

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>

1. Introductions ...

Who am I ...

researcher, lecturer, project worker, supervisor, editor, author, ...

DCU

What do I do and have I done in the past ...

- ESPRIT I MINSTREL
- ESPRIT II SIMPR
- VALUE project
- NLP & IR ...
 - ESSIR lectures, SIGIR tutorials 91, 92, conference & journal papers
- CEC: Information Engineering, Language Engineering, LRE
- I also work in IR & hypertext/hypermedia
- SIGIR is THE IR conference and I am on PC for last few years and I held in in 1994
- PC of TREC since TREC-1 and now also a TREC participant

... I am firmly in the Information Retrieval camp.

What am I doing here ... i.e. why a tutorial on NLP&IR at an EACL gathering ?

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.

Originally and for a long time, this market was

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

Then the following developments happened:

- The PC came, bringing distributed processing to the desktop ... users used tools themselves, directly, users started/wanted to do IR, users got comfortable with the autocontent wizard in PowerPoint etc, and now demand 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 ... 130 papers (twice) submitted to SIGIR instead of 60 and we also have TREC and SDAIR and HIM and others ...

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

- US Digital Libraries includes IR
- DARPA TREC and to a lesser extent MUC
- 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

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* ... by 2000 technology will allow access to US Library of Congress sized volumes of data ... how can we retrieve 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 will be inadequate for this.

So, what am I doing here at EACL (again) ?

Many different disciplines are “*looking at information retrieval*” as a vehicle for trying out other technologies, including at least ...

- logic processing, fuzzy logic, default logic, new logics
- fast secondary storage devices
- parallel processing / distributed processing architectures
- neural networks
- KB techniques (Cyc for example)
- machine learning
- Rule Based Systems (expert systems)
- user modelling
- databases
- data fusion
- ... and NLP ... tools techniques and resources

Much of this work hasn't a clue what it is doing in IR, isn't aware of the volume of (unsuccessful and successful) research which has already been tried ... read and cite Salton's 1989 book and suppose that is the field summarised ... it is following the gravy train.

This EACL tutorial is about “bringing out” information retrieval, roller-coasting through it, whizzing through some of the past and current uses of NLP tools, techniques and resources in IR applications, including a bit of self-indulgence.

An Overview ...

- This is the **introduction** ... who am I, you, etc.
- **Introduction to Informaiton Retrieval**... covering user's information needs, IR applications and application areas, nature of text, string searching ...
- **Indexing** as an IR process ... “bag of words/phrases”, indexing by words, stemming, frequency based weighting
- **Retrieval** ... implementations using inverted files or signature files, retrieval features including ranking, query expansion, relevance feedback ... simple similarity measures ... mathematical modelling using VSM or probability theory ... other aspects ...
- **Evaluation** ... statistical evaluation and TREC ... how do we measure in IR
- **Basics of NLP** ... just to get the terminology agreed ...
- **Role of NLP in IR** ... what can NLP be used for, and why bother
- **Indexing and Retrieval Using NLP** ... based on base forms, word senses, phrases and phrase handling and covering both indexing and retrieval together ...
- **Performance and Prospects** ... a summary
- **Further sources of information** ... printed and electronic, plus annotations.

In 3 hours we won't get much done, but a start and some pointers and a grasp of what is there.

So that's who I am, who are you and what are you doing here ?

- ... cost is only £17.50 if you are a student booking early or £45 if you are not and you book late, so you might as well be here !
- ... in Dublin at EACL anyway and an excuse to extend your visit
- ... understood my abstract, but not Martin Kay's
- ... may have actually been interested in coming

You ... know the basics of NLP so I don't cover things like parsing, KR formalisms, lexicons, grammars and grammatical formalisms, word senses, other ambiguities like nominal compounds and PP attachment.

Your background can be ...

- computational linguistics, applied or not
- cognitive science
- computer science
- HCI
- ... some combination

The language is English ... there is not much going on in IR for other languages (some LRE, some LE and some TREC) compared to English and many of the techniques could be applied to other languages.

My sources of information ... material is adapted from my own notes for a graduate course I teach in IR, past (SIGIR) tutorials, Lewis/Liddy tutorial & general awareness of my field.

2. Introduction to Information Retrieval

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, and an 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, extraction/summarisation, are increasingly important, due to demand.

Information retrieval techniques are being used in applications of non-text indexing and retrieval, eg on image/video captions, but only 'cos we can't (yet) do content retrieval on non-text media.

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 ...

- 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.

- Text usually conforms to a known grammar of rules specifying legitimate combinations of 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. There are many such sub-languages ... technical documentation, e-mail, fault-reports and diagnoses, weather reports, ...

Currently we have 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, i.e. grammatically sound and well-constructed

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.
- As NLP people, you know that 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.

Computational linguistics is the study of computer systems for performing automatic natural language processing (NLP).

If automatic natural language processing can process natural language efficiently, correctly and robustly then NLP obviously could have many roles in information retrieval.

- 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.

3. Indexing

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	a	a	a	and	and
are	but	combined	computable	concepts	connected
content	cop	document	does	equivalent	for
immediately	in	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				

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

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 ... but why phrases ?

