gown, evening, party

then each one of these contexts is defined as a sense, and we don't use unused senses at all but assign each occurrence of *ball* to one context.

Sanderson says he recently ran experiments confirming the frequency distributions he used in his SIGIR94 simulations were approximate to the word senses from a dictionary.

So, while MRDs kick-stqrted word sense indexing in IR, do we need such static sets of sense definitions, which because of their very fine distinctions (cf WordNet) or unused senses or sense definitions not fine enough, cause problems ?

Sperberg-McQueen (CORPORQ) talks about '*the implicit assumption that the senses in published dictionaries are disjoint'* ... not quite the same as statically-defined

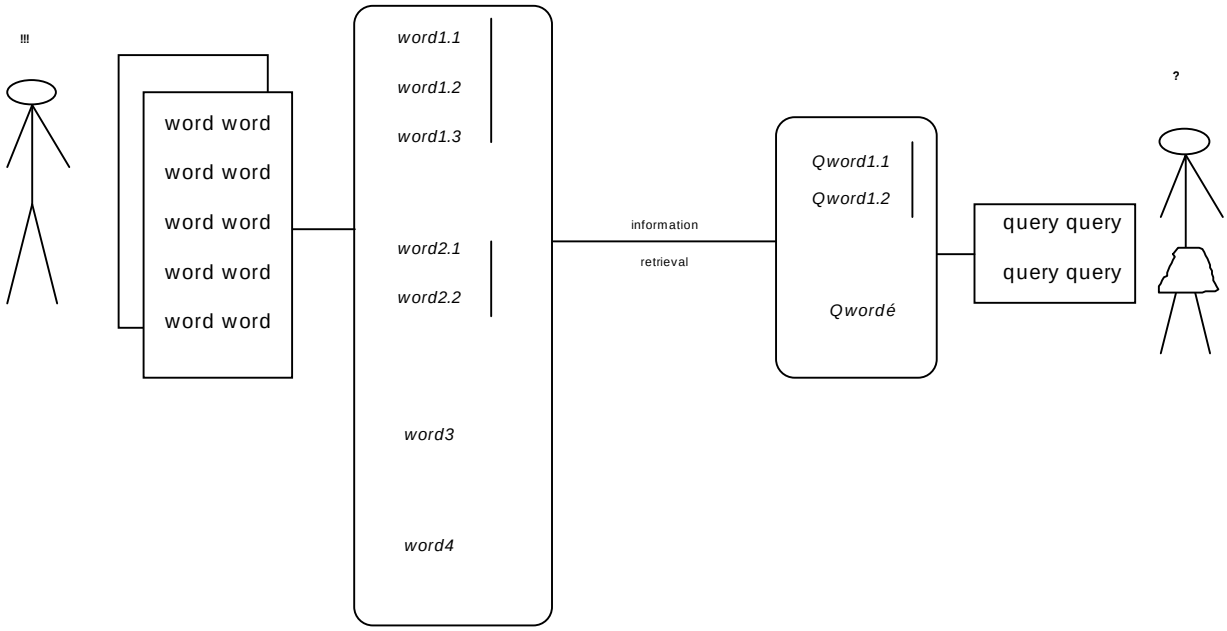
... '*Since senses are not always disjoint, any ambiguity resolution which chooses exactly one active sense is inherently wrong in any case where more than one sense applies'*

It may be that the ideas of Krovetz/Croft hold the best prospects for this area if we can never do accurate wsd anyway ... they believe it is not necessary to always determine the single correct sense of a word but rule out unlikely senses and weight likely senses highly is an improvement on indexing by words or word stems which, by implication, is indexing by **all** senses of a word

... in many cases it isn't clear anyway what senses of words are intended or because of fine-grained sense, they are indistinguishable

John went to the BAR for a drink

Perhaps the senses/contexts are defined from the corpus and if we never have a mention of the Mediterranean fish sense of the word BAR, then we don't have to know it exists !



So,why do IR people work in WSD ?

