SICS

Swedish Institute of Computer Science

Stylistic Experiments for Information Retrieval

Jussi Karlgren

Stylistic Experiments for Information Retrieval

Jussi Karlgren

A Dissertation submitted
for the Degree of Doctor of Philosophy
in Computational Linguistics

2000

Stockholm University
Department of Linguistics

Stockholm
Sweden

ISBN 91-7265-058-3

Swedish Institute of Computer Science
Human Machine Interaction and
Language Engineering Laboratory
Kista
Sweden

ISSN 1101-1335
ISRN SICS-D–26–SE
SICS Dissertation Series 26

Doctoral dissertation
Department of Linguistics
Stockholm University

ISBN 91-7265-058-3
This thesis was typeset by the author in the Computer Modern font using LaTeX.
Printed by Akademitryck AB, Edsbruk, Sweden, 2000.

Abstract

Information retrieval systems are built to handle texts as topical items: texts are tabulated by occurrence frequencies of content words in them, under the assumption that text topic is reasonably well modeled by content word occurrence. But texts have several interesting characteristics beyond topic. The experiments described in this text investigate *stylistic variation*. Roughly put, style is the difference between two ways of saying the same thing — and systematic stylistic variation can be used to characterize the *genre* of documents. These experiments investigate if stylistic information is distinguishable using simple language engineering methods, and if in that case this type of information can be used to improve information retrieval systems.

A first set of experiments shows that simple measures of stylistic variation can be used to distinguish genres from each other quite adequately; how well depends on what the genres in question are.

A second set of experiments evaluates the utility of stylistic measures for the purposes of information retrieval, to identify common characteristics of relevant and non-relevant documents. The conclusion is that the requests for information as typically expressed to retrieval systems are too terse and inspecific for non-topical information to improve retrieval results. Systems for information access need to be designed from the beginning to handle richer information about the texts and documents at hand: information about stylistic variation cannot easily be added to an existing system.

A third set of experiments explores how an interactive system can be designed to incorporate stylistic information in the interface between user and system. These experiments resulted in the design an interface for categorizing retrieval results by genre, and displaying the retrieval results using this categorization. This interface is integrated into a prototype for retrieving information from the World Wide Web.

Sammanfattning

Informationssökningssystem brukar byggas utifrån antagandet att dokument och texter bestäms av sitt innehåll: de konstrueras för att enkelt och effektivt kunna modellera sådan innehållslig variation. Men texter varierar inte bara efter innehåll, utan — bland mycket annat — efter stilsort. Texter innehåller stora mängder stilistiska markörer — små var för sig ointressanta eller nästan betydelselösa faktorer som en läsare ordnar ihop till en bedömning av hela textens stilsort eller genre och därigenom textens användbarhet för läsarens informationsbehov. Många stilistiska markörer är medvetna val av skribenten, som ordval mellan "ej", "icke" och "inte"; andra är hopvägningar av en mängd sådana val, som medelordlängd, interpunktion och medelsatslängd; andra återigen är mer bestämda av konventioner som ligger utanför skribentens individuella valmöjligheter, som genrespecifika textstrukturer, rubriksättning och andra formkrav. Denna text beskriver en serie experiment som gjorts för att se om stilistisk information är urskiljbar med enkla språkteknologiska medel och om isåfall sådan information kan användas för att förbättra informationssökningsystem.

De första experimenten visar att det går att särskilja textgenrer genom stilistiska skillnader i dem. Genom att göra enkla stilistiska mätningar som antal pronomen, antal siffror, meningslängd, ordlängd och väga ihop dem går det till exempel att skilja facktext från skönlitterär text och journalistisk text från vetenskaplig.

