Swedish Institute of Computer Science

Information Retrieval: Statistics and Linguistics

Jussi Karlgren

Information Retrieval: Statistics and Linguistics

Jussi Karlgren

A short introduction to
textual information retrieval.
2000

Swedish Institute of Computer Science
Human Machine Interaction and
Language Engineering Laboratory
Box 1263, S–164 29 KISTA
KISTA
Sweden

This text was typeset by the author in the Computer Modern font using LaTeX.

What This Is

This is a short introduction to basic concepts in information retrieval. It is written from a linguistic point of view, and argues towards more results from linguistics to be introduced in information retrieval system development. This is also the first part of my doctoral dissertation "Stylistic Experiments in Information Retrieval".

Contents

Chapter 1

Organizing a collection of documents

Organizing a document collection so that documents can be found easily is difficult, especially if more than one reader is expected to be able to use the collection. These first chapters give a brief overview of existing automatic methods for text indexing and retrieval — one widely used technology for organizing collections automatically or semi-automatically — and identify some directions for future research.

1.1 Manual vs automatic methods

The traditional way of organizing documents and books is sorting them physically in shelves after categories that have been predetermined. This generally works well, but finding the right balance between category generality and category specificity is difficult; the library client has to learn the categorization scheme; quite often it is difficult to determine what category a document belongs to; and quite often a document may rightly belong to several categories.

Some of these drawbacks can be remedied by installing an *index* to the document collection. Documents can be given several pointers using several methods and can thus be reached by any of several routes. *Indexing* is the practice of establishing correspondences between a set, possibly large and typically finite, of *index terms* or search terms and individual documents or sections thereof. Index terms are meant to indicate the topic or the content of the text: the set of terms is chosen to reflect the topical structure of the collection, such as it can be determined. Indexing is typically done by indexers — persons who read documents and assign index terms to them. Manual indexing is often both difficult and dull; it poses great demands on consistency from indexing session to indexing session and between different indexers. It is the sort of job which is a prime candidate for automatization.

Automating human performance is never trivial, even when the task at hand may seem repetitive and non-creative at first glance. Manual indexing by human indexers is a quite complex task, and difficult to emulate by computers. Manual indexers and abstractors are not consistent, much to the astonishment of documentation researchers. In fact, establishing a general purpose representation of a text's content is probably an impossible task: anticipating future uses of a document is difficult at best.

Typically manual indexing schemes control the indexing process by careful instructions and an established set of allowed index terms. This naturally reduces variation, but also limits the flexibility of the resulting searches: the trade-off between predictability and flexibility becomes a key issue. The idea of limiting semantic variation to a discrete set of well defined terms — an idea which crops up regularly in fields such as artificial intelligence or machine translation — is of course a dramatic simplification of human linguistic behavior. Natural use of human languages does not make use of definitions or semantic delimitations; finding an explicit definition in natural discourse "...is a symptom of malfunction." (Hans Karlgren, 1976)

By and large computerized indexing schemes have distanced themselves from their early goal of emulating human indexing performance to concentrating on what computers do well, namely working over large bodies of data. Where initially the main body of work in information retrieval research has been to develop methods to handle the relative poverty of data in reference databases, and title-only or abstract-only document bases, the focus has shifted to developing methods to cope with the abundance of data and dynamic nature of document databases today.

This is where the most noticeable methodological shift during the past forty years can be found. Systems today typically do not take the set of index terms to be predefined, but use the material they find in the texts themselves as the starting point: a shift from what sometimes is called *pre-coordinate* to *post-coordinate* indexing. This shift is accompanied by the shift from a set-theoretical view of document bases to a probabilistic view of retrieval: modern retrieval systems typically do not view retrieval as operations on a set of documents, with user requests as constraints on set membership, but instead rank documents for likelihood of relevance to the words or terms the reader has offered to the system, based on some probabilistic calculation. The indexes typically generated by present-day systems are geared towards fully automatic retrieval of full texts rather than a traditional print index which will be used for access to bibliographical data or card catalogs. A traditional print index naturally must be small enough to be useful for human users. Under the assumption that no human user ever will actually read the index terms, the number of index terms can be allowed to grow practically with no limit. This makes the task of indexing texts different from the task that earlier efforts worked on.

However, the field has not experienced major methodological and theoretical breakthroughs. Basically, the information retrieval systems of today work in an intuitively appealing simple way, using algorithms about forty years old. Most systems that are deployed for public use today are based on ideas that were known, established, and first explored empirically in the 1950's (Luhn, 1957; Luhn, 1958; Luhn, 1959). This should not be taken to mean that the actual retrieval services have not improved strikingly over the past forty years: early conjecture has been solidified into algorithms; algorithms based on early conjecture have been verified mathematically, tested on large corpora, and developed and enhanced since. Systems today can — largely thanks to better hardware — make better use of users to improve their performance. There are full texts available, the interfaces to the systems are faster and better designed, the processing speed is high enough to permit interactive search — interactive in the sense that the user can be expected to provide the continuity of the dialog process — and the computer literacy of the average reader has increased to the point where enough library clients can be expected to use a computer search system to search and find documents for such systems to be designed and deployed in most libraries in well-to-do neighborhoods.

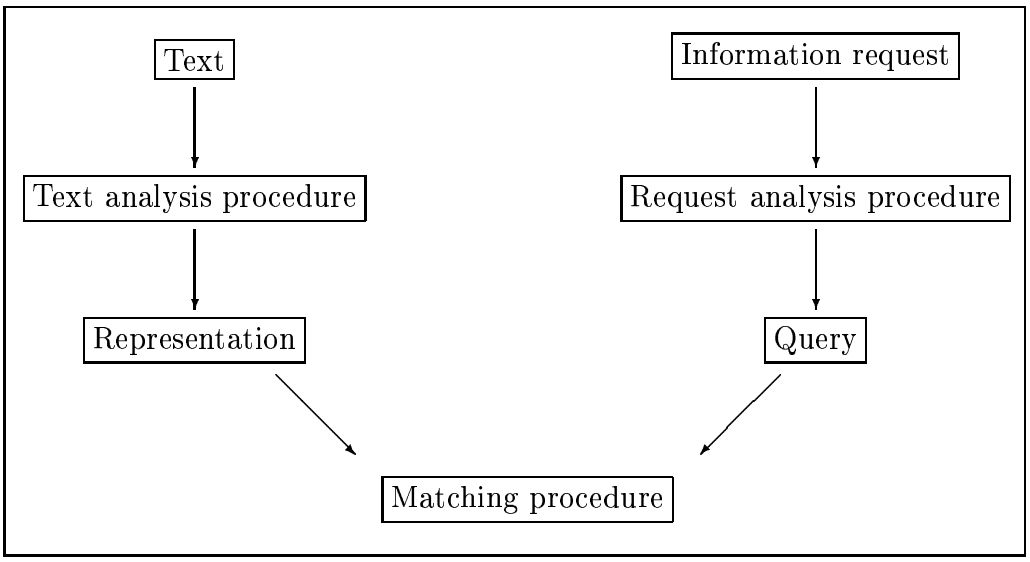

Figure 1.1: The Standard Model of Information Retrieval.

And this is a rapidly moving frontier. While the past decades have seen rapid development of full-text systems, it also has brought to renewed attention the value of manually provided indexes made up of few, well-edited terms. Manual indexing has not been supplanted by automatic full-text retrieval systems, but has a bright future ahead it, with tools to ensure consistency and raise productivity of human indexers.

The starting point of this text will be the design of typical information retrieval systems in use today, and it will go on to argue that certain shortcomings of these systems can best be addressed through the inclusion of more linguistic knowledge into the system — if the interface is competent to transmit this knowledge appropriately to the user.

1.2 Standard information retrieval systems

The standard model for information retrieval is roughly as shown in figure 1.1. There is a body of texts; information requests are put to a system which handles this body of texts; the texts are analysed by some form of analysis procedure to yield a non-textual representation of the same; the information requests are likewise analysed by an identical or similar procedure to yield a query. The two representations are then matched. The texts with the best matches are presented as potential information sources to fulfil the request.

In fact very little of this process is actually based on explicit knowledge of language. Typically both analysis procedures and matching procedure are performed using statistical methods. The role for linguistics or knowledge of language in general is usually assumed to be in improving the analysis of requests and texts; the representations are in some way assumed to be a-linguistic and amenable to pure formal manipulation. The point of the analysis operations is typically taken to be a) to reduce the amount of information in order to make the representations manageable — and the noise caused by

language and the freedom human languages afford their users are crucially important to reduce to that end — and b) straighten out the vagueness and indeterminacy inherent in natural language in order to facilitate matching.

This quite intuitive and in many ways appealing model hides the complexity of human language use from the matching procedure, which can then be addressed using formal methods. This is not entirely to the benefit of the enterprise. The very same mechanisms which make the matching complicated — the vagueness and indeterminacy of human language — are what makes human language work well as a communicative tool; awareness of this is typically abstracted out of the search process. The major difference between using an automated information retrieval system and consulting with a human information analyst is that the latter normally does not require the request to be transformed to some invariant and unambiguous representation; neither does the human analyst require the documents to be analyzed into such an representation. A human analyst not only copes with but utilizes the flexibility of information in human language: it is not an obstacle but a feature. "Vagueness may be the price that has to be paid in order to achieve the kind of gliding from one concept to another which is necessary for non-trivial retrieval" (Hans Karlgren, 1976): a seemingly unrelated text may contain valuable relevant parallels to a request.

The next few chapters will examine how information retrieval behavior is evaluated, how indexing schemes model and represent texts, how they elicit and model information needs, and how the dialog between reader and system is set up.

Chapter 2

Evaluating information retrieval

Information retrieval algorithms work with a well formalized and defined model of usage and utility. This has great benefits for the purposes of evaluating system usage, and information retrieval research has developed a well defined and well established set of evaluation tools. They are based on the notions of *precision* and *recall*. The figure from the previous section is repeated here (Figure 2.1), with a *result list* added to the process: a number of potentially useful documents will be presented to the reader in some way. Precision and recall are measured by examining how many relevant documents there are in the retrieved set.

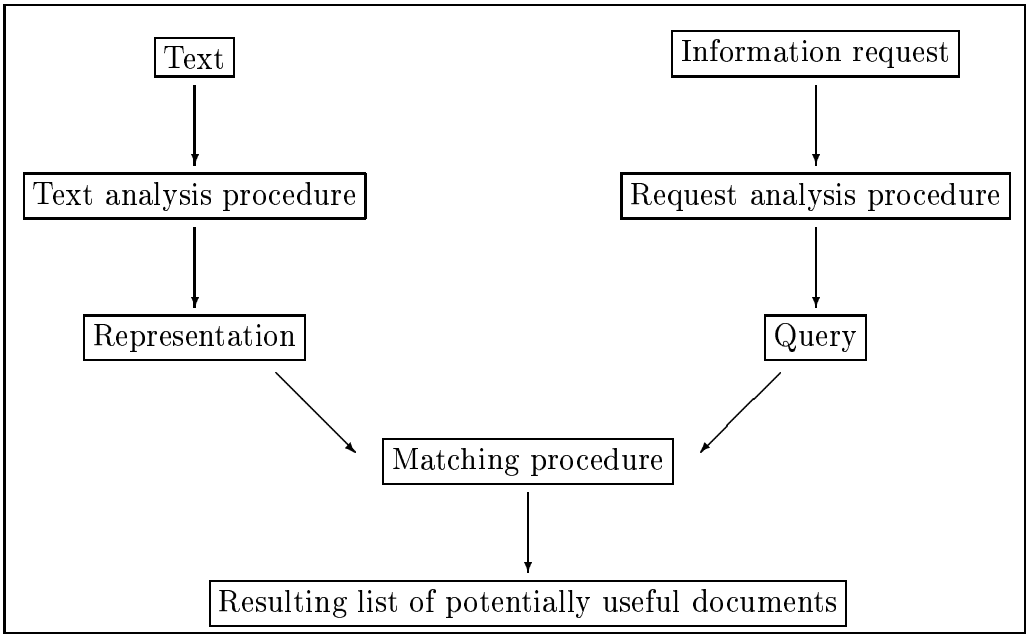

Figure 2.1: The Standard Model of Information Retrieval.