- it is an interesting challenge in its own right
- humans do it or at least don't get confused by sense ambiguity
- an MRD is a new and interesting thing and is a useful first approximation of the concept of word senses, so lets play with the new toy

Others (KSJ) are more sceptical about wsd in IR citing that other mechanisms in IR, i.e. adding many, many terms in query expansion indirectly achieves wsd through conjoining of terms or increased weights due to co-occurrence of terms.

C. Indexing by Phrases:

What about indexing into larger, more complex units of meaning ... phrases ?

Any piece of text or dialogue which contains information essentially consists of a description of an object/action relationship.

To encode the complexity of the information we deal with:

- objects may be modified with adjectives, prepositional phrases, etc. car green car green car with a dent

- actions may be modified in various ways (adverbs for example, “ran SLOWLY”) and the modifiers themselves may include descriptions of other information ("he ran slowly with an obvious limp"), so things can become terribly recursive and ambiguous

In order to capture the true meaning of text, the objects and actions taking place on those objects should be encoded as should the modifiers in their correct roles.

Single keywords, word senses, syntactic labels, don’t do this ... moving beyond indexing by single words or single yet independent tokens, no matter how disambiguated or precise, we have to look at more complex indexing units ... phrases.

When we perform **indexing by phrases** we index into a vocabulary, the set of phrases, which is richer than the set of words or word senses, thus if we have a richer representation format, and we can translate text into this accurately, we should get better quality retrieval.

... while the size of the set of words for a corpus of texts tends to level off at c.100,000 which is manageable in modern computing, the set of all possible phrase is some factor larger than that and we cannot manage, thus we define a subset of all possible phrases.

It has been assumed by researchers that in text it is the noun phrases that are the content-bearing elements

... certainly they are more content-bearing than single words but phrases are not a full representation of meaning, yet NPs are good indicators of text content, and for traditional IR, that is what we want.

Ignoring relationships (verbs) and relationship modifications (adverbs, PPs, etc) is part of the “cop out” of the IR task.

How do we identify phrases as indexing units ?

We can identify good words (single) using statistics and some have tried to identify good word groups, statistical phrases, using co-occurrence data but really one has to use NLP to identify phrases.

Statistical approaches to phrase identification may be more efficient (‘cos of the way computers are built) but NLP processes are getting faster, machines are getting more powerful, so the efficiency argument is weakening ... a few years ago when people statred looking at indexing by phrases and managing to do experiments, the statistical approaches were more attractive on computational grounds and the NLP-based phrase indexing was still finding its feet.

(Syntactic analysis) can be used to determine ((the boundaries of ((noun phrases)) in (text/queries))) but the problem with indexing by NPs has been the variety of ways of representing a concept which is so complex that it needs a complex NP

... this can lead to the same words used in 2 different phrases but with different use yielding completely different meaning(*blind venetians* for example).

Instead of just marking NPs in text which would not be so good for generating a usable index because phrases similar in content but different in syntax/word usage would be heterogeneous, parsing could be used to identify the heads of each clause ... that makes (intuitive) sense, doesn’t it ... heads are the most important parts of a NP, right ?

... but ambiguity still remains w.r.t. scope of modifiers (see later).

Unless the derived phrases are very short to address ambiguity, say only 2 words, then simply marking phrases is inadequate as there is too much to be done at retrieval time in identifying the variants in NPs.

To address this there have been 3 approaches tried to date:

- Ignore
- Normalise indexing phrases
- Index by structures which capture the ambiguities.

C.1 Ignoring Ambiguity in NPs:

This approach allows texts to be indexed directly by phrases as they occur in texts and depends on the matching/retrieval to do something about the problems of ambiguity, different ways of expressing the same concept.

A query can be coded as a pattern matching rule to operate on words and their syntactic patterns in text. Thus the pattern matching rule:

NP:[* adj:[large] * noun:[box] ? PP]

searches for noun phrases which have occurrences of the base forms of the words "large" and "box", optionally followed by a PP, and with * indicating zero or more other constituents.