Nästa serie experiment tar avstamp i en årligen återkommande evaluering av informationssökningssystem. I evalueringen söker deltagande system besvara givna sökfrågor på ett givet stort material, och de resulterande dokumentlistornas relevans bedöms av mänskliga domare. Mina experiment försöker identifiera gemensamma drag hos de dokument som ratats av domarna i tidigare års evalueringsomgångar, och använda dessa drag för att sortera om listan, med de förmodat irrelevanta dokumenten sist. Resultatet visar att sådana typiska drag går att finna, men att de inte ger tillräckligt stabila resultat för att kunna förutsäga relevans hos tidigare osedda dokument. Slutsatsen är att system måste byggas från grunden för att kunna hantera och presentera stilistisk information så att de kan ge läsaren bättre stöd att bedöma sökresultat och enskilda dokument.

En sista serie experiment syftar till att undersöka hur ett interaktivt system kan utformas för att inkorporera stilistisk information i gränssnittet mellan användare och algoritm. Om systemet kan visa läsaren vilken sorts dokument en sökfråga hämtat kan användare bygga vidare på informationen i fortsatta sökningar. Experimenten utmynnade i ett gränssnitt som utformats för att utnyttja genrer som sätt att beskriva stilistisk variation, och som utgår ifrån en dialogsituation konstruerad för att uppmuntra stegvis förfining av sökfrågan utifrån kategorier av dokument. Detta gränssnitt finns integrerat i en prototyp för sökning av dokument på WWW.

Acknowledgments

This collection of experiments is to its author a work on language and human communication, and its manifestations in the act of finding interesting things to read in a digital collection. This may not be obvious to the reader: the engineering aspects are brought to the fore of the material. But the general direction of research is brought about by strongly held views on language, on the situatedness, conventionality, flexibility, and negotiability of language — views which surprisingly enough seem not to be common goods and self-evident.

These views and beliefs did not originate with me. They were given to me by my father Hans. We tended to agree about important things, and to both of us, human communication and different aspects of language were at the top of the list.

At an important stage in my life, I was exposed to another and in important ways very different view of language, and a systematic and elaborate approach to research work during years of discussion with Don Walker.

And when I got around to doing things of my own on my own, Gunnel Källgren's unfailing encouragement and enthusiasm for implementation and experimentation on the basis of hunches and inspired guesses, and systematic criticism of reliance on ideas of others helped me formulate my own experiments.

This text would not be here in this form without my father, Don, and Gunnel. None of them are here to see the results, argue with them, or do anything about the shortcomings I would trust them to spot. I miss and need them very much. And, if you knew any of them and intend to read further, you will notice that so does this text. (In fact, this text would not be here in this form if they *would* still be here!) I cannot thank them enough!

A great thank you to Björn Gambäck, my colleague since many years, who put great effort into proofreading an early version of this text when it still wasn't. He convinced me to turn it into a text. (After following almost all of his detailed proofreading advice I rewrote much of everything to be able to insert new typos and malapropisms in place of the ones he had me take out.) An equally great thank you to Kia Höök, who after originally hiring me at SICS, has provided both enthusiastic encouragement and constructive criticism along the past few years and most especially during the writing of this dissertation text — hopefully she now is convinced that there is some fun and excitement in information retrieval as well! And another great thank you to Ivan Bretan, who after a brief discussion immediately understood how to put the results into a practical framework. Thank you!

Several kind colleagues have read the text and provided me with appropriate, well-founded and often crucial suggestions for improvements: Kristofer Franzén, Preben Hansen, Gunnar Eriksson and Benny Brodda. Thank you!

This work was begun and completed in the intellectually rewarding atmosphere at the Human-Machine Interaction and Language Engineering laboratory at SICS: it began with a short experiment performed by Douglass Cutting and myself when he visited us. In the years thereafter, my colleagues included Martin Eineborg, Anna-Lena Ereback, Christer Samuelsson, Mats Wirén, and Annika Wærn who all have taken turns to variously encourage, goad, or threaten me to think about, begin, or complete this dissertation. Thank you!