2.1 Recall and precision

Recall — How exhaustive is the search?

If one has a good estimate of how many relevant documents a document base con-

tains for some query, it is simple to calculate how many of the total set of relevant documents are found and retrieved by an algorithm. The ratio between the number of retrieved relevant documents and the total number of relevant documents, whether retrieved or not, is called *recall*.

Precision — How much garbage?
The retrieved set typically contains both relevant and non-relevant documents. The ratio between the number of relevant documents and the number of non-relevant documents in a retrieved set is called *precision*.

Combining precision and recall
Trivially, if an algorithm always retrieves all documents in a document base, it has one hundred per cent recall. However, it presumably has low precision. In this sense, precision and recall vary in an inverse relation. In many evaluations, precision is measured at a fixed number of retrieved documents: "precision at 25", e.g., gives a measure of how well an algorithm delivers at the top of the retrieved list. In others, recall and precision are plotted against each other: precision at a certain point of recall indicates how mych garbage readers have to wade through until they know they have found at least half of the interesting documents. In the TREC evaluations an "11-point" average measure is used, with precision measured at every 10 percent of recall: at 10 percent recall, at 20 percent recall, and so forth down to 100 percent recall, where all relevant documents are assumed to have been retrieved. The average precision at all those recall points is used as the total measure.

2.2 How good are the evaluation measures?

As evaluation measures precision and recall have several very attractive qualities. They are intuitively valid and can empirically be determined to be reliable. However, they suffer from some distinctive draw-backs. For the required calculations the evaluator must know how many relevant documents there are, how many documents there are in the base, how many documents are retrieved, how to weigh relevance to precision, how to determine what a query is, and how to judge relevance. All of these things can be done, but at risk of making the evaluation too ad-hoc and in itself irrelevant, as it were: information spaces seldom have a structure simple enough to be mapped to this sort of prototypical evaluative space.

Sampling
In general, we do not have a good picture of how many documents are relevant in a given document base. Unless we have a small experimental database under complete experimental control we must resort to sampling procedures.

Universe
Indeed, often we cannot determine what the 'entire document base' is: for instance, in the case of Internet retrieval, where the database is fluid and huge by most contemporary standards.

Iteration

A query is well defined experimentally, but what its counterpart in real life is less well defined. Users often cannot pose their information needs in succinct search terms but cycle through a number of iterations until the visible top few items in a retrieved set seem satisfactory. At what point should we evaluate the system? At each successive query? Or at the end of the session?

Retrieval

The retrieved set is not delimited in most probabilistic ranking systems. A list of several thousand documents is presumably not very useful to a human user. How many documents should we assume actually have been retrieved?

Precision vs recall

Averaging recall-precision trade offs in e.g. 11-point averages is common practice, but has the undesirable consequence to mask algorithm differences: some algorithms may do very well in high-precision searches and less well in high recall cases; some may do well in cases where there are very few documents to be found, others do better when the document base is saturated with material for the topic at hand.

Relevance and information need: what is "relevant"?

For some *tasks* relevance is very different to the assumptions underlying precision and recall. A user may be working on a court case or a patent application, and must have an exhaustive list of all documents in a database. For this task, recall is an important factor. But another user may need simply need an answer to a question — in which case the first document to come along may satisfy the entire need, and never mind the rest of the collection. And often there is no information need definable at all. If someone is scanning through a collection of family snapshots, the information need is satisfied when all pictures have been viewed or some more interesting task comes up, whichever comes first. A system for organizing and retrieving family photos will not be easy to evaluate per recall and precision.

There are characteristics of *documents* which make relevance judgments rather less than clear-cut. Firstly, documents may be of differing quality. Legal and medical advice can be found for free on the internet, without any quality assurance at all. Some information may be misguided and mistaken, and some may even be purposely misleading. In all, quality cannot easily be inserted into the binary distinction of relevance. Secondly, some documents supersede older documents. An old version of a manual has no relevance at all to a help query, if a newer version comes along; a refutation or counterclaim may lower the relevance of a referred document. Temporal or diachronic aspects of document collections can not be brought into relevance distinctions with any ease; documents can be *partially* relevant.

So, in summary, while precision and recall are very useful experimental concepts for testing algorithms and comparing algorithms to each other, their utility is less clear for measuring success for a system in real-world tasks for changing dynamic databases. Algorithms are but part of an information access system. Task definition, user preferences, document qualities, real-time factors, and cost and time constraints as well as other contextual factors all come into play for system evaluation.

The next chapters will informally describe standard algorithms and variations thereof, occasionally referring to precision and recall as standard measures of algorithm efficiency.

Chapter 3

Text as an object of study

Information retrieval technology is largely about text and the content of text. (Which of course is a limitation which may seem inappropriate in light of the large variety of information sources available to us.) In a research area which mainly concerns text there are numerous places where linguistic research results could profitably be applied, and numerous questions which well could be handed back to linguists for further study. However, in neither direction are these openings really utilized. This chapter will give an overview of what technology is being used today.

3.1 Words as indicators of document topic

The basic assumption of automatic indexing mechanisms is that the presence or absence of a word — or more generally, a *term*, which can be any word or combination of words — in a document is indicative of topic.

The central task in indexing is the choice of index term vocabulary. In the following the assumption is that the indexing vocabulary for the most part will be based on knowledge about the vocabulary of the texts, rather than a predetermined set. To understand the vocabulary of the texts, we need to understand how the language that the texts belong to work. Then the task reduces to: how can we pick relevant terms to describe a text, given that we know what terms are in it and how those terms are used elsewhere?

3.1.1 Analyze the document

The first steps to finding index terms automatically is to build a list of words in a text, and calculate their frequency of occurrence. The more frequent terms are considered more valuable in proportion to their observed frequencies. This design suggestion was first made by Hans Peter Luhn (Luhn, 1957; Luhn, 1959), and the measure is commonly called *term frequency* or, imaginatively, *tf*, for short. For this text, for instance, the list will be as shown in table 3.1. Typically, for each document the term weights are collected in a vector — a *term vector* — where each position in the vector represents a term, and each position holds the term weight for that document.

	Chapter 1 *Introduction*		Chapter 2 *Evaluation*		Chapter 3 *Text*
104	the	42	the	233	the
72	of	35	a	182	of
63	to	32	and	170	a
46	and	31	of	164	to
36	is	25	is	143	in
36	a	22	in	129	and
27	in	20	to	126	is
24	be	19	recall	70	be
18	terms	19	documents	64	text
16	retrieval	18	precision	62	that
16	information	15	for	60	for
16	indexing	14	be	58	or
	Chapter 4 *Queries*		Chapter 5 *Dialog*		Chapter 6 *Research*
117	the	72	the	101	the
55	of	61	of	100	of
54	to	40	to	83	and
50	and	39	a	68	to
45	in	35	in	55	in
41	a	32	and	43	a
40	is	31	information	40	is
33	documents	16	is	34	be
31	query	16	for	33	we
26	for	15	systems	31	for
23	as	13	interaction	27	are
21	terms	12	not	25	text

Figure 3.1: Frequency table of words in this text.

As a semantic representation, a term vector formed from a list such as the one in Table 3.1 is poor. An obvious improvement is to filter out certain words that seem to have little to do with topic. A list of such words, most often grammatical form words and other closed class words, is commonly called a *stop list* — an example can be seen in Table 3.2, and a resulting set of index terms in Table 3.3. Another route to improvement is to note — as Luhn does in his 1959 paper — that the most frequent words seldom are significant for this sort of enterprise, and that thus it might be possible to filter them out automatically, based on their frequency rather than their text-external or linguistic features.

the	a	and	that	one
it	two	may	could	such
next	just	half	both	of
to	in	for	which	its
		...		

Figure 3.2: Stoplist.

3.1.2 Knowledge about language

If we try to determine what terms in a document are significant for representing its content, we find that terms that are common in a document, but also common in all other documents, are less useful than others. The question is how *specific* a term is to a document, or how *uncommonly common* it is.

Collection frequency, *inverse document frequency*, or, again imaginatively, *idf*, is a measure of term specificity originally defined by Karen Sparck Jones (Sparck Jones, 1972). Idf is a function of K/d_i where K is some constant, typically dependent on N, the total number of documents in a collection, and d_i is the number of documents where a term i occurs — the *document frequency*. This measure gives high value to terms which occur in only a few documents. Used alone, it gives about as useful results as term frequency used alone.

There are several modifications of the *idf* measure. Paragraphs, instead of documents can be used as a basis, in order to model the fact that documents may not be homogenous (Lahtinen, 1998, e.g.); one may weight the measure in different ways based on the document properties.[1]

A potential problem with *idf* as a measure is that it is unclear what universe the document frequency should be calculated over. The calculation depends on having a total overview over all documents in an collection, and establishing what the general usage of a term is may be difficult, if not impossible. In some cases a collection is so

[1] As an example, Tokunaga and Iwayama suggest weighting the idf of a term for a given document by taking term frequency into account. Their measure — the *weighted idf* or *widf* is calculated as a function of df_i rather than d_i, where df_i is the frequencies of term i in the respective documents. Their experiments seem to indicate an improvement in performance — but they have sacrificed some of the probabilistic theoretical underpinnings of Sparck Jones´ original formulation. (Tokunaga and Iwayama, 1994)

	Chapter 1 *Intro*		Chapter 2 *Evaluation*		Chapter 3 *Text*
18	terms	19	recall	66	text
16	retrieval	19	documents	49	terms
16	information	18	precision	44	cite
16	indexing	10	information	40	document
16	human	9	relevance	38	information
12	systems	9	document	35	documents
11	texts	7	retrieved	33	words
11	language	7	relevant	30	texts
10	index	6	set	28	term
10	documents	6	retrieval	27	word
9	typically	6	query	24	retrieval
9	set	6	need	21	between
8	text	6	item	16	multi
8	document	6	base	16	language
	Chapter 4 *Queries*		Chapter 5 *Dialog*		Chapter 6 *Research*
33	documents	31	information	25	text
31	query	15	systems	22	texts
21	terms	13	interaction	18	need
19	boolean	11	user	18	language
16	information	11	model	18	information
16	document	11	access	17	retrieval
13	set	10	system	15	systems
12	systems	10	documents	15	analysis
12	search	9	tasks	13	words
12	retrieval	9	retrieval	13	word
10	use	6	set	12	section
10	cite	6	query	10	semantic
9	user	6	need	10	better
8	request	6	during	9	study

Figure 3.3: Frequency table of words in this text, filtered with stoplist.

well-defined that a collection internal *idf* is quite adequate; in others, where potential readers may not be aware of the collection setup or if the collection is very heterogenous, it may not. Table 3.4 shows d_i scores for words in this text, with the first six chapters viewed as individual documents: the scores range from one to six.