So searching for large boxes as above would not retrieve "a large box top" but would match "a large almost invisible box with a lid".

This is really string searching on more than the alphabet ...coding of the patterns is the problem ...no right-minded user would want to express a query in such a language but automatic construction ??? and recent improvements in approximate string searching applied to such processed texts ... interesting possibilities now though earlier attempts not successful.

Indexing texts by phrases as they occur has been carried out by at Cornell, initially by Fagan and more recently by Smith, Buckley and Salton. They have used a parse of text to identify head-modifier relationships from which indexing phrases have been derived.

They have also used statistical and adjacency information to index by phrases and have found comparable retrieval effectiveness levels using either method, though statistical is much more efficient.

Interestingly, the indexing phrase sets have little overlap, suggesting that neither approach is ideal.

C.2 Normalising the NPs in Indexing:

This approach is to index texts by some processed version of sets of words as they have occurred in texts. The advantage is that it yields a smaller vocabulary and makes retrieval less complex as syntactic variants in texts and in queries should always be normalised to the same form.

When this is done then the retrieval process can default to the techniques used to match keywords or word stems or word senses ... statistically based, weighting, etc ... the philosophy here is to make the **retrieval** operation as computationally lightweight as possible.

In the FASIT system, which is old and dated but whose principles are used in its more modern successors, syntactic labels were assigned to words in text and then a rule base examined the tags looking for content-indicating patterns.

Example rule:

NN NN-VB GN -> concept(1,2,3).

(Noun followed by a word which is either a noun or a verb, followed by another noun)

"Catalogues are produced on magnetic tape cartridge"

yields

"Magnetic Tape Cartridge".

The normalisation aspect appears in the rules which do not have to index by phrases which have the same word occurrence pattern as in the text.

"Cartridges for magnetic tape ..."

GN PP NN NN-VB -> concept(3,4,1).

An alternative approach to indexing by normalised phrases has been taken in the CLARIT project at CMU/CLARITECH

Before the indexing of input texts takes place, a first-order thesaurus for a domain is generated - this is essentially a word or phrase list for a domain and is based on linguistic processing to identify commonly occurring NPs.

Then an input text is parsed by a probabilistic or stochastic grammar and candidate noun phrases as content indicators for the text are generated, based on content-indicating patterns.

These are then matched against the phrase list and classified as:

- *Exact:* candidate terms are identical to those in the thesaurus so index by those terms.
- *General*: terms in the thesaurus are found as constituents of terms in the candidate set so index by the term in the thesaurus
- *Novel*: the leftovers require special processing

Example ... candidate term from parse ...

AUTONOMOUS ROBOT NAVIGATION SYSTEM

General match with thesaurus term: ROBOT NAVIGATION

CLARIT has been taking part in TREC and their performance has been (in TREC-2 & less so in TREC-3 anyway) among the best ... there is computational overhead with their methods but they have overcome this.

A similar approach to indexing texts by phrases where the phrase **vocabulary** is pre-determined from a linguistic analsis of a corpus segment, is taken by the UMass group (SIGIR95 paper) where they pre-construct an association thesaurus

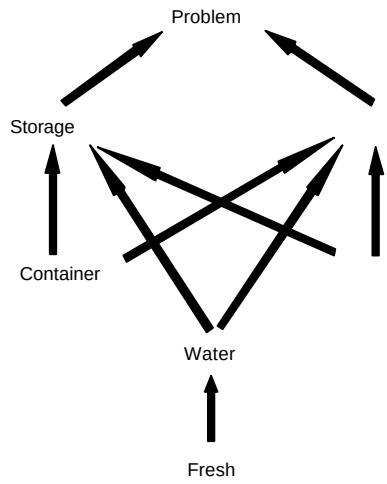
C.3 Capturing NP Ambiguities in a Structure:

The final approach to handling ambiguity in noun phrases for indexing is to encode the ambiguity in some structured representation in the indexing component and to allow retrieval/matching to handle the ambiguity automatically by offering the different (ambiguous) interpretations at matching time.