Concept level indexing can also produce words as indexing units, semantically less rich, still costly and laborious, manual but machine-assisted.

The realistic alternative to concept indexing is to produce word indexing which does

{word -> term}

rather than

{word -> concept -> term}

indexing, but it is achievable, and can be done automatically using a variety of approaches:

The simplest approach 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 noting that:

1. The more a document contains a given word, the more that document is about a concept represented by that word.

2. The more rarely a term occurs in individual documents in a collection, the more discriminating that term is.

This yields 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

Here 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.

3 stages:

- o remove high-frequency stopwords
- o suffix strip remainder ... many algorithms exist to do this whose performance in terms of effectiveness, are all about equal
- o detect equivalent stems and conflate (absorb, absorpt)

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

Porter's (1980) algorithm is popular:

1. remove plurals, -ED, -ING
2. terminal Y -> I when another vowel in stem
3. map double suffixes to single ... -ISATION
4. deal with -IC, -FULL, -NESS
5. take off -ANT, -ENCE

6. remove -E if word > 2

Each rule set has a set of conditions examining number of vowels, consonants, vowel-consonant patterns, etc.

There are other stemmers ... the Frakes/Baeza-Yates book has some with pointers to source code on the net ... other stoplists also available, again see that book.

Language dependent ... English, American, French,

May I have information on the computational complexity of nearest neighbour problems in graph theory.

INFORM, COMPUT, COMPLEX, NEAR,
NEIGHBOUR, PROBLEM, GRAPH, THEORI.

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 IR

... more elaborate representations are based on NLP tools techniques and resources, so we will come across them there.

4. Retrieval

So now we've seen how to index text (and queries ?) into a set of terms, stems, words, whatever, and possibly weighted. What can we do with them vis-a-vis retrieval ?

Two orthogonal aspects to retrieval are the implementation approaches and the retrieval algorithm used.

Regarding implementation, the usual methods are to create an inverted file, or a signature file, to act as an index ... text is turned into some internal representation which is transformed into an index to facilitate fast retrieval.

In the most common implementation strategy for IR, from a given query is generated a list of index terms or keywords in the vocabulary. User queries to such systems are Boolean combinations of word occurrences using AND, OR, NOT and parentheses. The retrieval operation is implemented by retrieving document sets for query terms and then using set intersection/difference/etc.

Experienced users use the system interactively, gradually building complex queries by refinement.

An essential piece of information during querying is the postings information ... how frequently a term is used in the collection of text ... the number of document entries in the inverted file record.

Enhancements to the basic retrieval on word or stem occurrences include:

- term truncation (using wildcards)
- adjacency/distance information which requires positional information in the inverted file, which in theory allows more precision.

Suitability of these as search mechanisms ?

- laborious, time-consuming -> costly
- sometimes ineffective, other terms ?
- intimidating & off-putting
- Boolean formulations are restrictive and not powerful for subtle queries
- Can be iterative as it is fast but no learning/adaptation, no feedback from the user.

Now that we've looked at implementation of retrieval and the simplest strategy, lets examine retrieval approaches and algorithms

There are a number of desirable features we would like in text retrieval:

- 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.

Simplest is the overlap, or number of terms in common between Q and Di.

Assume X and Y are objects (document/query) being compared

$$|X \cap Y|$$

... simple overlap measure but this generally yields only a partial ranking

Other measures normalise the "score" as

$$\frac{2 \cdot |X \cap Y|}{|X| + |Y|} \qquad \frac{|X \cap Y|}{\sqrt{|X|} + \sqrt{|Y|}} \qquad \frac{|X \cap Y|}{|X \cup Y|}$$

... giving us the Dice, Cosine and Jaccard similarity measures, respectively ... and there are others also.

These heuristic methods from other fields cannot go far, but 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 in many instances two modelling approaches have led to the same retrieval mechanism !

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.

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 $n \times t$ matrix is reduced to an $n \times \delta t$ 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 is evolving 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.

5. Evaluation of Text Retrieval Methods ...

Means the evaluation of all processes is normally based on the performance of the ultimate, retrieval.

Normally via test collections

- set of text requests for information, 50 to 100
- set of documents
- set of relevance judgements, i.e. which documents to retrieve for which queries

Sometimes via interactive experiments with real users, but not often.

Recall is the proportion of retrieved documents which are relevant
Precision is the proportion of relevant documents which are retrieved.

PR figures are calculated for each (of 50 ?) query, interpolated and averaged to give average PR for the “run” ... sometimes the single average P at standard recall points is given, sometimes the P at 11 standard R points, sometimes P at a cutoff of 10 or 30 documents.

For many decades and up to just a couple of years ago, the test collections were small ... few Mbytes, few thousands of documents ... CACM collection is typical ... larger bodies of text existed but no queries/reldocs existed ... then came TREC.

TREC (Text **RE**trieval Conference) is part of DARPA HLT program, same lineage as MUC ...