6	analysis		...
6	begin	1	adjacency
6	better	1	algebraic
6	document	1	assessment
6	documents	1	authoritativeness
6	general	1	book
6	information	1	bookcase
6	material	1	bursty
6	model	1	collocations
6	relevance	1	interactively
6	retrieval	1	interactivity
6	search	1	morphology
6	system	1	psycholinguistic
6	systems	1	textuality
6	terms	1	thesaurus
6	text	1	synonymous
	...	1	synonymy

Figure 3.4: Document frequencies for terms in this collection.

3.1.3 Combining tf and idf

There are various ways of combining term frequencies and inverse document freqeuencies, and empirical studies (Salton and Yang, 1973) show that the optimal combination may vary from collection to collection. Generally, *tf* is multiplied by *idf* to obtain a combined term weight. Alternatives would be for instance to entirely discard terms with *idf* below a set threshold — which seems to be slightly better for searches that require high precision. Both measures are usually smoothed by taking logarithms rather than the straight measure — or some similar simple transformation — to avoid dramatic effects of small numbers.

3.2 Beyond single word frequency counts

So far, the methods outlined above use knowledge of language only indirectly. But linguistic methods have obvious roles to play for index term selection. One reason to apply linguistic knowledge to index term selection is to provide multi-element terms effectively. This is assumed to provide gains in precision, by allowing finer grained distinctions between similar but non-identical multi-element terms with differing internal structure, or by establishing more elaborate relations between identified term elements. Thus, it would be possible to distinguish between representation-wise similar but non-identical documents.

	Chapter 1 *Intro*		Chapter 2 *Evaluation*		Chapter 3 *Text*
5	natural	6.3	recall	13.5	word
5	indexers	6	precision	13	idf
4	indexing	3.2	documents	11	text
4	flexibility	3	comes	10	tf
3.2	human	3	base	9.3	term
3	terms	2	sampling	9	technical
3	procedure	2	per	9	materials
3	manual	2	measures	8.2	terms
3	forty	2	lower	7.5	texts
3	body	2	internet	7	hearst
3	analyst	2	family	6.7	document
2.8	texts	2	exhaustive	6.6	words
	Chapter 4 *Queries*		Chapter 5 *Dialog*		Chapter 6 *Research*
9.5	boolean	5.2	information	6.5	word
6.2	query	5	visualization	5.5	texts
6	vectors	5	points	5	units
5.5	documents	4.3	interaction	5	described
5	spoerri	3	seeking	4.5	study
3.5	terms	3	interactivity	4.5	language
3.5	feedback	3	delivery	4.2	text
3	treated	2.5	systems	3.6	need
3	theoretically	2.5	support	3.5	linguists
3	limited	2.2	tasks	3.3	semantic
2.7	information	2.2	access	3	information
2.7	document	2	turns	3	weak

Figure 3.5: Frequency table of words in this text, filtered with both idf and stoplist.

Another reason is to conflate similar variants into one index term. This is assumed to provide gains in recall, by allowing more documents with only trivial diffences to be keyed by the same set of terms. The first has typically involved research in syntax, word dependencies, derivational morphology, and terminology; the second in inflectional morphology. So far, Sparck Jones finds that no conclusive improvement from using either technique has been established (Sparck Jones, 1999). All of these techniques can more or less be approximated using purely statistical methods.

3.2.1 Reducing the number of terms: Conflation

Morphological conflation

As can be seen in tables 3.1 and 3.3 the words "document" and "documents" both show up in the beginning of the list. The words "indexed" and "indexing" do not, and probably should — they show up further down in the list. Word form analysis, or *morphological analysis* would conflate these forms, and raise their combined weight.

Morphological analysis to identify morphological variants of a lexeme are normally implemented as *stemming* or simple suffix stripping. Porter describes a widely adopted and efficient context-sensitive stemming algorithm for English based on a suffix list (Porter, 1980). Alternatively the user can be encouraged not to enter full words but truncated forms.

The utility of stemming for English is debatable, (Harman, 1991) but its intuitive merits are good enough and its cost in processing quite low, so many systems make some effort in this direction. "This means that matches are not missed through trivial word variation as with singular/plural forms." (Robertson and Sparck Jones, 1996). English, of course, has an exceedingly spare morphology, with few morphological variants and tends not to form graphical compounds as often as other languages: both these characteristics would seem to decrease the utility of an elaborate morphological analysis for this language.

It is unfortunate for the generality of the results in the field that the research and business language of the world currently is English. Simple stemming is sufficient for English, but not for most other languages of the world. In comparison, where experiments on morphological analysis based normalization on material from languages other than English have been performed, they do provide improved results: how, and exactly what is useful depends on the language. (Slovene: (Popovic and Willett, 1992); Finnish: (Koskenniemi, 1996); Dutch: (Kraaij and Pohlmann, 1996); French: (Jacquemin and Tzoukermann, 1999), Swedish: (Hedlund *et al.*, 2000)).

Synonyms or semantic conflation

Another aspect of conflation is finding sets of synonyms — such as they may exist — or near synonyms, and equating them for search purposes. This is typically done with a static word list — a *thesaurus*[2] — based on compiled lexical knowledge. This approach is often developed into building larger networks of knowledge elements or senses with

[2]From the Greek *thisauros*: treasure, as it were.

extensive more or less domain-specific information. These knowledge resources tend to be cumbersome to standardize, build, and maintain, but provide distinct improvements in the field they are designed for.

Alternatively there are statistical techniques which reduce a large set of words to a smaller set of senses, most notably Latent Semantic Indexing. (Deerwester *et al.*, 1990) Latent Semantic Indexing works from the observation that a matrix of index terms by documents is sparse: most terms do not appear in most documents and the matrix will mostly contain nil values. This matrix can be reduced to a smaller, and thus denser, matrix by various mathematical techniques, e.g. singular value decomposition, which will result in conflation of terms with very similar distributions. The resulting entries are in some sense *senses* or meanings of words and terms: they group terms that have to do with each other, which presumably will be useful in a search situation. How much one wants to reduce the matrix is a question of how much information one is willing to sacrifice to gain the better recall given by the conflation.

3.2.2 Increasing the number of terms: Complex terms

Multi-word terms

Counting solitary words is fine, but the idea that lone words by themselves carry the topic of the text is one of the more obvious over-simplifications in the model so far. Indexing texts on ice cream on "ice" and "cream" is intuitively less useful than looking at the combination "ice cream". However, in experiments designed to test the usefulness of multi-word terms, any addition past single word indexing is cumbersome and expensive in memory and processing, while adding comparatively little to performance. In any case, the discriminatory power of single word terms is much stronger than that of any other information source (Strzalkowski *et al.*, 1996, e.g.). Finding multi word terms can be done by statistical techniques or by linguistically motivated techniques.

Collocations and multi-word technical terms

One way of expanding the search to words beyond single terms is simply tabulating words that occur adjacently in the text — *n-grams*. For instance, Magnus Merkel has implemented a tool for retrieving recurrent word sequences in text (Merkel *et al.*, 1994; Merkel, 1999).

Using more theoretical apparatus, other types of arbitrary and recurrent combinations in the text — *collocations* — can be recognized and tabulated as well. Frank Smadja has implemented a set of tools (Smadja, 1993) for retrieving collocations of various types using both statistical and lexical information; he identifies three major types of collocations: predicative relations such as hold between verbs and their objects in recurrent constructions, idiomatic noun phrases, and phrasal templates, where only a certain slot varies from instance to instance.

To extract collocations of the second type, Justeson and Katz have added lexical knowledge to simple statistics, and use it to extract *technical terms*. Technical terms are a specific category of words which behave almost like proper names. They cannot

easily be modified — their elements cannot be scrambled or replaced by more or less synonymous components, and they usually cannot be referred to with pronouns. Thus, the technical terms tend to stay invariant throughout a text, and between texts (Justeson and Katz, 1995).

Justeson's and Katz' appealingly simple algorithm to spot multi-word technical terms tabulates all multi-word sequences with a noun head from a text, and retains those that appear more than once. This method gives a surprisingly characteristic picture of a text topic, given that the text is of a technical or at least non-fiction nature. Their major point is well worth noting: the fact that a complex noun phrase is used more than once in identical form is evidence enough for its special quality as a technical term. It is *repetition*, not frequency, that is notable for technical terms.

Linguistic methods are often suggested for the purpose of extracting multi-word terms. First mention of linguistic methods — in this case, transformations — for normalizing syntactic variation in text is in the late fifties (Harris, 1958), and indeed, "The main modern rationale for linguistically motivated indexing is in capturing multi-element terms effectively." (Sparck Jones, 1999). The research in linguistically motivated indexing has typically taken statistically generated multi-word terms as a baseline and attempted to identify better terms. An example is Strzalkowski's work in trying to find linguistically motivated content word combinations through statistical analysis of word pairs and the dependence relation between them (Strzalkowski, 1994b). Strzalkowski has experimented using head modifier structures from fully parsed texts to extract index terms: this normalizes phrases such as "information retrieval" and "retrieval of information" to the same index representation.

However, it has been repeatedly shown that compound terms do not improve retrieval performance for English material more than marginally, (Fagan, 1989) and that the effort needed to implement and run linguistic methods in general is not worth the gain (Sparck Jones, 1999). On this note, Robertson and Sparck Jones discourage implementers from considering other than previously known multi-element terms: "Discovering, by inspection, what multi-word strings there are in a file is ... a very expensive enterprise. ... In general these elaborations are tricky to manage and not recommended for beginners." (Robertson and Sparck Jones, 1996).

3.3 Document length

As the term weight is defined in the *tf* component of the combined formula, it is heavily influenced by document length. A long document about a topic is likely to have more hits than a short one will for a relevant term; this may not reflect its greater likelihood of being relevant but simply its greater length.

Most algorithms in use introduce document length as a normalization factor of some sort, if the documents in the document base vary as regards length (Salton and Buckley, 1988). It is common to reduce each term weight in the document vector of a document d by dividing it with $\sqrt{(\sum_{i=1}^{N} tf(document_d, w_i))}$ or by some other factor derived from the document length in words or characters; the strength of the reduction may be controlled by a parameter which is set after experimenting with the collection at hand

(Robertson and Sparck Jones, 1996). This gives quite a strict normalization: it promotes short documents disproportionately, and in practice the effects of usually have to be damped somewhat, as long documents often turn out to be more interesting than what this normalization would assume would assume (Singhal *et al.*, 1995).

3.4 Texts are sometimes written — and read! — in languages other than English

As has been argued above, typological bias renders most discussions of the utility of linguistically motivated indexing moot. Non-linguistic and linguistic methods alike have been tested on English texts. English is a typologically special language. It relies more on word order than do most languages, and its morphology is more impoverished than most. These characteristics have effects not only on the linguistic methods but on the design of purely statistical algorithms as well. If linear order is important, collocations can be assumed to simpler than if long distance relations are marked by agreement markers of some sort. In general, it must be assumed necessary to perform new sets of experiments on each language a retrieval system is moved to, to ascertain that the mechanisms employed indeed give satisfactory results in the new language.