This work would not have been completed without the support and encouragement from the part of SICS as an organization. This is thanks to our recent managing director Rainer Berling who actively supported this line of research, and all the other people at SICS in various ways have made my environment work: Peter Ehlin, Eva Gudmundsson, Kersti Hedman, Agneta Hellström, KPJ, Sirpa Juslenius, Lotta Jörsäter, Birgitta Klingenberg, Janusz Launberg, Jan Lundin, Lars Nilsander, and Marianne Rosenqvist, and most importantly, from an information science standpoint, Preben Hansen, and then Vicki Carleson, by providing most of the references. Thank you!

Most of the experiments presented in this dissertation have been done during my time as a visiting researcher at Proteus project at New York University, supported by the National Science Foundation under grant IRI-93-02615 and by the Defense Advanced Research Projects Agency under contract 94-F157900-000 from the Office of Research and Development. While there, I participated in the Text Retrieval Conference with a research team composed of members from General Electric, New York University, Lockheed Martin, and Rutgers University. The experiments have benefited greatly from discussions with my project colleagues: Louise Guthrie, Fang Lin, José Pérez-Carballo, Troy Straszheim, and Tomek Strzalkowski. Without the suggestions for improvements and corrections — some of which I heeded — given me by Ralph Grishman and Slava Katz at New York University, these experiments would not hold up at all. Thank you!

The final design of the Easify interface and the DropJaw prototype is due to the intellectual effort and hard work contributed by Johan Dewe, Anders Hallberg, and Niklas Wolkert who exhibited suitable proportions of enthusiasm and pragmatic creativity and built a beautiful prototype system from a handful of ideas. Thank you!

And my family: Cia, Erva, Kasper, Klas, Silja, and Teodor have all in their various ways been helpful — and done their best to divert my attention to things arguably more important!

Thank you, all!

Contents

Overview

This dissertation builds on work (both joint and original) in several projects. It is divided into five parts.

Part I. Information Retrieval: Statistics and Linguistics

This first part introduces the basic concepts of information retrieval technology and gives an outline of currently implemented methods. It makes no claims of being complete or exhaustive nor does it give formulæ or implementation guidelines — it is more an introduction to the field and an outline of approaches taken by most information retrieval systems to date, to motivate some of the research questions addressed in detail in the subsequent parts.

The first chapter introduces indexing as a special case of research in information access. The second chapter points out some well-known characteristics of the well-established evaluation measures precision and recall that have bearing on the experiments in this collection; the third chapter discusses models of text, and the relationship between information need and text; the fourth discusses queries and how they are modeled; the fifth discusses modeling information access dialog.

The section ends by setting out a research agenda for linguists in information retrieval. The issues set out for further study are numerous, but are generalized to the study of texts, systems, and users reading. The first and the last of those three study objects are arguably linguistic questions. The second may be. The claim is that we, meaning those of us who would like to design better information access systems, need better knowledge of texts, and of how they are used — but to make use of this information we need to work it into a usage context and a dialog which will enable users and systems to make sense of the knowledge in question.

The texts in this section have been used as instructional material and can be obtained together with companion experiments suitable for students.

Part II. Initial Experiments

The material in this chapter is essentially unchanged from the paper with the same title by Douglass Cutting and me presented to the 14th International Conference on Computational Linguistics (COLING-94), Kyoto, July 1994.

These first experiments in computational stylistic analysis were made by Douglass Cutting and me in the Spring of 1993. We took the Brown corpus, which is reasonably well genre categorized, and tried to work out how to recognize the various categories automatically. We drew up a list of variables of linguistic variation based on our intuitions, constrained to be such that we knew we could process easily: relative frequencies of various function words, measures of word and sentence length,

etc. Almost all turned out to vary significantly across the Brown categories. We combined them into categorization functions using discriminant analysis, a standard technique from descriptive statistics. The results were dependent on the genre palette chosen: the discriminant functions turned out to recognize some categories quite adequately, whereas others were difficult to distinguish. For instance, the Brown corpus had several subcategories under Fiction: Mystery, Romance, Science Fiction, and so forth, and it proved difficult to tease them apart using the data we had extracted. The more general categories were recognized excellently: distinguishing Fiction from Informative text was done with 96 per cent correct categorizations.