Organised and run by NIST ... it is a benchmarking exercise where IR groups with IR systems run same queries on same texts, top X are pooled and manual relevance assessments made on the pool

... participating systems can then have accurate PR figures computed and at a closed workshop/conference participants present their systems and their results ... is it a competition or a benchmarking ?

TREC has shaken IR into reality w.r.t. size ... collection is 2+Gbytes of WSJ, SJMN, Federal Register docs, patent applications, bibliographic records, ... heterogeneous in nature.

1995 is TREC-4 with 55+ groups applying to take part.

TREC participation is open, 3 categories, unfunded but free, international, most of the IR groups are there, from 512 node SPARC multiprocessors which read the **entire** collection into RAM, to 486 PCs processing CD-ROMs ...

From TREC-1 the hardest part of participation was engineering the size of the collection, but in 3 years we have come a long way in computing.

It is planned to have the TREC data, queries (currently 200), reldocs, results from runs, etc, released to all researchers, but it is not hard to get hold of this now ... Donna Harman (harman@magi.ncsl.nist.gov)

TREC benchmarking is of ad hoc retrieval and static routing and in TREC-4 there are specialist tracks of NLP, multi-lingual ad hoc, collection merging, corrupt/OCR data, and, interactive.

And Europe ... CEC LE program has a preparatory action as part of 4FP looking at evaluation and assessment of NLP technologies and since LE has as one of its stated applications, it is looking at evaluation of the NLP, possibly in the IR context.

6. Basics of NLP

... this is really just to get terminology right.

Computational linguistics aims are to develop systems for processing natural language ... 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 with things like grammatical coverage, principles of grammar ...

Theoretical linguistics feeds into computational linguistics.

CL is an engineering rather than scientific discipline.

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 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.

For IR, the levels of interest are lexical, syntactic, semantic and discourse.

6.1 Lexical level

... identify words and their grammatical class (not word senses), word at a time, handling word morphology, utilising dictionary/lexicon.

Ideally lexical lookup determines one base form and grammatical class for each word but not always so ... “leaves” and “covers” are examples of words which are ambiguous ...

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

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 do this.

Processing at this level is efficient and lexicons are being made available but doesn't give us much on its own

6.2 Syntactic level

...traditionally syntax meant the structure of a sentence, the parts-of-speech and their set of rules acting on them determining grammaticality, or simply the set of rules determining legitimate sequences of words

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.

6.2.1 *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.

6.2.2 *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.

6.2.3 *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 is invented).

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.
- It can be made efficient;
... much work has been done on developing efficient parsing strategies 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 ambiguities 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.

6.3 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.

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.

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;

6.4 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 !
- Syracuse have developed rules for anaphor resolution which do not replicate human cognition, they can't, they simply capture most of the linguistic patterns of anaphor occurrence.
- 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.

7. The Role of NLP in IR

Traditional keyword based approaches to text retrieval (statistical, probabilistic) involving statistics 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
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 in different domains:
"Sharp" can be a measure of pain intensity in medicine or
the quality of a cutting tool.

Restrictions like these provide the simple motivation and justification for attempting to use 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 (in 1976!) 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"

... 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.

So, given that cop out, 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, which has received most attention in applying NLP to IR

... and so the fundamental question is, what should we replace the bag of stems with ?

Although we now look at indexing, the retrieval operation which would have to follow can default to statistically-based retrieval as the impact of NLP upon IR processes has been to try to improve the quality and range of the internal representation of D and Q, and retrieval simply follows.

Other IR-related applications are also potentially suited to using NLP ... automatic abstraction / summarisation, back-of-the-book indexing, thesaurus generation, NL interfaces, etc ... but we will restrict this to indexing and retrieval.

8. Indexing and Retrieval Using NLP

Previously I have presented this as indexing and then retrieval ... here I will bundle the two together.

Simplest attempts 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 been 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.

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.

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 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, 10^5N/m^2
- (prep) except, as in racing.

... and there could be more ! The bar is a legal exam in the U.S.

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

Some interesting facts ...

... the 20 most frequently occurring nouns in English have an average of 7.3 senses and the 20 most frequently occurring verbs have an average of 12.4 senses

... in user queries it was found that terms have c.7.5 senses and document terms have c.4 senses

... this suggests a need for word sense indexing.

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.

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.

It is possible that statistical approaches to retrieval (and indexing) could be used on top of word sense indexing.

With these goals in mind, researchers set out to investigate and much work has been reported in recent IR literature, but the limited 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 the most 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 (good) and they got good results.

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.

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 ... Reuters is GP text, so more ambiguities in words whereas CACM is fairly domain-specific ... some lexicon vocabularies have finer-grained senses than others, eg WordNet is notorious for this.

It may be that the ideas of Krovetz/Croft are best prospects if we can never do accurate wsd they believe it is not necessary to determine the single correct sense of a word but rule out unlikely senses and weight likely senses highly ... many cases it isn't clear anyway.

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 thing-1, and thing-2 that was done to that first thing-1, i.e. an object/action relationship.

To encode the complexity of the information we deal with, the thing-1 may be modified with adjectives, prepositional phrases, etc.