Moreover, texts written in a language the reader does not comprehend risk being ignored for the wrong reasons. While fully automatic general purpose high quality machine translation remains a seemingly attainable but in reality elusive and perpetually distant goal for ever new generations of language engineers and computational linguists, special purpose translation machinery already does show promise of usefulness. Especially if the distinctions between crude, raw, and skim-only-translations (Hans Karlgren, 1987) are made clear to system providers, we may expect to be able to peruse texts usefully in languages we do not master — if we can find them.

And more distressingly, texts written in a language the reader *does* comprehend risk being ignored if readers specify their information need in English rather than in any of the other languages they know. This of course may lead to a vicious cycle of such materials being made available or produced with less enthusiasm than materials in English.

Mechanisms to handle *multi-lingual retrieval* — i.e. retrieval of texts in several languages, and *cross-lingual retrieval* — i.e. retrieval of texts in another language than the query, are currently being designed and tested with some success. Such systems can be built using several different methods. The query itself may be translated. The document representations may be translated. Or the document representations and queries may both be represented in a common language: recent experiments use Latent Semantic Indexing, presented above, for that purpose (Dumais *et al.*, 1997).

3.5 Structured text and structuring text

3.5.1 Text is more than a bag of words

Text is more than the set of words in it, and specifically, it is more than a plain sequence of words. While texts at first glance are one-dimensional entities, in that linguistic objects such as syllables, words, and clauses *follow* each other in a strict sequence, *relations* between referents, terms, words, entities, subtexts, segments, clauses, paragraphs — or whatever other type of thing one wishes to postulate as suitable items of study — are present in the text, not constrained by local adjacency in the string or sound pattern, and can range quite far over the length of the text. This gives texts a fractal nature of sorts, a character of reaching beyond the one dimension the string affords it. Discovering these relations is largely what text understanding is about. A series of different techniques try to organize the text material into chunks larger than terms. The relations between textual items can take a number of forms, much dependent on the form of analysis chosen.

For text retrieval the standard model presented in Figure 1.1 and discussed so far has appealingly simple and intuitively understandable properties. Find out what a document is about, and find out what the user wants, and match the two; a document is about what is mentioned in it; what is mentioned is mentioned using words; count and tabulate the words. This is easy to understand and to implement.[3] But this simple model has its faults.

3.5.2 Word distributions are bursty

For instance, most statistical approaches assume words appear more or less randomly in a text, in a Poisson-style distribution, independently of each other and previous occurrences. This is naturally a gross simplification: words appear in a text not in a memoryless distribution but following a pattern governed by the textual topic progression and communicative conventions (Hans Karlgren, 1975; Katz, 1996). If text segments more likely to be topically pertinent are chosen and terms within them weighted up as compared to terms from other sections this weighting would reflect the topical make-up of the text better than a non-progressional model. These following sections cover some existing techniques potentially useful for this, such as summarization and text segmentation.

3.5.3 Fielded search

Some materials are structured to begin with. A search in a database for the yellow pages, for instance, will naturally make use of free text search as well as separate searches for company names or addresses in separate fields. A search in a press archive will naturally allow (or should allow) for searching in the byline and date fields separately from searching in the text itself.

[3] From personal experience I know that a class of computer science students can be taught to understand, appreciate and implement a working information retrieval system from scratch in less than one day.

3.5.4 Text segmentation

But most materials are not organized beforehand. It is to some degree possible to assume a structure for textual material which is not explicitly organized. Texts can be reasonably reliably split into structural and topical segments: a text may consist of several subtopics in sequence. Typically, such analysis is done without regard to likely search requests under the assumption that there is a structure which is possible to chart by inspection of texts as they are. To some degree this is true, although for instance in text segmentation tasks human subjects do not always agree on where segment boundaries can be assigned[4]. Texts can be split up in subtopic segments based on word occurrences: if word frequencies shift noticeably from one stretch of stretch to another, it is reasonable to assume that there is an attendant shift of topic. (Hearst and Plaunt, 1993; Hearst, 1994a; Reynar, 1994; Salton and Allan, 1994; Hearst, 1997). This is the underlying assumption of most text segmentation algorithms.

3.5.5 Passage retrieval and question answering

Sparck Jones and Kay wrote in 1973 that "... there is little doubt that it is from this direction [fact retrieval or question answering] that many of the new ideas introduced into documentation over the next few years will come." (Sparck Jones and Kay, 1973) This promise seems from today's perspective not to have been fulfilled. The optimism of the early seventies for solving artificial intelligence and knowledge representation issues was clearly unfounded. It is clear that — similarly to text segmentation — passage retrieval is non-trivial even for humans, even when the data set is quite small.

However, there are some treads along the path to systematic fact retrieval that actually have proved both promising and useful. In general, the idea that a system can find information in text by extracting structures that originate from the information requests themselves is much more tractable than attempting to organize texts in anticipation of future requests.

As an example, algorithms for entity spotting, starting with person, place, organization name spotting, and date spotting to more general entity spotting functions achieve some degree of success (Strzalkowski and Wang, 1996, e.g.), and add noticeably to information retrieval performance when combined with less inventive single and multi word term information. Similarly, technical terms have a more rigid structure than other multi-word terms — rather similar to names, in their linguistic behavior, in fact — and can be picked out through pattern matching techniques augmented with lexical information from a part-of-speech lexicon (Justeson and Katz, 1995).

This type of technique can be extended quite far to perform *information extraction*. This is a technique which is a descendant of the original description of "scientific sublanguages" or specific varieties of language — both as regards lexicon and syntax — used in specific contexts by Zellig Harris (Harris, 1958). And in application, matching recurrent

[4] Passonneau and Litman found that subjects did agree; Hearst found they did, more or less, within a range. Passonneau and Litman used spoken material, and Hearst used written popular science texts. Most likely the richer information in the spoken mode accounted for the difference in results. (Passonneau and Litman, 1993; Hearst, 1994a; Hearst, 1994b)

patterns for certain predictable pieces of information can be done with a useful level of accuracy and speed, such as is demonstrated in the yearly Message Understanding Conferences. These techniques are typically based on stereotyped and relatively stable information requests and elaborate linguistic variant detection algorithms that are precompiled to simple pattern matches (Grishman, 1996, e.g.). If an information need is relatively general and persistent over time, an information extraction system can trawl the information space for documents that describe instances of the relation or event requested: an example could be a system for finding news items that describe air traffic accidents, identifying location, type of aircraft, date, number of casualties, airline, and possible causes of the accident. These system do not attempt "text understanding", but search for locally consistent expressions that fit the given pattern. That local expression is then analyzed relatively thoroughly, using efficient syntactic analysis and semantic combination rules tailored for the domain, pattern, and text type.

Real live semantic analysis is beyond the scope of automatic systems today, but systems for higher level textual analysis are already at the point where the inclusion of semantic knowledge such as precomputed general concept hierarchies such as Wordnet, or of well typed domain-specific selectional restrictions (Grishman and Sterling, 1990, e.g.) improves extraction results, and question answer systems typologize queries to understand what type of answer the query expects: person, place, date, etc. (Voorhees and Tice, 1999, e.g.)

3.5.6 Abstracting and summarization

A common problem in information retrieval is that there is a large number of documents which may be relevant and may be not, and that deciding which are which is time-consuming. For this purpose, automatic abstracting, summarization or gisting algorithms attempt to provide a compact version of a text. Most automatic abstraction algorithms work on the assumption that selecting a number of sentences from the text will provide a picture of the text topic progression (Luhn, 1958).

3.6 Other qualities of text

3.6.1 Text ecology

Texts are used, and often used systematically. When any certain text is read, certain other texts are likely to be read as a consequence, or to appear in its vicinity in some way. In addition to the *content* and the *form* of the documents retrieved, documents have a *context*, or *ecology*: texts are actually used by people for whom they are important. One way of utilizing knowledge about use, usage, and the social characteristics of text is to consider it when designing interfaces, to support different strategies of use, but this type of information can also be used directly in retrieval (Walker, 1981; Walker, 1991; Belkin, 1994). In addition, documents often hold references to other documents. Citation analysis is a well understood tool for organizing scientific materials — scientific material has explicit citations, quotes, and other pointers to other similar materials. Today, hyperlinks are common in materials published on the WWW, and using linking

information gives a guide for finding more information, and judging the authoritativeness and topical focus of materials that are linked together. In addition, hyperlinked materials in a sense form an amorphous whole: searching after material on the WWW involves finding *some* document with relatively pertinent data — a reasonable assumption is that further links can be found in the vicinity of a near hit.

Recommendation systems

In the last few years, interest in utilizing text usage as another indicator of text usefulness has resulted in a number of studies and indeed a number of implementations which recommend items to users based on their previous access habits. These types of system usually build on readers broadcasting or submitting their opinions or evaluations of documents or whatever items are under considerations, and these judgments are then weighed together to produce a prediction of how well a user is likely to like an article. (Brodda, 1990; Karlgren, 1990; Karlgren, 1994; Resnick *et al.*, 1994; Shardanand and Maes, 1995; Hill *et al.*, 1995)

This practice rests on two observations. Firstly, users often have a fair idea of what they have read, and they can relate their query to their own readership history; the situation and the request in information retrieval situations can often be formulated as a form of "I read *A Good Book* — I want more of the same" posed to a librarian or a colleague, or to a number of them.

Secondly, an ordinarily unorganized bookcase may self-organize — somewhat unsystematically — based on users' behavior. Interesting documents may be found next to each other. They are interesting because someone placed or left them there, and they are placed there because they have some relation to the original document. In fact, in a library or a bookstore, people around an interesting bookcase tend to be interesting people. You tend to be able to get good reading tips from them. Similarly, a good librarian will remember that a certain book tends to be read by a certain set of people, and another book by the same set of people, and that there may be a similarity between the books, even though they may not be catalogized together – as of course, they often will not be. Anyone who has tried to organize a bookcase by topic knows how many cases of unexpected category conflict one encounters.

These systems assume the existence of a user model, containing user judgments of some sort on documents. Whether the grades are acquired by the system by explicit user recommendation or observation of user behavior, the methods for *using* the information in them will be similar. The information in a set of simple user models can be used to compute a proximity measure between documents. The first step is to define *interest* as a relation between a user and a document. Then documents can be recommended to users based on the following statement, the **Recommendation Hypothesis**: If a user A is interested in documents K and L, and another user X is interested in K, it is likely that X will also like L. Or less formally: If users agree on a document they will agree on others as well.

Now, having defined recommendations, the question is how to use it. The likelihood of a recommendations being useful grows with the number of users that agree, and the number of documents they agree on. The whole point is to sum recommendations over all

users. The proximity from a document to another can be defined as a sum over all shared readers' interests in them. This sum can then be used to cluster documents. Note that as defined, the proximity measure does not need to be symmetrical: the proximity from document K to document L does not have to be equal to the proximity from document K to document L: this would correspond to the idea of some document superseding another, or the sequence of a series of interrelated documents.

3.6.2 Document style

Texts are so much more than just sets of words. Indeed, texts are more than just what they are about. Texts vary in many ways. Authors make choices when they write a text: they decide how to organize the material they have planned to introduce; they make choices between synonyms and syntactic constructions; they choose an intended audience for the text. Authors will make these choices in various ways and for various reasons: based on personal preferences, on their view of the reader, and on what they know and like about other similar texts.

A style is a consistent and distinguishable tendency to make some of these linguistic choices. Style is, on a surface level, very obviously detectable as the choice between items in a vocabulary, between types of syntactical constructions, between the various ways a text can be woven from the material it is made of. It is the information carried in a text when compared to other texts, or in a sense compared to language as a whole. This information — if seen or detected by the reader — will impart to the reader a predisposition to understand the meaning of text in certain ways. Or, more roughly put, style is the difference between two ways of saying the same thing.