The results were quite encouraging, and we felt they warranted further study. Our idea was that this sort of mechanism could well prove useful as an additional filter on typical information retrieval systems in a cascaded model.

Part III. Stylistics and Relevance

In the Fall of 1995 and during 1996 I worked in the Proteus project at New York University, participating in the Text Retrieval Conference (TREC) with a project group consisting of Tomek Strzalkowski and Fang Lin from General Electric Centre for Research and Development; Troy Straszheim and myself from New York University Computer Science Department; Jose Carballo-Perez from Rutgers University School of Library and Information Science; and Louise Guthrie, Jim Leistensnider, and John Wilding from Lockheed Martin. During this time I ran several experiments to systematically investigate the utility of stylistic analysis for information retrieval, following the hypotheses Douglass Cutting and I had formulated after our first initial study.

The Text Retrieval Conference is organized as a competition: several participating systems engage in a common task using a document database of several hundred thousand documents and a set of predetermined queries. The systems attempt to retrieve those documents from the database which are relevant to the queries. The retrieved documents are assessed for relevance by the same human judges who originally formulated the queries.

Distinguishing relevant from non-relevant may not seem the obvious way of using stylistic statistics. The TREC tasks are by design generalized and void of extra-topical information, and the texts themselves are from fairly well-edited and homogenized textual sources such as the Wall Street Journal and the Federal Register — but I did find considerable variation and significant correlation between relevance and stylistics. The following four chapters describe some ways I made use of the variation I found.

Some of the material in this part has previously been published, most comprehensively as part of a chapter titled "Stylistic experiments in information retrieval" written by me in the volume "Natural Language Information Retrieval" edited by Tomek Strzalkowski (Karlgren, 1999). Some of the first experiments were presented as a poster at SIGIR'96 in Zürich (Karlgren, 1996a), and some of the TREC results as part of a paper titled "Natural Language Information Retrieval: TREC-5 report" by Tomek Strzalkowski, Louise Guthrie, myself, Jim Leistensnider, Fang Lin, Jose Perez-Carballo, Troy Straszheim, Jin Wang, and Jon Wilding presented to the fifth Text Retrieval Conference (Strzalkowski *et al.*, 1996). Continued experiments

were presented in a paper titled "Stylistics and Relevance" to the Second International Conference on New Methods in Language Processing (NeMLaP-2), Bilkent, September 1996 (Karlgren, 1996b).

Chapter 8. Stylistics and relevance

This report details a set of experiments I performed during 1995-96. I used the initial experiment performed on the Brown corpus as a basis, and added a large number of more sophisticated variables: syntactic complexity measures: parser performance data, e.g., textuality measures such as number of subtopics detected, and others. I used the Wall Street Journal portion of the TREC material as a corpus, and tried to detect variation between those articles which are found relevant for some query and those which are found not to be.

Chapter 9. Genres and relevance After finding that the difference between relevant and non-relevant documents is clear and significant. I relate this to (intentionally) sloppily defined genres, and find that even very ill-defined genres — ill-defined to the point of being practically random — improve categorization efforts. The conclusion is that genres are important as a basis for categorization.

Chapter 10. Non-parametric statistics

This (previously unpublished) discussion details some shortcomings of earlier application of standard statistical tools for research in the area of language processing. Other fields such as engineering or behavioral sciences with a longer tradition and better established methodology for statistical studies have categorized typical patterns of variation into parametrizable families of functions, such as normal, binomial or Poisson distributions, each with well-known characteristics and one or two parameters for fitting an observed variation to the familiar function.