The TINA/COPSY project at Siemens applied shallow parsing to input texts and used this to identify noun phrases. From these NPs, dependency trees were built which identified explicit links between words.

These dependency links mirror all possible head-modifier relationships in NPs and the approach is to create links of equal importance and type between all possible dependencies, from the parse.

... problems of fresh water storage and transport in containers or tanks...



These dependency trees can be used in retrieval where similar dependency trees/links are generated from queries and the database is searched for

graph isomorphisms with a partial ranking generated the stronger the overlap.

Another way to use the dependency trees would be in helping a user frmulate a query ...

User:	I am interested in storage
System:	What kind of storage ... I have milk storage (10) or water storage (2) or heat storage (1)

... interactive query formulation using frequencies of dependency links to home in on link occurrences **known** to be in the database

... query formulation IS retrieval !

A group at the University of Pittsburgh developed the Constituent Object Parser (COP) and also building dependency trees from a syntactic analysis of text. These trees were binary and at each level the dominant branch (containing the head) is marked with an *.

The "dominant branch" in a phrase is the branch which is modified in some sense (adjective, PP, etc) and the COP system assumes that dominance is transitive, i.e. if A modifies B and B modifies C then A modifies C

Dependency trees cater for syntactic variants of the same concept, or for a simple concept embedded in a complex phrase:

In the SIMPR project, we at DCU have use a linguistic analysis and identification of content-bearing text fragments as earlier, to generate ako dependency tree like Siemens, except we encoded rather than enumerated possible dependency/modification links as in COP.

In the phrase “water storage and transport” we encode the ambiguity with the scope of the modifier "water" on transport.

In terms of retrieval we have evaluated this in TREC-3 and it was not as good as simple statistical weighting on single word terms as we were generating too much noise.

D. Using NLP Resources for Query Manqgement

Previously we have looked at the inadequacies of keyword/word stem based retrieval for handling word variants, same meaning but different words, etc. All of the work on indexing using NLP that we have looked at to date has addressed only cases were the same words in different syntactic relationships describe the same concept.

NLP tools, techniques ***and resources*** may also be used in addressing another keyword inadequacy, handling related terms. This can be done using NLP resources rather than NLP processes, in the same way word sense indexing uses MRDs

A well-established technique in IR is query expansion ... adding extra index terms ot the query based on occurrences in reldocs and non-occurrence in nonrels ... or using *a prioiri* statistical co-occurrence distributions, nearest neighbours, min/max spanning trees, etc ...

This is becoming popular with CLARIT qnd recent Umass work on association thesaurus ... corpus-specific determination of word relationships.
Massive query expansion (c.000s terms per query) adding statistically-derived terms works well in TREC-3 (Cornell)

This, however, is statistical exploitation of term-term relationships.

From a linguistic viewpoint, there are structures which yield term-term relationships... thesauri ... which may be domain-independent or domain-specific.

Hover, these are outside the context of a given query or document, which is probably to their detriment, longterm, but not yet ‘cos there could be some mileage to be had in them.

The largest initiative in this field is Cyc but this is ongoing and we wait and see ... some are sceptical they will ever complete, others doubt the

homogeneity of the effort over such a large domain and timespan, most people wait and see.

Roget's thesaurus is available but those using it have found it limited, lightweight, small and inadequate, but if it is all you have ... it is a start.

Others are trying automatic thesaurus construction from linguistically analysed texts ... ongoing ...

Miller's WordNet, from Princeton, has had mixed reviews and has/is been used in IR ... I know of 3 groups at least who have bolted it on as a reference for users during query formulation ... a freebie version of the thesaurus in word processors !

On the automatic side, Voorhees has expanded TREC (-1 and -2) queries by adding WordNet synonyms of original nouns, weighted down slightly over original terms and average results more effective than SMART retrieval but higly variable across queries ...

... some queries are improved, others disimproved by adding synonyms of incorrect senses of words.