So, the variation in a text or differences between texts that is not primarily topical, that has not to do with meaning, is stylistic. Naturally, demarcation of stylistic variation to topical variation is impossible. Certain meanings must or tend always to be expressed in certain styles: legal matters tend to be written in legal jargon rather than hexameter; car ownership statistics in journalistic or factual style. The impossibility of drawing a clean line between meaning and style has lead to much browbeating among stylisticians and linguists, and discussion about if there in fact are styles at all (Enkvist, 1973).

For the purposes of information retrieval, it is in fact all the more interesting to investigate the workings of stylistic variation if it is not completely divorced from topical variation. The purpose would be to find methods to *complement* topical information retrieval, not by improving topical recall nor necessarily topical precision, but by improving the likely subjective quality of the retrieved documents.

The variation that will be most useful to complement topic-based information retrieval is not variation between authors, nor between individual texts, but the consistent, predictable, and distinguishable variation that sets of text may show. The goal is to look for textual characteristics that are measurable quantities — *stylistic items* — and use them to posit variables to categorize or sort texts. The aim is to find *functional styles* (Vachek, 1975) that can be used to understand which *genre* a new or hitherto unknown text belongs to, and thus to predict the likelihood of the text being interesting to the reader, given that its topic has been determined correctly by topical analysis.

3.7 What can we do with information about text?

The preceding sections have given examples of how text can yield more information than word statistics, and argued for a deeper analysis of text. But can we make any use of this type of information? This depends crucially on how clients, users, or readers express their information needs, and how the system understands them. The next chapter describes how queries are processed.

Chapter 4

Understanding information needs: Requests and Queries

The preceding sections have concentrated on document analysis. Central to the enterprise of searching document databases is the information need, as experienced and understood by the user, and as communicated to the system. This representation of information need is then matched to the database in some manner. These sections will outline how the information need relates to a document database.

4.1 Typical query processing

In the standard model the information request, posed in English or as a set of search terms, is treated much like the documents in the database: it is analyzed on the basis of term occurrence, and is transformed into a vector of term weights — for which the term *query* typically is reserved — similar to the term vectors computed for the documents.

But documents and information requests are typically very different. Luhn's original model was for the searcher to compose an essay of approximately the same form as the sought for documents. The documents were undoubtedly assumed to be short, in the form of abstracts rather than full texts, which made this model seem practical; this of course no longer is true: documents can be quite long. And conversely as has been established both by informal observation and several formal experiments, information requests to information retrieval systems tend to be very short: the majority being three words or less (Rose and Stevens, 1996; Rose and Cutting, 1996). This gives very little purchase to most linguistically oriented methods, and one would wish to find methods which would encourage searchers to produce longer requests using more terms.[1]

Given that requests are of different length and different type than the target documents, the respective term vectors are usually treated differently. For instance, most systems use a binary frequency calculation for query terms: occurrence, rather than frequency, is used as a basis for weighting the query terms. (Salton and Buckley, 1988)

[1] For a very simple, yet successful attempt, see Karlgren and Franzén. (Karlgren and Franzén, 2000)

4.2 Matching queries and documents: search

Given a query and a document representation in vector forms, however the elements in the vector are computed, the question is how to match the two term vectors to find documents that fit the request. Most search engines put more effort into indexing to avoid complicated matching algorithms: a common method is to use the conventional scalar product of the two vectors, by simply adding the pairwise products of each element of the two vectors under consideration — thus obtaining a *similarity score* for each document as compared to the vector. Most systems use variations of this method. There are numerous variations to this scheme, with various adjustments made to fit the collection and the user population at hand. Most published models have a number of parameters to reflect these variations.

4.2.1 Boolean matching

A special case of the matching process outlined above is *Boolean* retrieval. Boolean retrieval is based on simple algebraic set theory, and uses binary weights for modeling the collection: documents are represented by occurrence of the words or terms in them, not frequency — as are the queries. The set of documents in the document database are examined to see which ones share terms with the query, exactly, with no regard to frequencies or other weighting functions — meaning, if the scalar product, mentioned above, of the two vectors is more than zero. This means that for each document, a binary decision is made: either the document is in the set, and matches the request, or the document is outside and does not. If a document matches a request it is presented to the user, if it does not, it will not be.

This works reasonably well if the set of index terms is limited by design through keywords or specially assigned index terms — or alternatively for document bases where the documents are short and concise: a list of literature abstracts or document titles, for instance. A document database of short items has as a side effect that the set of available terms is limited and the number of spurious terms in each document is low. This approach has several desirable characteristics: it is easy and efficient to implement and theoretically comprehensible through the well understood properties of set theory.

In cases where the set of index terms is limited or the search algorithm is Boolean, it is common to formalize the query formulation through a query language based on set theoretical expressions. Such search interfaces allow the combination of search terms in a logical structure, typically using Boolean operators such as AND, OR or NOT. Boolean query languages are most often, but not necessarily, connected to a Boolean search algorithm. As with the search algorithm, Boolean query languages have several desirable characteristics: above all, they have theoretically comprehensible through the well understood properties of set theory.

Boolean systems have definite drawbacks, however. As an example, Anselm Spoerri shows in an example how Boolean search can be difficult to work with (Spoerri, 1994). In an example database of several tens of thousands of items on computer science, he poses a query to retrieve items on "visual query languages for retrieving information and that consider human factors issues". He formulates a Boolean query:

```
1. (graphical OR visual)
2. information retrieval
3. query languages
4. human factors
```

This query can be understood and processed in two distinct ways: if AND is used to conjoin the four terms, the query is very restrictive. For this query, Spoerri retrieved one single document in the database. If OR is used, the query is not very specific: for this case, Spoerri obtained 19691 documents. This shows how Boolean search methods can have unexpected results in spite of its theoretically attractive predictability, and that the OR and AND of Boolean logic are deceptive in that they invite comparison with the "or" and "and" of everyday discourse. Most importantly, Boolean query formulation makes no concession to the built-in uncertainty and vagueness of human language use, but presuppose a well-defined indexing terminology and exceedingly simple dependences between terms. Spoerri goes on the define a graphical query language to overcome the rigidity of Boolean query languages while preserving some of the desirable qualities.

4.3 Boolean vs. probabilistic retrieval

Boolean systems are widely in use, especially for trained documentalists; for untrained users and wide ranging document collections, the Boolean approach has been largely abandoned in full-text retrieval systems — although the name *retrieval* has been retained for an activity which only in its extreme cases resembles retrieval — in favor of *probabilistic retrieval* approaches, which *rank* the retrieved documents by likelihood of providing relevant data for the resolution of the information request as expressed by the search terms. This in some sense provides a model of uncertainty, as well as obviates most of the need for a specific query language, at the price of lessening predictability, effectiveness, and, to some extent, expressiveness of the interface. In many ways this is a step towards using linguistic or at least language-oriented methods for processing documents and queries. But still, the idea is to perform as much analysis as possible in advance to simplify the matching process — the flexibility of which, as was argued in the introductory chapter, is not incidental but crucial to how language works.

The typical differences between prototypical probabilistic systems and Boolean systems are

1. the *weighting* of terms by assumed importance for a document's representation, e.g. by some of the methods presented in preceding chapters;
2. a matching algorithm which ranks documents by likelihood of relevance; and
3. relatively free query formulation mechanisms.

4.4 Query expansion and relevance feedback

Query and documents and different. In length: queries are short; documents long. In material qualities: documents are text; queries are mostly a small number of content words. Most mechanisms based on linguist analysis would require more textual material to model the query; all mechanisms based on statistical analysis are data-intensive and would deliver better results given more input.

One method to obtain more textual material to flesh out an overly concise information request is to use the first query as a seed, and then use the results from that retrieval as an initial first cycle.

4.4.1 Relevance feedback

A retrieval system can present the list of retrieved documents to the user, and have users note which documents seem useful at first glance. These relevant documents are then used to generate a new query. This can be done either automatically, or by presenting the user a set of terms culled from the set of documents confirmed relevant. These terms can then be individually included into the improved query.

Analogously, non-relevant documents can be discarded in the first iteration, and the terms in them weighted down in subsequent iterations. This technique — *relevance feedback* — was first formulated in the seventies (Robertson and Sparck Jones, 1976; Salton and Buckley, 1990) and is now in common use in many implementations. (Robertson and Sparck Jones, 1996)

Some researchers have doubted the usefulness of the technique – especially negative feedback, the exclusion of terms culled from discarded documents to increase precision, can lead to surprising results for the hapless user. However, in user tests it seems to work quite reasonably, and evidence seems to show that users do understand how relevance feedback works, appreciate the function, use it effectively to improve retrieval results, and use it better if afforded more control over its working. (Koenemann and Belkin, 1996)

Relevance feedback can be extended by clustering the retrieved documents in similarity sets, if the system has an efficient clustering algorithm. Cutting et al have implemented the *Scatter/Gather* algorithm for this purpose (Cutting *et al.*, 1992). Users can select not only single documents but entire clusters of documents which can then be further scattered and clustered in subsequent iterations.

4.4.2 Query expansion

A more automatic method to obtain more textual material than the original short request to the system gave is using the best matches from the retrieved set of documents as a seed, and then extracting terms from them to construct a new query. This in effect is relevance feedback without ever consulting the user, and is based on the hope that the first few documents indeed are relevant. (Strzalkowski *et al.*, 1998, e.g.) If the initial search is focused on precision this is a way of improving recall.

4.5 From queries to dialog

In conclusion, requests for information are in most systems treated more or less as documents. To get these requests to more resemble documents, systems should elicit more information from users, and encourage users to refine a submitted request, as indicated in the preceding section. In general, most systems today suffer from a narrow information flow from user to system. The next chapter will further discuss dialog with information access systems.

Chapter 5

Information access processes: dialog

In the preceding chapters it has been assumed that people will walk up to an information system and state their information need, and then wait for results. Information retrieval research tends to abstract away from the general aspects of interaction, and relatively little work has been put into understanding the special requirements information access tasks pose on interface design. But at several points in the preceding text, I have pointed out the need to look a little at further turns in the interaction: both in terms of relevance feedback and evaluation.

These questions are not new. Many questions from early information retrieval research are still valid and no less important today. In a paper from 1971, Bennett outlines some parameters concerning interactive information retrieval and interface design (Bennett, 1971):

- The characteristics of the searcher
- The conceptual framework presented to the searcher
- The role of feedback
- Operational characteristics such as command language, display, response etc.
- The constraints of the computer and IR techniques
- The effect of the IR system on the user interface for search
- Introduction of search facilities to the user
- The role of evaluation and feedback in the redesign cycle.

In a later paper he adds:

- The task to be performed,
- The user, and
- The information content

to the properties of information system design needs to take into account (Bennett, 1972). This list has not aged. In a recent workshop on interactive aspects of information retrieval, we used Bennett's paper as a starting point to pose the question "What have we learnt about interactive information retrieval in the past 25 years?" (Hansen and Karlgren, 1997). The answer seems to be that we have a greater understanding of almost all aspects of interactive information retrieval Bennett mentions, but that there is a lack of *method* in applying this knowledge to system design.

5.1 Goals and tasks

The prototypical information access scenario covers only parts of typical information needs: several different models and studies show how users behave in various ways depending on what *information access task* they are working on. Tasks can be postulated on any level of granularity: tasks may involve examining a certain item of information carefully or researching an entire topic superficially.