The thing-2 action (verb phrase ?) may also 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

In order to capture the true meaning of text, the objects and actions taking place on those objects should be encoded.

Single keywords, word senses, syntactic labels, don't do this ... moving beyond indexing by words, 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.

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 IR.

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.

Syntactic analysis can be used to determine the boundaries of NPs 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 same words used in 2 phrases but different use => completely different meaning.

Instead of just marking NPs in text which would not be so good for generating a usable index, parsing could be used to identify the heads of each clause but ambiguity still remains w.r.t. scope of modifiers.

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.

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

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

8.1 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".

Hand coding of the patterns is the problem.

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.

8.2 *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, 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)

yields "Catalogues are produced in magnetic tape format"¹
 "Magnetic Tape Format".

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.

"Formats 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.

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

¹ Ignore the verb sense of the word format

General match with thesaurus term: ROBOT NAVIGATION

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

Their real bottleneck is in the indexing since they default to statistically-based approaches on **phrases**

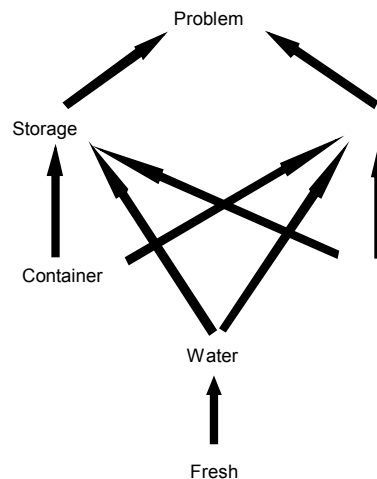
8.3 Capturing NP Ambiguities in a Structure:

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.

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 formulate 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 a 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.

At the start of the previous section we 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 where 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 to the query based on occurrences in reldocs and non-occurrence in nonrels ... or using *a priori* statistical co-occurrence distributions, nearest neighbours, min/max spanning trees, etc ...

Massive query expansion (c.300 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, outside the context of a given query or document ... thesauri ... which may be domain-independent or domain-specific.

The largest initiative in this field is Cyc but this is ongoing and we 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 ...

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 as 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 highly 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 expansion for statistical IR, we (DCU) have derived hierarchical concept graphs from WordNet, weighted links 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 assignment.

Our work to date has tried some queries from TREC on WSJ data only, but not successful ... we have major problems of computational overhead which we are now addressing and we hope to try many-to-many query-document similarity measurement soon.

... though we are finding we are limited by wsd problems.

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 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 they 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 mandatory 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.

One 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 Copenhagen 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, RESEARCHED, 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.

9. Performance and Prospects for NLP in IR.

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 seems intuitive but *wsd* problems remain and hold this up from developing further

... indexing by phrases, based on NLP rather than statistical techniques, again seems intuitive, but no major leap in progress to date.

... 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, 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.

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.

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.

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.

IR is also good at using tools and resources from NLP.

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.

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 ...

- 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 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 !

- 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 than theories of how language words will help us with IR ... seems sensible !

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 Gbytes of text and reading at 180 wpm it would take 2.1 years 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.

That’s why we apply NLP to IR, but it is difficult.

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 Schäuble, Paraic Sheridan, Tomek Strzalkowski and many others.

Further Sources of Information ...

- Salton 1989 ... “The Analysis, Retrieval and Transformation of Information by Computer”, G. Salton, Addison-Wesley, 1989 ... a standard undergraduate / graduate textbook, the only General Purpose text in the field though there was an earlier 1983 text by Salton and McGill ... it, however, quite dated now, but has some of the basics.
- van Rijsbergen 1979 “Information Retrieval”, C.J. van Rijsbergen, Butterworths, 1979. ... out of print so don't bother ... very well cited but very specialist ... most of it is dated but Ch6 on probabilistic IR and the stuff on evaluation, remain seminal ... available on WWW, somewhere from <http://www.dcs.gla.ac.uk/ir>
- Bill Frakes and Ricardo Baeza-Yeates 1992 “Data Structures and Algorithms ??” ... a series of contributed chapters covering all aspects of IR but mainly implementation issues ... software contributed to volume available on the net.
- Computer Journal, June 1992 ... a special issue on information retrieval ... a variety of topics and a good snapshot of the breath of the field.
- 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)
- Conferences ...
 - The SIGIR Conference alternates annually both sides of the Atlantic
 - SIGIR95 ... Seattle, July, ACM Press
 - SIGIR94 ... Dublin, Springer
 - SIGIR93 ... Pittsburgh, ACM Press
 - SIGIR92 ... Copenhagen, ACM Press
 - SIGIR91 ... Chicago, ACM Press
 - SIGIR96 ... Zürich, August
 - SIGIR97 ... Princeton
 - SIGIR98 ... Australia (possibly)
 - The TREC conference is annual, probably up to TREC-6 ... open to participants and govt. agencies, but proceedings are published by NIST ... these are really impacting our field ... special issue of IP&M coming up.
- Summer School ... probably the Second European Summer School in Glasgow in September ... a week-long event, early September

- IRList electronic digest, ir@mailbase.ac.uk
- On the Web ... <http://www.acm.org/sigir> ?? ... with pointers to IR sites (Dortmund, UMass, Glasgow, Virginia Tech) and the older, smaller, test collections.