WordNet has its pros and cons, but IR does not know how to use it effectively yet.

Taking an alternative tack to query exansion for statistical IR, we (DCU) have derived hierarchical concept graphs from WordNet, calculated information content values for nodes by frequency of co-occurrences from 19M word noun corpus and developed a mechanism to traverse these trees to measure word-word semantic distances.

In comparison with psychological testing, we are as good as humans at ranking the similarity of noun pairs.

Our work to date has tried some queries from TREC on WSJ data only, but not successful ... we have addressed problems of computational overhead by pre-computing 150M pair similarities and we have demonstrated improvements in Q vs (short) image captions.

What about the potential of using NLP resources ... (thesauri and MRDs)?

... they have not been the *significant* breakthrough, but they have given us insights

E. Indexing into Semantic Formalisms

All the material to date has been about using NLP tools, techniques and resources for conventional IR ... what about trying more advanced IR ?

In indexing into formalisms based on semantics we can try to go beyond traditional IR functionality where semantic level NLP can be used to process input text into a semantic representation of the contents of the text

... however dynamically building an accurate semantic representation of a text (document or query) is hard, so much so that it is usually done by hand in other NL applications.

Thus, the KR formalism used to represent the content of text should be something as easy to encode as possible.

The most commonly used formalism in IR-like applications is based on frames.

What makes frame based representations suitable for dynamically encoding information from NL is that the pre-defined or prototype frames are blank and are gradually filled by the language analysis yielding instance frames ... frames are a richer representation format than independent words or phrases because the bind these elements together

There is no necessity for all slots in a frame to be filled as each slot can be classified as optional or mandatary with respect to its filling ... so it is not all-or-nothing !

Frame-filling in NL analysis is usually assisted by a domain-specific knowledge base which can represent information about words, their lexical properties, their relationships and their constraints, as frames, or as semantic nets.

A component of domain-specific knowledge which is often needed in dynamic NL analysis are scripts which are domain-dependent and describe typical sequences of events in the domain.

> Scripts are usually hand coded as in SCISOR, but FERRET explores learning of scripts from language analysis.

An example of a frame for the sentence:

> "Alan is a senior lecturer at Dublin City University"

Person Frame:

Agent:	Alan1
Occupation:	senior lecturer
Employer:	Dublin City University
Salary:	- unknown
...	

A subsequent sentence:

> "Alan took an Aer Lingus flight to Glasgow yesterday."

Flight Frame:

Agent:	Alan1

Origin:	- unknown
Destination:	Copenhagen
Carrier:	Aer Linguss
Date:	(today - 1)
Time:	- unknown
Fare:	- unknown
...	

A correct analysis would note the connection between the sentences and would fill the **Agent** slot of the **Flight** frame by the **Person** frame filled by the instance Alan1.

There would be a constraint that the agent of a flight must be a person name or person frame and there would be a script for flying which looks for agents, origins, destinations, etc, to identify fillers for slots.

As mentioned when introducing semantic level NLP, these kind of huge, domain independent KBs required for IR-scale processing simply are not present yet.

The series of MUC exercises (same lines as TREC) presented this task of text analysis into frames ... arguably this is not IR but halfway between IR and KBS, and it was a very narrow domain.

The FERRET system from CMU parses texts into case frames providing traditional IR functionality but most work on indexing into more elaborate KR formalisms tries to provide conceptual information retrieval or question-answering, ... START, SCISOR, RESEARCHER, OpEd, etc.

Further details on this in my Computer Journal overview paper.

One final point about QAS and conceptual IR is that it is very very difficult to evaluate quantitatively in the sense that IR indexing and retrieval techniques can be evaluated and measured via P-R.

5. The Role for NLP in IR.

Large-scale applications of NLP tend to be domain-dependent requiring much coding of Kbs, so we are not going to get full interactive, domain-independent language processing of large text bases for retrieval, but do we need it in IR ?