For instance, Belkin analyzes information seeking behavior into four prototypical tasks: searching for a known item, scanning through a list for potentially interesting material, reducing a large number of potentially interesting items to a smaller number, or examining a certain document to verify its qualities (Belkin, 1998).

But tasks are not primary. Tasks are but a way of accomplishing *goals*, of which we usually know little. We may ask the user to state their goal explicitly; we may infer some of the goals through analyzing user background and professional context (Hansen, 1997, e.g.). Mostly, the goal is unknown to the system.

In fact, quite often there will be no one single set of tasks leading to one goal: various tasks are chosen and taken on during the course of pursuing a goal, and the goal is likely to change during the session. Accessing information is more often than not a learning process — and especially with today's interactively organized networked information, where related information often is linked together.

The task set Belkin proposed is but one possible analysis — what is clear is that people have a wide variety of *information seeking strategies* which they employ during the course of a session. Building interactive information access systems with an eye on user strategies, and supporting multiple strategies simultaneously will enable people to find their tasks and shifts between them supported flexibly under way — which is more effective than constraining the user to follow a set path (Belkin, 1998).

5.2 Interaction models — beyond single queries

The rise in response speed and interactivity has given users the possibility of searching through a sequence of queries. The first systems did not make use of sequences — they take a sequence of queries not as a dialog but as a sequence of one-shot requests.

> "Like so many other kinds of self-'service', from supermarkets and filling stations upwards, [recent full text search systems] have been promoted by

salesmen who style the absence of service as quickness and the user's labour as automation". (Hans Karlgren and Donald Walker, 1980)

The model from the previous chapters, as shown in Figure 2.1, does not account for interaction with the material found. While it has lead to useful generalizations in the design of system innards, the systems are mostly not built to handle interaction with the retrieval results: each query is viewed as a separate session. In most information access processes, the users learn and develop new views of the material during search and retrieval. The material they scan through while trying to find a relevant item will give them a picture of how the information space is structured and what sorts of material they can expect to encounter.

Similarly to any type of dialog system, information retrieval systems should provide support for sessions, not only single requests. At a minimum, systems should support persistent and modifiable dialog objects: the previous turns in the discourse should not just go away from query to query. The recent formulation of a query, and the documents retrieved for it should be available for backtracking and for reference during the dialog.

5.2.1 Interaction points

Douglas Oard formulates a model of four interaction points between user and system: (1) posing an information request, (2) selecting documents from a set of them, (3) examining documents in detail, (4) ordering documents for delivery. (Oard and Resnik, 1999) These fit nicely into a description of a standard information retrieval system as shown in Figure 5.1. At each interaction point, the user may back up to refine or modify the original request.

Systems today typically follow this model from left to right, in that they always expect interaction to start with some form of specification of an information need, in the form of a small set of topical terms. This will give the user a ranked list of documents, which can be used either to select individual documents for further perusal, or discarded, in order to improve the original query.

Activity	**Specification**	**Visualization**	**Assessment**	**Delivery**
Example in today's systems	Input query terms	Inspect ranked list of documents	Scroll up and down in a document	Press "Print"
Future systems	Beyond words	Beyond ranking	Into the text	Beyond ASCII

Figure 5.1: Points of interaction with an information access system.

The previous chapter discussed specifying information need. Discussing the delivery of documents falls outside the scope of this text. Visualization and assessment are the foci of an increasing amount of research and development, and we can expect major advances in the next few generations of systems in these areas.

Presenting retrieval results in a list has been the standard way of communicating the contents of a document database to the user. This is in many ways a constraint on

understanding not only effects of the request the user posed, but also a limitation on the user view of the document space and interrelations between items in it. There are numerous visualization tools available that provide alternative views of retrieval results, but their presumed beneficial effects are hard to evaluate, the communicative conventions they make use of are ad hoc, and — on a more trivial level — they are usually not portable enough to be made available more than in specific situations. But most importantly, usually visualization tools in retrieval systems only communicate the same data that the list does in some slightly more accessible form. To be useful, visualization tools need to add substantially more value to the interaction.

Information retrieval is more than search. Reading and learning from documents modifies user behaviour during a session; depending on the type of search, users may be more or less prepared in advance. Tools to support information refinement and discovery — or indeed, to support *reading* — are an important future part of the information access framework.

5.2.2 Different types of information access

Oard's interaction model allows for more types of information access, as indeed it should. The interaction point model can assembled differently to model different types of system. It readily admits various types of information seeking behavior, and gives pointers to where interaction with a system can be understood as a subsystem of its own; Belkin's task oriented prototypical behaviors give an understanding of what the bottlenecks of information access systems can be.

For instance, it is not necessary to limit oneself to one entry point in the model. One may well envision cases where the starting point is a set of documents, inventively displayed, or segments or bits of one single document. And there is any number of interesting transition between different activities in the model to allow for better interactivity in an information access system. Today's systems mostly are not designed to support other than backtracking in the basic left-to-right model.

5.3 Research issues

This chapter touched on several obvious points where information retrieval systems need to develop further. In fact, much of the recent development in the field is in working on aspects of interactivity and result presentation: some of the obvious drawbacks in moving systems originally designed for information professionals to be used by the general public have to do with dialog design. The next chapter will recapitulate the research questions posed in previous chapters together with the ones posed here to formulate or sketch a research agenda for the field.

Chapter 6

Open research questions for linguistics in information access

The material in this section has been presented in an invited address to the 12th Nordic Computational Linguistics Conference in Trondheim, in December 1999.

To recapitulate from the previous chapters: besides all computationally generally interesting questions and questions specifically related to statistics and algorithm design, many research questions are specifically related to information access.

Systems based on the mechanisms outlined in the previous chapters can be improved. Not because of any obvious drawbacks in the mechanisms themselves: they provide consistent and stable results, with variation from system to system surprisingly small; the reason to continue work is that the stable results are not only consistent but consistently mediocre.

6.1 A role for linguistics

Accessing information is a primarily linguistic activity, and the documents available for retrieval in information access systems of today are for the most part texts. Linguists know about texts, and should know about discourse and dialog. Information access research should need linguists; linguistics should need the experience designing and deploying information access systems can afford them. And the application of statistics on large bodies of language data itself is a form of study of language. The information found is not in an explicit form, but if a result from practical systems is that two content words within a four word span from each other tend to form content-bearing associations where longer spans do not, this in itself ought to be interesting for the study of language. Finding generalizable topical clusters of documents irrespective of the language they are written in ought to be interesting for the study of language in itself. If retrieval of admittedly shoddy output from speech recognition systems works on average as well as retrieval of carefully proof-read texts this ought to be interesting in itself for the study of language. But results such as these are not appreciated by linguists or information scientists, for other than motivation for engineering efforts.

6.2 What is a document? — Two views

Information retrieval systems view documents as carriers of topical information, and hold words and terms as reasonable indicators of topic. The techniques used for analysis and and organization of document collections are primarily focused on word and term occurrence statistics.

Linguists believe linguistic expressions are composed of words which form clauses which form in turn text or discourse. Words have predictable, situation-, speaker- and topic-independent structure which can be described formally. Clauses have largely predictable, situation-, speaker- and topic-independent structure which can be described formally. Texts have largely unpredictable situation-, speaker-, and topic-*dependent* structure, which cannot be described formally.

Given that linguistics focuses on the theory of clause structure, and information retrieval on appearance of words and texts, the lack of contact between the fields may not be entirely surprising.

> "What is needed is a theory of language which makes it possible to make fairly gross statements about large units of text, and this is a matter on which linguistics has had very little to say." (Sparck Jones and Kay, 1973)

But an optimistic later quote by Karen Sparck Jones och Martin Kay seems to indicate that some progress is being made to broach the divide:

> "We take heart particularly from two facts: first, linguists are turning their attention more and more to larger units of discourse than the sentence, and second, on-line retrieval systems are likely to involve retrievable units smaller than traditional documents. We believe that the relevance of these fields to one another will become more apparent as the size of the text units they deal with becomes more commensurable." (Sparck Jones and Kay, 1976)

Research on larger units of language use such as texts, dialogs or discourse in general has not succeeded in providing generalizable results. The goal is less concrete: texts are not regular in the sense sentences are, and when formalization is attempted, it only succeeds in prototypical cases. Still, there is reason for optimism. With large amounts of texts available for automatic analysis of texts, linguists can test, discard, verify, and refine methods for large-scale analysis with the same efficiency clause-level analysis was performed earlier.

6.3 Linguistic methods in information retrieval

But so far, relatively few results have been evident. Morphological analysis is used both for variant conflation and compound term analysis. Syntactic analysis of entire sentences for the purpose of matching analyses to analyses of information requests have been experimented with for a long time (Sussenguth, 1964; Walker, 1969, e.g.), and syntactic

analysis is used to generalize over relations between entities in the text, to cluster term variants (cf. Figure 6.1) — but this latter sort of relation can, at least for English, be captured almost as well by extracting content terms within some short distance from each other.

Content analysis Analysis of content Systems that analyze content ... When content is being analyzed ...	`analysis+content`

Figure 6.1: Example of normalization of syntactic variants of multi-word terms.

So what is wrong with syntactic analysis? Why does it not help? Apparently structure on a clausal level is insufficient to clearly improve word based indexing schemes, and to back this up, there is some psycholinguistic evidence that the surface appearance of clauses is promptly forgotten after the clause has been analyzed and internalized.

The implicit semantic models in today's systems are based on single word occurrence and, occasionally, cooccurrence between words. This takes us a bit of the way, but not far enough for us to claim we have text understanding within reach. Words are besides vague both polysemous and incomplete: every word means several things and every thing can be expressed with any one of several words.

Semantic models for information access work either with lexical resources or knowledge bases or ontological networks of some sort; some schemes such as Latent Semantic Indexing work with clustering words based on occurrence patterns, and thus have a semantic model based on word occurrence, but on a higher level of abstraction. The lexically based models are brittle and do not age well, and share with the statistically based models the limitation that they model relatively atomic units of meaning, or senses — not relations, dependencies, actions or events: the stuff whereof discourse is made.

We need far better semantic models: better in the sense that they model language *use* rather than Language in the abstract. We need a better understanding of how meaning is negotiated in human language usage: fixed representations do not seem practical, and do not reflect observed human language usage. We need more exact study of inexact expression, of the *homeosemy*, (homeo- from Greek *homoios* similar) or near and close synonymy of expressions of human language (Hans Karlgren, 1976). This means we need to understand the temporality, saliency, and topicality of terms, relations, and grammatical elements — it means modeling the life cycle of terms in language, the life cycle of referents in discourse, and the connection between the two.

6.4 Multilinguality

Multi-lingual retrieval needs to be explored, for the obvious reason that interesting material may be available in the wrong language. Equally crucially, multi-lingual retrieval may improve retrieval in general by clearing the decks from the linguistic bias of results so far.

A strong case for continuing experiments on indexing schemes even in the face of the reasonably stable results obtained to date, is the fact that no substantive research has been performed on other than English text. English is a typologically special language in that it relies more on word order than on inflection than most other languages; this can be expected both to decrease the value of normalization through morphological analysis and the utility of linear precedence based statistical metrics. If we can expect words to appear adjacently in a predictable order with minimal variation from occurrence to occurrence, the systems we build will be very different than if we assume there are long range dependencies between haphazardly appearing words marked with agreement features.