KJS and Stephen Robertson have (Dec 94) produced a Cambridge U TR ... only a few pages ... a beginners guide to how to implement a non-NLP, statistically based IR system ... i.e., what works in IR ...

The most popular IR research tool, most mature in terms of versions, incorporating evaluation routines, is SMART from Cornell, now on version 11 and publically available for a number of platforms

if you know and do NLP and want to try it out on IR and need a kickstart IR system, SMART is worth looking at, though OKAPI is supposed to become public soon

Trip Report on TREC-3
The 3rd Text REtrieval Conference

V 1.1

2-4 November, 1994

National Institute for Science and Technology
Gaithersburg, Washington D.C.

DISCLAIMER: This report is a trip report prepared by me personally and has no official standing with TREC, the TREC organisers, NIST or DARPA. It is a purely personal overview of my impressions and released for limited distribution only. No part may be quoted in any context or forum whatsoever without my expressed written permission.

© Alan Smeaton, 1994.

1. Introduction

In 1994 the National Institute of Standards in Washington D.C. organised and ran the third Text REtrieval Conference, TREC-3 sponsored by ARPA. This is part of ARPA's Human Language Technology Program which includes work in the areas of speech recognition (CSR), speech understanding (ATIS), machine translation (FAMT), text understanding (MUC as well as document detection (TREC). The aim of the series of TREC initiatives is to further research into large scale text retrieval by sponsoring a benchmarking exercise in which information retrieval systems varying from the prototype systems of research groups to full commercial products, run the same set of the user queries on the same text database, at the same time. Manual evaluation or relevance assessment of the top documents retrieved by participating systems for each query is then performed and a battery of evaluation tests are performed on the results submitted from each group.

As with previous TRECs there were two retrieval paradigms evaluated in TREC-3, ad hoc querying corresponding to standard user querying of a text database, and routing, corresponding to filtering of documents relevant to a static user profile. The documents are newspaper articles, news postings, journal abstracts, government documents, etc. Queries were formulated by NIST representatives who then made the relevance assessments and queries were formulated to have less than 200 relevant documents each. TREC participants were classified as either category A or category B; category A meaning that a group worked with a full 2 Gbytes of text (about 750,000 documents) while category B participants worked with about 500 Mbytes as their approaches to indexing/retrieval may have been more computationally demanding.

A total of more than 45 requests for participation were submitted from across the world and of these some were chosen for paper presentation at the TREC-3 workshop in November while the others were invited to present a poster. The only difference between paper and poster presentations as far as TREC-3 goes is simply the mechanism for presentation of results; all requests for participation were accepted and invited to use the data, compute results and to have a paper included in the proceedings. Of the initial applicants some dropped out due to underestimation of the size of the task or other problems. In this report we will report on paper and poster presentations treated equally.

2. The Participating Groups:

The groups participating in TREC-3 have been arranged below into a very personal and subjective categorisation, simply for ease of presentation. Some groups would fit neatly into more than one category, some would even fit into all categories, so the grouping is not definitive. The first letter in parentheses after the group name indicates whether the group is commercial (C) or academic/research (A) and the second letter indicates whether the group used the full TREC dataset of 2 Gbytes (A) or a reduced set of 550 Mbytes (B).

(a) Combination of Different Techniques:

Xerox PARC (CA): a proprietary approach to information retrieval; their routing used a neural network classifier based on LSI dimensionality reduction while their ad hoc was a

Cooper/Fuhr logistic regression. They indexed by words and word pairs co-occurring frequently in the corpus ... they used an intelligent segmentation or partitioning of long documents into coherent pages by measuring similarities between adjacent sentences and seeking the "valleys" in similarities, at approx 100 word boundaries. Found that this text tiling method improves results noticeably.

University of Massachusetts (AA): led by Bruce Croft and Jamie Callan, this is a TIPSTER group which was consistently one of the best-performing systems in previous TRECs. This time round they used their own POS tagger, identified sequences of nouns or Adj/Noun pairs as input to their PHRASE operator, broke long phrases into 2-word substrings and used PhraseFinder to create a concept space for the corpus based on proximities, computed document rankings based on a combinations of evidences and best matching passages (of 200 words), automatically produced queries were then manually checked for validity and sense and then a complex document-query similarity measure was used. In summary, they throw in anything that has worked and they stir it all around in a very precise blend and it works 'cos they are a TIPSTER group who have the resources to invest. The basic theme of the UMass group has not changed; highly structured queries with much query processing combining multiple sources of evidence

Siemens (CA): led by Ellen Voorhees who with previous TRECs concentrated on query expansion using the WordNet online thesaurus. This time the effort is solely on data fusion or combining results of more than one collection sub-set. Official results are not good because, in general one can profitably merge the results of more than one search, the peculiarities of the TREC collection make this hard to do consistently in TREC.