Linguistic data cannot safely be assumed to follow specific known distributions simply because they have been observed to occur frequently in other fields. Language is different. I argue that until we know how, we should not uncritically adopt black-box tool kits solely because they work in other fields. This is easy enough to do, and I myself used discriminant analysis for the first experiments described in this collection, but here I claim that the use non-parametric measures rather than measures based on complex distributional assumptions would be more appropriate, until we find functions that fit our data. I describe a classification tool, C4.5, which I use to induce rules for distinguishing relevant from non-relevant documents.

Chapter 11. Stylistics and precision

Here I generate rules to distinguish relevant from nonrelevant, using the categorization tool and the stylistic analyses defined in previous chapters. These rules are used to rerank output from the retrieval machinery we use in TREC, in the hope that the overall result will be improved by the stylistic information.

The rules show promise, but in a general case test they fail: output from some queries improves, some other queries are not affected, and yet others fail. The variation from query to query is quite large expressed in average precision; the end result is slightly worse than without stylistic reranking.

The conclusion is that one needs more information on the queries themselves.

Chapter 12. Experiments in query categorization

I tried clustering the queries by query features — the presence of language that seemed to indicate number data, for instance — and by features found in the set of retrieved documents. Neither approach gave consistently useful results. It seems TREC queries give too little information to give purchase for this type of method. I conclude that we need to involve the user interactively with the system to utilize stylistic information.

Part IV. Stylistics and Interactive Systems

So using stylistics for relevance judgments seems to be rather iffy, and yielded no conclusive results in the TREC experimentation. It seems one might be better off enriching the representation of the document in the presentation of search results, rather than trying to fold an entire document space into a one-dimensional list, supposedly sorted by relevance.

I started this avenue of investigation while still visiting New York University. Together with Troy Straszheim, I designed IKSVIT, a visualization tool for displaying the results from a stylistic analysis of retrieval results. Later, after returning to SICS and Stockholm, I participated as a technical advisor, mainly on issues of text categorization, in the DropJaw project, which was initiated, supervised, and organized by Ivan Bretan. Anders Hallberg and Johan Dewe implemented the DropJaw system as part of their M Sc projects — thoroughly reported in their M Sc theses (Dewe, 1998; Hallberg, 1999), and Niklas Wolkert designed the Easify interface. DropJaw was designed to improve WWW information retrieval, and besides including stylistic analysis addressed some of the other general principles discussed in Chapter 6.

The experiments and designs here have been described in a series of publications. IKSVIT was briefly mentioned in a paper presented at the Second Conference on New Methods in Language Processing in Bilkent in September 1996 (Karlgren, 1996b) and in a paper written with Troy Straszheim in the proceedings of the 30th Hawaii International Conference on System Sciences, Maui, January 1997 (Karlgren and Straszheim, 1997); the empirical basis for the DropJaw project was presented to the 11th Nordic Conference on Computational Linguistics in January 1998 (Dewe *et al.*, 1998); Easify was presented in a paper by Ivan Bretan, Johan Dewe, Anders Hallberg, Niklas Wolkert and myself titled "Web-Specific Genre Visualization" presented to the Webnet conference in Orlando, Florida in November 1998 (Bretan *et al.*, 1998) and also as part of a paper by the same set of authors titled "Iterative Information Retrieval Using Fast Clustering and Usage-Specific Genres" presented to the tenth DELOS workshop on Digital Libraries in Stockholm in October 1998 (Karlgren *et al.*, 1998). The DropJaw project was awarded *Framsteget* in November 1998, an annual prize given for the most innovative Swedish research project in information technology by *Ny Teknik*, Sweden's leading engineering weekly.

Chapter 13. Visualizing stylistic variation

This chapter describes the implementation of IKSVIT, the first visualization interface used for genre analysis, and details some of the design decisions made.