6.5 How textuality could be utilized better

Texts do have structure — that much is evident. So far, little of this structure has been used explicitly for information retrieval. There are numbers of experiments that wait to be performed: if a text can be structured by some means, and its components indexed separately, such a composite index might well provide a richer picture of text topic than a simple list. Clause weighting approaches, topic-focus detection, foreground-background clause identification, summarization, and subtopic segmentation are all techniques available for experimentation: these show promise to perform differently from the single word and multi-word term frequency based indexing schemes detailed in the previous section.

Understanding more of why texts are texts rather than word containers, and why texts in important ways are more like pictures than dictionaries will give more depth to text analysis. The objective is some level of topical or semantic analysis, and from the discussion above and in the introduction, it seems abundantly clear this should be performed in interaction with the intended reader of the text. The reader or user is not a single one-shot question submission module — the user is accessing text for some reason, and this reason is not irrelevant for information retrieval purposes.

However, studying topical progression in a text is complex. Local effects — the distinction between given and new information in a clause, say — have been studied and partially formally described, but not robustly enough to be useful for predictive work, which is what information retrieval requires. "It is not easy to identify the topic and focus of a printed sentence, especially in such a language as English, where the surface word order is grammatically bound to a great extent." (Sgall, 1980) And later experiments cover — by author admission — prototypical cases only. (Hajičová *et al.*, 1995) There is a systematic problem in automatic text analysis in that text in itself is an entire semantic object, and has transcended much of the syntactically governed constraints that clause structure adheres to: surface cues give us only incidental traces of semantic linking of text. (Källgren, 1978; Källgren, 1979) But it is clear that human understanding of text hinges crucially on *expectations* and *hypotheses* on the part of the reader as well as the data itself as encoded in the text. It is not the structure of the text alone but of the *story* that leads readers right.

6.6 Other properties of texts

Further, texts have many kinds of properties besides being topical. Texts can be characterized, described and categorized in numerous ways. None of the criteria are independent of each other; some of them are weak and unreliable; all are not applicable to all items. Texts can be vague, abstract, legal, discussions, monologues, illustrated, difficult, short, repetitive, lucid, persuasive, focused, ungrammatical, schizophrenic, annoying, newsprint, offensive, obsolete, trendy — and so forth. Many of these types of characteristics are salient for readers, and *could* be used in retrieval contexts.

And when further modalities come into play, a more general view must be taken. For instance, experiments with audio database indexing involve not only a textual representation of the spoken data, but type of speech: dialog, monolog, etc. (Kimber *et al.*, 1995; Oard, 1997) Whatever dimensions of variation can be accepted as valid for an area or a set of texts, it is clear that a mono-modal text representation — whatever it is, and however well it is designed — simply will not be able to capture more than very simple characteristics of a text, and thus will ultimately constrain the utility of the matching functionality.

6.7 Reading — and who is the reader?

Given the variation in different types of knowledge about text, we understand that texts give many each in themselves weak signals to the reader. Still the reader judges texts quickly and efficiently. What is the connection between text and reading experience? What clues can we as system designers find and utilize? How can we merge several weak knowledge sources to make simple polar or near-polar judgments?

But the decisions made by users — even if they boil down to a polar "will read" or "won't read" are made by way of judgments on a relatively high level of abstraction. A reader will judge a text according to its authenticity, its suitability, its quality, as perceived.

We must formalize the subjective aspects of text categorization. And in practice, for system design, we need to investigate how to create and make use of several different indexing methods simultaneously.

And to understand reading better, we must have a way of understanding readers and users better. We cannot discuss reading in the abstract. In fact, general designs are likely not to be useful in building usable systems: tailoring systems to a specific set of goals will probably be better. But to get here we need to systematize the acquisition of knowledge about users, tasks, and goals. Readers come in many shapes, but they are not likely to be haphazard or disorganized. We will be able to understand trends and typical cases if we try.

6.8 Beyond ASCII

A picture says more than a thousand words. Building a system for accessing non-linguistic data will focus on several problems that must be addressed for textual and other linguistic systems as well. We do not have recourse to the short-cut words afford us. And this, in fact, may be to our benefit: the fact that text consists of readily identifiable words with obviously regular local dependencies to each other could be said to have lead language engineering up the impractical path of compositional semantics. Most likely, text retrieval and text access cannot be understood in any real way until more general questions, e.g. image access, have been understood well enough to have been posed.

The utility of the notion of homeosemy, introduced above, becomes all the more clearer if we raise our perspective beyond that of text retrieval to attempting retrieval of non-textual documents. It is the task of linguists to make obvious the connection between picture and text. No-one else will.

6.9 System evaluation

We need to study texts, systems, and users reading. The first and the last of those three study objects are arguably linguistic questions. The second may be.

We need to understand texts better. We obviously need *more* than the syntactic and semantic models of today can offer us. We need a *better* semantic theory than word occurrences. We also need to study more *global* textual phenomena rather than the local information organization and argument stucture. To this end we need good and reliable syntactic analysis — the sort of tools that are being made available today. While the immediate utility of these tools for information retrieval purposes is unclear, they are absolutely necessary for any further steps.

We need to understand aspects of language use through studies of the practice of human question answering outside laboratories rather than study of models of question answering in model worlds. We need to understand how users combine large amounts of data into a simple judgment of relevance. We need to understand the concept of relevance better.

And after providing various ways of enriching the representation of texts, and enriching our understanding of users and their needs, tasks, and goals, we must improve human-machine dialog, by building search systems that cope with such enriched representations.

References

Nicholas J. Belkin. 1994. "Design Principles for Electronic Textual Resources: Investigating Users and Uses of Scholarly Information". In Nicoletta Calzolari, Antonio Zampolli, and Martha Palmer, editors, *Current Issues in Computational Linguistics: In Honour of Donald Walker*. Kluwer, Dordrecht.

Nicholas J. Belkin. 1998. "An overview of results from Rutgers' investigations of interactive information retrieval". In *Visualizing Subject Access for 21st Century Information Resources, Proceedings of the 34th Annual Clinic on Library Applications of Data Processing*, pp. 45–62, Champaign-Urbana. University of Illinois School of Library and Information Science.

J L Bennett. 1971. "Interactive bibliographic search as a challenge to interface design". In Don Walker, editor, *Interactive bibliographic search: The User/Computer Interface*, pp. 1–16. AFIPS, Montvale, New Jersey.

J L Bennett. 1972. "The user interface in interactive systems". *AnnualReview of Information Science and Technology*, 7:159–196.

Benny Brodda. 1990. "Gimmie More O'That". In Peter Seipel, editor, *From Data Protection to Knowledge Machines*. Norstedts, Stockholm.

Douglass Cutting, D. Karger, Jan Pedersen, and John Tukey. 1992. "Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections". In *Proceedings of the 15th International Conference on Research and Development in Information Retrieval*, Copenhagen, Denmark, August. ACM SIGIR.

Nicoletta Calzolari, Antonio Zampolli, and Martha Palmer, editors. 1994. *Current Issues in Computational Linguistics: In Honour of Donald Walker*. Kluwer, Dordrecht.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. "Indexing by Latent Semantic Analysis.". *Journal of the American Society for Information Science*, 41:391–407.

Susan T. Dumais, Todd A. Letsche, Michael L. Littman, and Thomas K. Landauer. 1997. "Automatic Cross-Language Retrieval Using Latent Semantic Indexing". In *Notes from AAAI Spring Symposium on Cross-Language Text and Speech Retrieval*, Stanford University, California. AAAI.

Nils Erik Enkvist. 1973. *Linguistic Stylistics.* Mouton, The Hague, Netherlands.

Joel L. Fagan. 1989. "The Effectiveness of a Nonsyntactic Approach to Automatic Phrase Indexing for Document Retrieval". *Journal of the American Society for Information Science*, 40:115–132.

Ralph Grishman. 1995. "The NYU system for MUC-6, or Where's the Syntax?". In Beth Sundheim, editor, *Proceedings of the 6th Message Understanding Conference.*

Ralph Grishman and John Sterling. 1990. "Information Extraction and Semantic Constraints". In Hans Karlgren, editor, *Proceedings of the 13th International Conference on Computational Linguistics*, Helsinki, Finland, August. ICCL.

Preben Hansen. 1997. "An exploratory study of IR interaction for user interface design". In *Proceedings of the 20th International Conference on Research and Development in Information Retrieval*, Philadelphia, Pennsylvania, July. ACM SIGIR. Poster presentation, with abstract in proceedings. Long version available as: SICS Technical Report T97:03.

Zellig Harris. 1958. "Linguistic Transformations for Information Retrieval". In *Proceedings of the International Conference on Scientific Information*, Washington, DC.

Donna Harman. 1991. "How Effective is Suffixing?". *Journal of the American Society for Information Science*, 42:7–15.

Marti Hearst. 1994a. "Context And Structure In Automated Full-Text Information Access". Doctor of Philosophy Thesis, University of California, Berkeley, California.

Marti Hearst. 1994b. "Multi-Paragraph Segmentation of Expository Text". In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruses, New Mexico, June. ACL.

Marti Hearst. 1997. "TextTiling: Segmenting Text into Multi-Paragraph Subtopic Passages". *Computational Linguistics*, 2.

Preben Hansen and Jussi Karlgren. 1998. "Interactivity and Interaction". In Preben Hansen, editor, *Proceedings of Eighth DELOS Workshop on User Interfaces in Digital Libraries*, pp. 9–11, Långholmen. ERCIM.

Marti Hearst and Christian Plaunt. 1993. "Subtopic Structuring for Full-length Document Access". In *Proceedings of the 16th International Conference on Research and Development in Information Retrieval*, Pittsburgh, Pennsylvania, June. ACM SIGIR.

Turid Hedlund, Ari Pirkola, and Kalervo Järvelin. 2000. "Aspects of Swedish Morphology and Semantics from an Information Retrieval Perspective". Technical report, Department of Information Science, Tampere University, Tampere, Finland.

Will Hill, Larry Stead, Mark Rosenstein, and George Furnas. 1995. "Recommending and Evaluating Choices in a Virtual Community of Use". In *Human Factors in Computing Systems, CHI '95, Conference Proceedings*, pp. 194–201, Denver, Colorado, April. ACM.

Eva Hajičová, Hana Skoumalová, and Petr Sgall. 1995. "The Organization and Use of Information: An Automatic Procedure for Topic-Focus Identification". *Computational Linguistics*, 21:81–95.

Karen Sparck Jones and Martin Kay. 1973. *Linguistics and Information Science.* Academic Press, New York.

Karen Sparck Jones and Martin Kay. 1976. "Linguistics and Information Science: A Postscript". In Donald E. Walker, Hans Karlgren, and Martin Kay, editors, *Natural Language in Information Retrieval - Perspectives and Directions for Research.* Skriptor, Stockholm.

John S. Justeson and Slava M. Katz. 1995. "Technical Terminology: some linguistic properties and an algorithm for identification in text.". *Natural Language Engineering*, 1:9–27.

Karen Sparck Jones. 1972. "A statistical interpretation of term specificity and its application in retrieval". *Journal of Documentation*, 28:11–20.

Karen Sparck Jones. 1999. "What is the role of NLP in Text Retrieval?". In Tomek Strzalkowski, editor, *Natural Language Information Retrieval.* Kluwer, Boston.