Rutgers University/Paul Kantor: This group has been developing the approach to data fusion or combining the results of more than one independent retrieval strategy into one ranking. In this TREC they combined statistically-based, Boolean and NLP-based approaches which represent really independent approaches, and their results are expected in the proceedings.

Université de Neuchâtel (Switzerland) (AB): This is the first time for this group in TREC and their approach is based on the construction of relevance links among documents from relevance assessments of previously run queries. Obviously depends on a large sample of relevance assessments being available and for the TREC category B data only 7% of documents have any relevance judgements, so there is a weakness here. When combined with statistically-based methods (the best of SMART) it only tweaked performance up a little bit.

Westlaw/West Publishing (CA): Howard Turtle was a co-developer of the Bayesian inference network approach which has proved so successful for UMass (the INQUERY system). Westlaw has now developed its own implementation called WIN which was tried out here and is designed for large collections. Divided documents into paragraphs and performed well.

Swiss Federal Institute of Technology (ETH) (Switzerland) (AA): led by Peter Schäuble, involved in TREC-2 and using Hidden Markov Models to perform passage retrieval on a vector spaced ranking to re-rank the initial ordering. This time they have developed a retrieval strategy which is a combination of statistically-based ranking (similar to SMART), a hypertext link based method where hypertext links are automatically created, and the passage retrieval from last year. As with most work on combining independent retrieval ranks, their results are good.

Queens College, New York (AA): previous involvement in TREC was poor because they were just tooling up but this time around they completed the experiment and with much improved results. Based on probabilistic indexing by words and 2-word phrases, identified a priori from text samples. They also break documents into sub-documents and combine multiple retrieval methods using spreading activation and soft boolean techniques.

(b) NLP-Based:

New York University (AA): led by Tomek Strzalkowski a previous TREC participating group who concentrate on NLP-based indexing by phrases or word pairs. Their main contribution in TREC-3 is scaling up to the full collection. They actually parse 3.3 Gbytes of text ! Structural ambiguities in text are not distinguished and all possible word pairs are generated. As with previous TRECs the NYU group's approach has not changed much, just the scale of it.

CLARITECH Corporation (CA): led by David Evans and were very good in TREC-2 in terms of effectiveness using an NLP-based phrase indexing method and planning to extend their work in the same direction. Unfortunately their results in TREC-3 were not as great as previously and their presentation was short on details.

Dublin City University (AB): This group was a first timer in TREC participation and used an approach based on indexing by structures derived from syntactic analysis. The results were poor and showed that using structure from syntax alone is insufficient for retrieval purposes. Could possibly be improved by a more judicious combination with statistically-based ranking.

(c) Vector Space or Probabilistic Model Based:

Cornell University (AA): led by Chris Buckley developer of the SMART system and extending their previous TREC work with massive query expansion and also local matching within global similarity measures. The SMART stuff is fairly standard at this stage and the TREC-3 contribution is the identification of phrases, identified as adjacent non-stopwords and incorporation of same into massive query expansion. The efficiency of the SMART implementation is commendable.

City University (AA) (UK): have participated in TREC before but have had problems in previous TRECs. Their approach is based on refining term weights for probabilistic weighting. Results this time around make them one of the best overall systems in TREC-3.

Univ. Calif. Berkeley (AA): led by Bill Cooper, a variant of the probabilistic model for information retrieval, also a previous TREC participant and presented more refinements of probabilistic weighting with some logistic regression. Performed reasonably well.

(d) Efficiency Issues:

CITRI (Australia) (AA): led by Ross Wilkinson and Justin Zobel, concentrating on the computational aspects of engineering information retrieval on large data collections, a previous TREC participant. Once again they are concerned with issues of scaling up to large collections by partitioning the TREC data into separate databases, each with their own index and a query is broadcast in WAIS-like fashion to a number of co-operating databases.

Australian National University (Australia) (AA): First time in TREC and using a massively parallel Fujitsu machine to implement Boyer-Moore string searching in a couple of seconds per query ... they have 8 Gbytes of RAM on their machine !

(e) Interactive Retrieval:

For the first time in TREC there was support for a specialist theme, in this case support for interactive retrieval. The motivation for interactive IR is an attempt to move TREC from rocket science to reality ! Interactive retrieval in this context means letting real users, search intermediaries, formulate queries as best they can after interacting with the system to discover term frequencies, etc. The 4 groups which took part in interactive retrieval did so under the routing paradigm and each reported extra aspects of their systems like the type and number of users, their prior experience with IR searching, time taken per query, etc.

Rutgers University/Nick Belkin (AA): led by Nick Belkin; Rutgers have been in TREC before but this is a departure into user-centred retrieval. They used UMass INQUERY, real users and searchers who actually used the non-boolean operators like synonym, fixed order proximity and unordered window, but they found that user habits often force searchers to mould new tools to these old habits.

City University London (UK) (AA): Led by Steve Robertson building on their previous TREC work with the OKAPI system and extending their probabilistic model to include a variety of term frequency information, in interactive mode with real searchers. Got about the same performance as other interactive groups.