It is believed by many that the problems NLP wrestles with are unimportant for information retrieval, which already has so much vagueness and imprecision inherent ... its tolerance of “noise” is great.

Some (KSJ for example) have argued that trying to do natural language **understanding** for IR on large text bases is not only not on but it is unclear whether full-fledged NLP would yield the desired payoff in retrieval effectiveness ...

> If a user wants to retrieve documents about apples or about elephants, an IR system does not need to know what an apple or an elephant is, or what the difference between them is, it just needs to find areas of its corpus which ***might*** be about apples or elephants because the decision on relevance is something that is ultimately made by the user, not the system.

Weizenbaum, while discussing Schank’s CD, has stated that “it is hard to see ... how Schank’s scheme could probably understand (the sentence “will you come to dinner with me this evening?”) to mean a shy young man’s desperate longing for love” ... (that was in 1976!)

... but maybe the kind of deep, meaningful analysis required to do this kind of processing is not only beyond us, but not needed in IR

... why ?

... ‘cos in IR we don’t need to comprehend or wrestle with the meaning at all ... all we need to do (in IR) is distinguish texts from each other, in the context of a specific query ... perhaps sub-texts, perhaps generate ranking, whatever the task is.

Also, current NLP does not suit IR anyway, they have different philosophies altogether;

A weakness or NLP is that it is designed to determine whether or not its input is well-formed, rather than the question more pertinent to IR which is ... ‘what does this sentence mean ?’

This is due to the way in which we have evolved our thinking on syntax and semantics ... syntax has been concerned with characterising well-formed structures in a language and all work in semantics which is computatioally implementable is usually piggy-backed on top of thiswhich cannot be good as that evolved *view* of semantics is not an IR view !

So, given this *cop out*, that current NLP does not suit IR, what can NLP be used for in IR ...

- Indexing ... as a way to identify coordinated terms of good phrases as content indicators as an alternative to the “bag of words” ... the “bag of phrases” ?
- Query formulation ... NLP analysis of a user query dialogue to support information seeking
- Comparison operation ... matching Q with D with dynamic NLP analysis, involving inference perhaps
- Feedback ... altering a query in response to user judgements
- ... others ?

In practice it is indexing, and by implication, retrieval, and conventional retrieval at that, which has received most attention in applying NLP to IR

Although we looked at indexing, the retrieval operation which would have to follow can default to statistically-based retrieval as the impact of NLP upon IR proceses has been to try to improve the quality and range of the

internal representation of D and Q, and retrieval simply follows using standard, conventional approaches.

... and so the commonly asked question is, what should we replace the bag of stems with but is this the right question to be asking ?

Certainly this is the question to ask in an incremental approach to IR research.

What can we say about the performance of all these approaches to information retrieval based on NLP techniques

- ... the emphasis has been on NLP of text at indexing time but some believe that work on phrase extraction should not be done during indexing but during retrieval, in the context of a given query.

  This would *seem* to make sense but goes against the tradition of IR where the work is done at indexing time in order to provide fast retrieval.

- word sense indexing seemed intuitive at first but wsd problems remain and hold this up from developing further ... there is new-found doubt about the validity of a static set of sense references anyway

- indexing by phrases, based on NLP rather than statistical techniques, again seems intuitive, but no major leap in progress to date although incremental improvements are being found ... problem is that this is in the context of treating a phrase as an indexing token and defaulting to word-based retrieval techniaues.

- NLP-based systems are impacting the IR research community and are now impacting the commercial marketplace, but tend to be quite specialist and expensive (CLARIT, READWARE, ORACLEs ConText (which I hqve found it impossible to get further information on) for example)

- semantic based (FERRET, SCISOR, etc.) is VERY domain-dependent and specialist and a long-term goal. These “knowledge-intensive” approaches have not been evaluated yet on large domains.

- Statistically-based text retrieval is efficient, large scale, domain-independent and, despite years of people saying "... has reached its upperbound of achievable effectiveness" ... just keeps getting better ... look at how TREC results have improved in a few years ... *'there is more mileage in the old dog yet'*[1]

- The biggest success for NLP in IR is at the morphological level while techniques based on relationships, within and between phrases has had only marginal success to date ... 'cos we don't know how, not 'cos it can't be done.