Edward H. Sussenguth Jr. 1964. "The sentence matching program - graph". In Gerard Salton, editor, *Information Storage and Retrieval, Scientific report No. ISR-7 to the National Science Foundation.* The Computation Laboratory of Harvard University, Cambridge, Massachusetts.

Chris Jacquemin and Evelyn Tzoukermann. 1999. "NLP for term variant extraction: Synergy between Morphology, Lexicon and Syntax". In Tomek Strzalkowski, editor, *Natural Language Information Retrieval*. Kluwer, Boston.

Gunnel Källgren. 1978. "Deep Case, Text Surface, and Information Structure". *Nordic Journal of Linguistics*, 1:149–167.

Gunnel Källgren. 1979. *Innehåll i text, Ord och Stil 11*. Studentlitteratur, Lund.

Hans Karlgren. 1975. "Text Connexivity and Word Frequency Distribution". In Håkan Ringbom, editor, *Style and Text — Studies presented to Nils Erik Enkvist*. Skriptor, Stockholm, Sweden.

Hans Karlgren. 1976. "Homeosemy – On the Linguistics of Information Retrieval". In Donald E. Walker, Hans Karlgren, and Martin Kay, editors, *Natural Language in Information Retrieval - Perspectives and Directions for Research*. Skriptor, Stockholm.

Hans Karlgren. 1987. "Making Good Use of Poor Translations". *International Forum On Information And Documentation*, 12.

Jussi Karlgren. 1990. "An Algebra for Recommendations". Technical Report 179, Syslab, Department of Computer and System Sciences, Stockholm University, Stockholm, Sweden.

Jussi Karlgren. 1994. "Newsgroup Clustering Based On User Behavior — A Recommendation Algebra". Technical Report T94004, SICS, Stockholm, Sweden, February.

Slava Katz. 1996. "Distribution of content words and phrases in text and language modelling". *Natural Language Engineering*, 2:15–60.

Jürgen Koenemann and Nicholas J. Belkin. 1996. "A Case For Interaction: A Study of Interactive Information Retrieval Behavior and Effectiveness". In *Proceedings of the 19th International Conference on Research and Development in Information Retrieval*, pp. 205–212, Zürich, Switzerland, August. ACM SIGIR.

Jussi Karlgren and Kristofer Franzén. 2000. "Verbosity and Interface Design". Technical Report TR00002, SICS, Stockholm, Sweden, February.

Ferenc Kiefer, editor. 1980. *Questions and Answers*. Reidel, Dordrecht, Holland.

Kimmo Koskenniemi. 1996. "Finite state morphology in information retrieval". *Natural Language Engineering*, 2.

Wessel Kraaij and Renée Pohlmann. 1996. "Viewing Stemming as Recall Enhancement". In *Proceedings of the 19th International Conference on Research and Development in Information Retrieval*, Zürich, Switzerland, August. ACM SIGIR.

Hans Karlgren and Donald E Walker. 1980. "The Polytext System - A New Design for a Text Retrieval System". In Ferenc Kiefer, editor, *Questions and Answers*. Reidel, Dordrecht, Holland.

Donald Kimber, Lynn Wilcox, Francine Chen, and Thomas Moran. 1995. "Speaker Segmentation for Browsing Recorded Audio". In *Human Factors in Computing Systems, CHI '95, Conference Companion*, pp. 212–213, Denver, Colorado, April. ACM.

Timo Lahtinen. 1998. "The Use of an Index Term Corpus to Develop an Indexer". In *Proceedings of the Conference on Computational Linguistics in the Netherlands*.

Hans Peter Luhn. 1957. "A Statistical Approach to Mechanical Encoding and searching of Literary Information". *IBM Journal of Research and Development*, 1:309–317.

Hans Peter Luhn. 1958. "The Automatic Creation of Literature Abstracts". *IBM Journal of Research and Development*, 2:159–165.

Hans Peter Luhn. 1959. "Auto-Encoding of Documents for Information Retrieval Systems". In M. Boaz, editor, *Modern Trends in Documentation*, pp. 45–58. Pergamon Press, London.

Magnus Merkel. 1999. *Understanding and enhancing translation by parallel text processing*. Department of Computer and Information Science, Linköpings universitet, Linköping, Sweden.

Magnus Merkel, Bernt Nilsson, and Lars Ahrenberg. 1994. "A Phrase-Retrieval System Based on Recurrence". In *Proceedings of the Second Annual Workshop on Very Large Corpora*, Kyoto, Japan.

Douglas W. Oard. 1997. "Speech-Based Information Retrieval for Digital Libraries". In *Notes from AAAI Spring Symposium on Cross-Language Text and Speech Retrieval*, Stanford University, California. AAAI.

Douglas W. Oard and Philip Resnik. 1999. "Support for Interactive Document Selection in Cross-Language Information Retrieval". *Information Processing and Management*, 35:363–379.

Rebecca J. Passonneau and Diane J. Litman. 1993. "Intention-based Segmentation: Human Reliability and Correlation with Linguistic Cues". In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, Columbus, Ohio, June. ACL.

M. F. Porter. 1980. "An algorithm for suffix stripping". *Program*, 14:130–137.

Mirko Popovic and Peter Willett. 1992. "The effectiveness of stemming for natural-language access to Slovene textual data". *Journal of the American Society for Information Science*, 43:384–390.

Daniel E. Rose and Douglass R. Cutting. 1996. "Ranking for Usability: Enhanced Retrieval for Short Queries". Technical Report Apple Technical Report number 163, Apple Computer Inc., Cupertino, California.

Jeffrey C. Reynar. 1994. "An Automatic Method of Finding Topic Boundaries". In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruses, New Mexico, June. ACL.

Håkan Ringbom, editor. 1975. *Style and Text — Studies presented to Nils Erik Enkvist.* Skriptor, Stockholm, Sweden.

Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergström, and John Riedl. 1994. "GroupLens: An Open Architechture for Collaborative Filtering of Netnews". In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, Chapel Hill, North Carolina, October. ACM.

S. E. Robertson and Karen Sparck Jones. 1976. "Relevance Weighting of Search Terms". *Journal of the American Society for Information Science*, 27:129–146.

S. E. Robertson and Karen Sparck Jones. 1996. "Simple, proven approaches to text-retrieval". Technical Report 356, Computer Laboratory, University of Cambridge, Cambridge, England.

Daniel E. Rose and Curt Stevens. 1996. "V-Twin: A Lightweight Engine for Interactive Use". In Donna Harman, editor, *Proceedings of the 5th Text Retrieval Conference*, National Institute of Standards and Technology, Gaithersburg, Maryland, November.

Gerard Salton and James Allan. 1994. "Automatic Text Decomposition and Structuring". In *Proceedings of the 3rd International Conference on Intelligent Multimedia Information Retrieval Systems and Management*, pp. 6–20, New York, October.

Gerard Salton and Christopher Buckley. 1988. "Term-Weighting Approaches in Automatic Text Retrieval". *Information Processing and Management*, 24:513–523.

Gerard Salton and Christopher Buckley. 1990. "Improving retrieval performance by relevance feedback". *Journal of the American Society for Information Science*, 41 (4):288–297.

Petr Sgall. 1980. "Relevance of Topic and Focus for Automatic Question Answering". In Ferenc Kiefer, editor, *Questions and Answers*. Reidel, Dordrecht, Holland.

Tomek Strzalkowski, Louise Guthrie, Jussi Karlgren, Jim Leistensnider, Fang Lin, Jose Perez-Carballo, Troy Straszheim, Jin Wang, and Jon Wilding. 1996. "Natural Language Information Retrieval: TREC-5 Report". In Donna Harman, editor, *Proceedings of the 5th Text Retrieval Conference*, National Institute of Standards and Technology, Gaithersburg, Maryland, November.

Upendra Shardanand and Patti Maes. 1995. "Social Information Filtering: Algorithms for Automating "Word of Mouth"". In *Human Factors in Computing Systems, CHI '95, Conference Proceedings*, pp. 210–217, Denver, Colorado, April. ACM.

Frank Smadja. 1993. "Retrieving Collocations from Text: XTRACT". *Computational Linguistics*, 19:143–177.

Anselm Spoerri. 1994. "InfoCrystal: Integrating Exact and Partial Matching Approaches throug Visualization". In *Proceedings of the 3rd International Conference on Intelligent Multimedia Information Retrieval Systems and Management*, pp. 687–696, New York, October.

Amit Singhal, Gerard Salton, Mandar Mitra, and Chris Buckley. 1995. "Pivoted

Document Length Normalization". Technical Report TR95-1560, Department of Computer Science, Cornell University, Ithaca, New York.

Tomek Strzalkowski, Gees Stein, G. Bowden Wise, Jose Perez-Carballo, Pasi Tapanainen, Timo Järvinen, and Jussi Karlgren. 1998. "Natural Language Information Retrieval: TREC-7 Report". In Donna Harman, editor, *Proceedings of the 7th Text Retrieval Conference*, National Institute of Standards and Technology, Gaithersburg, Maryland, November.

Tomek Strzalkowski. 1994. "Building a Lexical Domain Map from Text Corpora". In *Proceedings of the 15th International Conference on Computational Linguistics*, Kyoto, Japan, August. ICCL.

Tomek Strzalkowski, editor. 1999. *Natural Language Information Retrieval.* Kluwer, Boston.

Tomek Strzalkowski and Jin Wang. 1996. "A Self-Learning Universal Concept Spotter". In *Proceedings of the 16th International Conference on Computational Linguistics*, København, Denmark, August. ICCL.

Gerard Salton and C. S. Yang. 1973. "On the Specification of Term Values in Automatic Indexing". *Journal of Documentation*, 29:351 – 372.

Takenobu Tokunaga and Makoto Iwayama. 1994. "Text categorization based on weighted inverse document frequency". Technical Report 94 TR0001, Department of Computer Science, Tokyo Institute of Technology, Tokyo, Japan.

Josef Vachek. 1975. "Some remarks on functional dialects of standard languages". In Håkan Ringbom, editor, *Style and Text — Studies presented to Nils Erik Enkvist.* Skriptor, Stockholm, Sweden.

Ellen Voorhees and Dawn Tice. 1999. "TREC-8 Question Answering Track". In Ellen Voorhees, editor, *Proceedings of the 8th Text Retrieval Conference*, National Institute of Standards and Technology, Gaithersburg, Maryland, November.

Donald E. Walker. 1969. "Computational linguistic techniques in an on-line system for textual analysis". In *Proceedings of the 3d International Conference on Computational Linguistics*, Sånga-Säby, Sweden, September. ICCL.

Donald E. Walker. 1981. "Contributions of Information Science, Computational Linguistics, and Artificial Intelligence". *Journal of the American Society for Information Science*, 32:347–363.

Donald E. Walker. 1991. "The Ecology of Language". In Nicoletta Calzolari, Antonio Zampolli, and Martha Palmer, editors, *Current Issues in Computational Linguistics: In Honour of Donald Walker*. Kluwer, Dordrecht.

Donald E. Walker, Hans Karlgren, and Martin Kay, editors. 1976. *Natural Language in Information Retrieval - Perspectives and Directions for Research*. Skriptor, Stockholm.

Nu har jag städat mitt skrivbord
och lagt papper i lådor och fack
och satt utkast och infall i pärmar
så nu är här så propert, ack,
nu är här så rent och prydligt
och allt har så snyggt stoppats ner
att redan idag är det tydligt
att jag aldrig hittar det mer

Alf Henrikson, ur *Anacka*, 1980.