VERITY (CA): these are the developers of the TOPIC commercial text retrieval system and although they don't say much, their system is well-described elsewhere. This was a bit of a one-man effort and results were average

University of Toronto (AB): led by Mark Chignell and a first time TREC participant and like VERITY, an almost solo effort, basically hand-constructed simple and or boolean queries ... results OK.

In concluding on interactive retrieval, it was expected that the results from the interactive groups would have set the ceiling or upperbounds for achievable retrieval but in practice, many of the best automatic processes bettered the results of the interactive groups. One reason for this could be that it was the first time this was tried in TREC (an argument used to explain poor performances of all first time participants) or it could be that the large amounts of training and or statistical data is exploited better by statistical than by human approaches or it could even be that what was actually done was not interactive querying but manual involvement in automatic processes. This is important for applying IR research to real situations.

(f) Miscellaneous/Novel Approaches:

Environmental Research Institute of Michigan (AA): ERIM has participated in TREC previously and once again concentrated on a weighted trigram approach. This time their efforts were concentrated on quad-grams and their performance was better than previously but short of the best systems.

University of Central Florida (AA): Jim Driscoll has been trying to complete TREC since it started but this time he seems to have got it right. From each topic is generated an EER diagram (yes !) which is extended into a text filter which slides through the text in a window of 155 words looking for occurrences of EE entities or their synonyms ... could be termed massive query expansion.

George Mason University (AA): Used a trigram algorithm implemented on a Teradata database machine. Their first time completing and didn't do so well.

Mead Data Central (CA): Mead are a commercial information provision company but in this TREC they used SMART to experiment with a variety of query reduction strategies and statistically-based implementation to show short queries in such a context, don't work. This was a surprising thing for them to try out and they got the results one would have expected.

Bellcore (AA): led by Sue Dumais and as with previous TRECs concentrating on Latent Semantic Indexing, a dimensionality reduction technique, but using SMART tokeniser, single terms and refining the TREC-2 method. Results good.

NEC (Japan) (CA): tried out a new dictionary-based stemmer and generated an inverted file for the index. For queries, they extracted the noun phrases using a 140K word dictionary and generated weighted boolean queries and computed a statistically-based similarity measure with ranking. Results are only fair.

Teknos/University of Minnesota (AA): Topics are turned into conceptual graphs and concepts/relations have recognition expressions which are boolean queries, implemented on top of Personal Librarian, so the high level abstract ideas about concept recognition degenerate into complex boolean queries.

TRW/Paracel (CA): a query generation workbench for automatic query formulation ... results too preliminary and incomplete to say anything.

In addition to the groups outlined above there were others who participated in the experiment but who, for a variety of reasons including lack of funds were not able to be present at the TREC-3 workshop but whose work will be described in the full proceedings.

3. Spanish in TREC-3

In TREC-3 the organisers facilitated the evaluation of retrieval techniques on Spanish texts, the first time a language other than English had been tried. Spanish was chosen because of the availability of a corpus of Spanish newspaper stories and the availability of Spanish-speaking relevance assessors in the Washington D.C. area. Many groups initially expressed

an interest in performing retrieval on Spanish texts but in the end there were only 4 groups who completed this exercise. As this was the first use of Spanish, the only retrieval was ad hoc retrieval, and there were 25 queries on 193 Mbytes of Spanish texts from a Mexican newspaper. The 4 groups and their approaches were:

Mass: University of Massachusetts, Amherst, developed a Spanish stemming algorithm based on Porter's English equivalent and used the INQUERY system. This group obtained the best performance figures for Spanish.

Cornell: also developed a stemming algorithm for Spanish, albeit a much cruder version than Mass. They then used this as input into the SMART system and performed a vector spaced retrieval algorithm.

Environmental Research Institute of Michigan: indexed documents and queries by quadgrams, 4-letter overlapping substrings and piggy-backed a statistical weighting strategy on top of that. As the unit of indexing was a quadgram which is less than a word, the results were poorer than Cornell's

Dublin City University: the DCU approach was to index queries and texts by overlapping substrings called trigrams and to perform matching based on weighting trigrams based on frequency of occurrence. The approach, like that of ERIM is language-independent yet trigram-based retrieval has not had much success in English. The results obtained by this group were the poorest of the Spanish participants as would have been expected

The Spanish retrieval in TREC-3 was really only a starting point. With only 4 groups participating there are very few relevance assessments actually made and the pool is quite small, so any results other than the official results submitted for evaluation, must be taken with a grain of salt. The real contribution of Spanish in TREC-3 is to provide a minimal training set for Spanish in TREC-3. The organisers at NIST also have English translations of Spanish queries, which opens the possibility of multi-lingual retrieval involving this data in some future work.

4. Conclusions

There are many differences between TREC-3 and previous TRECs. One obvious difference is that this time around, the top 200 documents for each run were manually assessed for relevance as opposed to the top 100 only. This, coupled with the fact that there were so many participating groups who completed runs, makes results more reliable. An interesting statistic from TREC-3 is that with 48 submitted runs for adhoc retrieval, there could have been 4800 documents in the pool for all runs, taking the top 100 rank positions per run whereas in fact there were only 1005 documents, a noticeable decrease over previous TRECs. This suggests that participating groups are getting better.