In short ... it is a mixed bag of results we have to date ... we know what does not work and a few things that do.

Lewis & Liddy have said that like Edison, we have discovered 1000 things that do not work, and a few that do ... they have also noticed a number of important phenomena for IR:

First the things we can handle ...

- Words exhibit morphological variation

- Words are not all good indicators of content

- Words are polysemous ... one word, multiple meanings.

- Two words can have related meanings, i.e. be synonymous

And the awkward things which make IR difficult ...

- Queries and their relevant documents are rarely identical since only parts of each match parts of the other, and which parts and even the matching is not obvious

- Documents are not about one thing ... they are long and compositional ... original information retrieval was for abstracts with

[1] a quote attributable to my mother !!!

high consistency, IR on full text would perform better if it took into account the linguistic characteristics of full text and did (even simple) discourse linguistics ... text tiling is a good example of this !

- Not all things are explicitly said ... when we write text we assume an intelligent interpreter ... ourselves ... not an information retrieval system.

David Blair wrote a book in 1990 and a follow-up article in the June 1992 Computer Journal discussing the Philosophy of Language and how it bears on the task of Information Retrieval where he states that "*because of the linguistic nature of Information Retrieval there are simply too many degrees of freedom in design for us to arrive at good designs hapazardly*."

From that it follows that if IR is based on language in some way as he suggests than theories of how language words will help us with IR ... this seems sensible and an obvious follow-on!

But, he also makes the point that "our language was never meant to make the kind of subject distinctions that it is being called upon to make in large-scaled systems" ... i.e. NL evolved as a mechanism for man-man communication but are we now straining the information-bearing capacity of our language and will this cause us to re-think and reconsider the levels of effectiveness we can expect to obtain when searching large corpus ...

TREC data is 2.5 Gbytes of text and all (80 ?) people here in this room reading at 180 wpm, no breaks, no rests, 24 hours per day; it would take just over 12 days to read that amount ... in IR it is now "standard" to search that volume ... forget about efficiency, disk space, resources, etc., ... that is all natural language and doing something more clever than simply counting words must improve the quality.

I used to be very upbeat about the potential of NLP for IR tasks, and so were many people but because of the lack of significant breakthrough, the slow plodding progress, there is a hangdog feeling.

I am still upbeat though and my sights have not been lowered, I just realise it will take a lot more time to get there.

ACKNOWLEDGEMENTS:

Over the years I have benefitted from discussions and correspondence with the following people who have contributed either directly or indirectly to the material presented in this tutorial ...

Yves Chiaramella, Bruce Croft, David Evans, Joel Fagan, Donna Harman, Karen Sparck Jones, David Lewis, Liz Liddy, Ruairi O'Donnell, Ray Richardson, Keith van Rijsbergen, Mark Sanderson, Peter Schaüble, Paraic Sheridan, Tomek Strzalkowski and many others.

Further Sources of Information on NLP &IR ...

- The three copied reference papers have good bibliographies
- IEEE Expert recently had a special track on knowledge based information retrieval in which there were some papers.
- Journals ... IR & NLP papers appear in
    - Information Processing and Management
    - Journal of the American Society for Information Science
    - ...others scattered in C.ACM, IEEE Computer, ACM TOIS, Computer Journal, AI Review (sometimes, special issue on KBS and IR planned)
- IRList electronic digest, ir@mailbase.ac.uk
- corpora@hd.uib.no has bursts of activity, much to do with IR
- The REALTOR language resources server from the CEC project of the same name has resources in speech and text corpora, lexicons, NLP programs and tools and related database systems. HTTP://www.XX.realtor.research.ec.org where XX stands for the 2-letter country code of nearest EU country ... resources are obtainable by ftp and some through AFS where executables can be shared ... advice@realtor.research.ec.org