The idea I had was that the variation of stylistic item statistics in texts is interesting in itself, and should be described as such; the system displays the document space with documents plotted on a 2-D surface where the axes are

linear combinations or individual stylistic items. This worked fine, and the display screens are colorful and seemingly friendly — the prototype system garnered some attention when demonstrated at workshops and conferences. The problem was for the hapless reader to understand what was going on. Variation along obscurely labeled dimensions was not immediately relatable to the information needs or the conceptual model of the user.

Chapter 14. Genres and visualization

IKSVIT was first boosted with genre indicators to aid orientation in the stylistic analysis space — the document plot points varied by genre: some documents were displayed as triangles, other as squares or balls.

Later, balloon help was added to IKSVIT, so that areas of the stylistic analysis space was labeled for likely genres. The conclusion was that the variation needed to be displayed using terms that relate to the information need of users: either genres or informative dimensions of variation.

Chapter 15. Assembling a balanced corpus from the internet

For some of the experimentation in the DropJaw project we found that we needed a corpus for training and testing our algorithms. To give useful results, we wanted the corpus to somewhat reflect the variation of the material at hand; to obtain such a corpus and to define the genre palette for the corpus, our methodology was based on user questionnaires. The paper gives a picture of what sort of trade-offs are necessary when matching user needs to technical feasibility.

Chapter 16. Stylistic analysis integrated into an interactive system

This chapter describes the Easify prototype interface designed by the DropJaw project. Easify brings together stylistics-based genre prediction with rapid document clustering in an attractive information retrieval interface. The prototype is designed to encourage and support iterative refinement of queries, and shows how richer document representation beyond simple term frequencies can be utilized in an integrated context.

The complexity of the previous interface is reduced, since no variation is displayed except in the form of document categorization. This reduces the amount of subjective decisions the reader is required to make at each point.

Part V. Concluding Remarks

Conclusions, Encouraging Observations, Optimistic Generalizations, and Shortcomings.

Part I

Information Retrieval: Statistics and Linguistics

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.

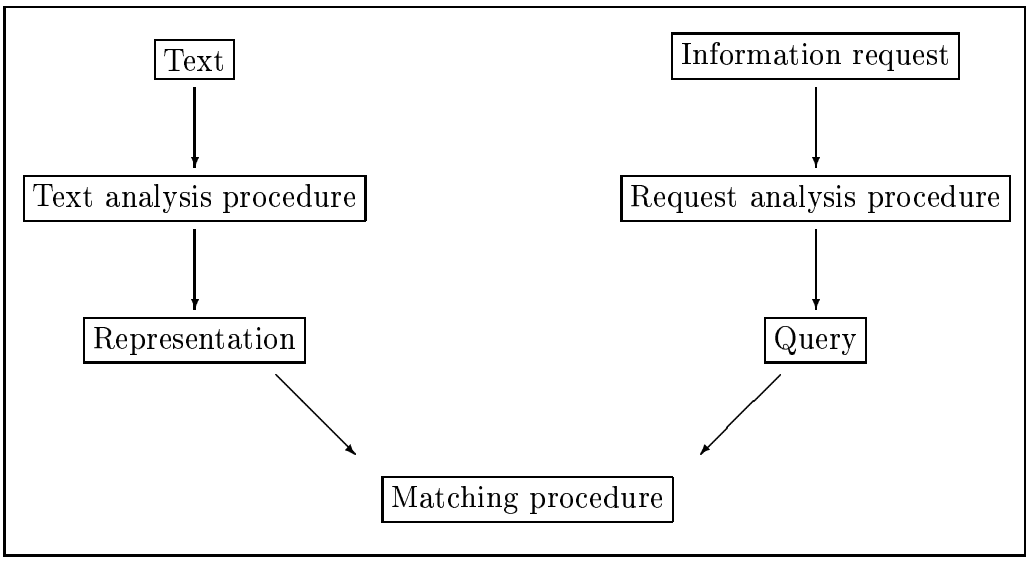

Figure 1.1: The Standard Model of Information Retrieval.

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.