As would be expected, many of the groups participating in TREC for the first time struggled with the difficulties of engineering the retrieval exercise on time. This has been a feature of all TRECs though groups which struggle on their first time around tend to do better in subsequent TRECs. Also, because TREC does not fund participation in any way now, groups must find their own resources and this can be difficult.

With so many things happening in TREC it is impossible to draw any conclusions into an "executive summary". If you want to get a grasp of what is going on then you must wade into the results yourself to decide if NLP is useful, if weighting is useful, if query expansion is useful, because these are not the kind of questions which TREC sets out to answer. TREC promotes research, it is not vulgar competition. It delivers research results but these results are too complex and against the spirit to compare and generalise. However, there are some aspects worth mentioning in a conclusion:

- For the first time in TREC there was an experiment of trying interactive querying and surprisingly the results obtained by the interactive groups was poorer than some automatic processes. We don't know why but a start has been made and work will continue in this track in TREC-4.
- The results in TREC-3 are better, in terms of precision and recall, than in TREC-2 or TREC-1 and this could be due to the topics being easier or the systems being better, we do not know.
- The worldwide interest in TREC continues to grow and the influence of TREC in the information retrieval research community is significant. Apart from those officially participating in TREC there are many others who have the data and who use it for research.
- There is interest in techniques which combine the results of more than one type of retrieval, data fusion being an example. There is an acceptance that combining rankings from independent retrievals does lead to improvements overall.
- Passage retrieval has been identified as a hot topic, i.e. where in response to a query, a passage or location within a document is retrieved for a user, not just the whole document. This raises arguments about the size of the sub-document to be retrieved (fixed sized window or logical component ?) A workshop led by Gerald Salton concentrated on this by motivating the need for it and citing work underway at Cornell and elsewhere.
- There is still a good mix of commercial and academic participation in TREC. Commercial participants tend not to obtain the best of the results but that does not deter their participation.
- The overwhelming emphasis is still on effectiveness and not efficiency though there are a couple of groups (CITRI and Australian National University for example) who are interested in effectiveness issues.
- There is interest in multi-lingual IR. The participation for Spanish in TREC-3 was disappointing but once people saw that Spanish was achievable the interest in Spanish for TREC-4 is much stronger.
- TREC is where a great deal of the application of IR research is reported. With respect to the annual SIGIR conference and other similar events, everybody in TREC has actually engineered a system of some kind and completed some runs, so there is a concentration and focus and this adds an air of collective coherence to the event. TREC has moved the IR field on considerably and ranks with any IR event in terms of importance. It is not more or less important than a conference like SIGIR, it is just different.

5. TREC-4 and Beyond.

There will be a TREC-4 held during 1995. A call for participation will be distributed on Dec 1 1994 (email me for a copy) with applications for participation due back Jan 1 1995 and data will be distributed soon after that. Ad hoc queries will be distributed on June 1st with results due back to NIST on Aug 1. Deadlines for multilingual TREC-4 will be later.

In TREC-4 the experiment will take a somewhat "hub and spokes" approach. The main task will be adhoc and/or routing retrieval but in addition, a participating group may take part on one or more TREC tracks, or specialist themes. For each track a co-ordinator has been appointed. The TREC-4 tracks are:

- data corruption, caused by OCR for example, and the track will distribute an algorithm to corrupt data and see how this affects performance ... Paul Kantor to co-ordinate.
- collection merging, not data fusion, but running queries on different sub-collections (Wall Street\Journal, Ziff, etc) and merging (not combining) results ... Ellen Voorhees to co-ordinate.
- interactive retrieval, extending work done in TREC-3 ... Stephen Robertson to co-ordinate (ser@is.city.ac.uk)
- NLP, identifying overlaps in systems, and possibly sharing phrases identified automatically ... Jamie Callan to co-ordinate (callan@cs.mass.edu)
- multilingual IR, facilitating ad hoc querying, possibly once again in Spanish, but also possibly using a corpus of parallel (language-wise) documents ... Alan Smeaton to co-ordinate (asmeaton@compapp.dcu.ie).

We can expect TREC-4 to be bigger with more participating groups, more data, more queries, more special themes and an even bigger impact on IR research and development. If you are a researcher or developer in this field, then even though participation would cost you in terms of resources, the benefits of participation are there to be seen.

Alan F. Smeaton
School of Computer Applications
Dublin City University, Glasnevin
Dublin 9, IRELAND.

☎ +353 - 1 - 7045262, Fax +353 - 1 - 7045442 email asmeaton@compapp.dcu.ie

Alan Smeaton has been a member of the TREC program committee since it started and was a participant in TREC-3 for both English and Spanish ad hoc retrieval. The proceedings of the

TREC-3 conference will be published by NIST in Spring 1995. Proceedings of TREC-2 and TREC-1 are available from NIST (email harman@magi.ncsl.nist.gov for details). Recently, a special issue of *Information Processing and Management* was devoted to TREC.