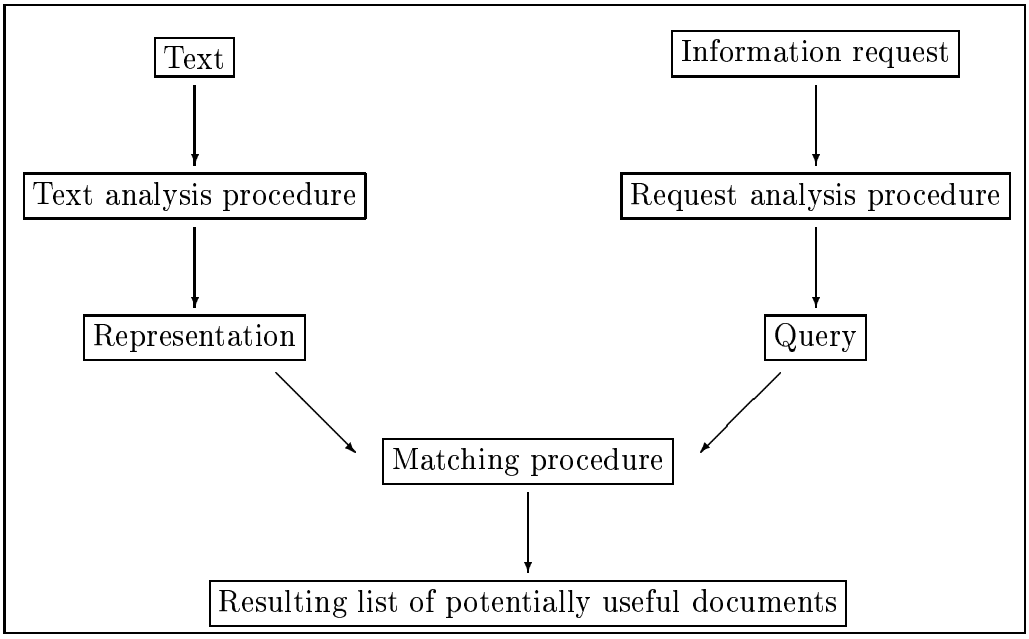

Figure 2.1: The Standard Model of Information Retrieval.

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

contains 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 number of non-retrieved relevant documents 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.

Part II

Initial Experiments

Chapter 7

Recognizing text genres with simple metrics using discriminant analysis

The material in this chapter has previously been published in a similarly titled paper by Douglass Cutting and me presented to Coling 1994 in Kyoto (Karlgren and Cutting, 1994). It describes work done by Douglass and me, who visited SICS in the Spring of 1993.

A simple method for categorizing texts into pre-determined text genre categories using the statistical standard technique of discriminant analysis is demonstrated with application to the Brown corpus. Discriminant analysis makes it possible to use a large number of parameters that may be specific for a certain corpus or information stream, and combine them into a small number of functions, with the parameters weighted on basis of how useful they are for discriminating text genres. An application to information retrieval is discussed.

7.1 Text types

There are different types of text. Texts "about" the same thing may be in differing genres, of different types, and of varying quality. Texts vary by several parameters, all relevant to the general information retrieval problem of matching reader needs and texts. Given this variation, in a text retrieval context the problems are (i) identifying genres, and (ii) choosing criteria to cluster texts of the same genre, with predictable precision and recall. This should not be confused with the issue of identifying topics, and choosing criteria that discriminate one topic from another. Although not orthogonal to genre-dependent variation, the variation that relates directly to content and topic is along other dimensions. Naturally, there is co-variance. Texts about certain topics may only occur in certain genres, and texts in certain genres may only treat certain topics; most topics do, however, occur in several genres, which is what interests us here.

Douglas Biber has studied text variation by several variables, and found that texts can be considered to vary along five dimensions. In his study, he clustered features according to covariance, to find underlying dimensions (Biber, 1989). We wish to find a method for identifying easily computable parameters that rapidly classify previously unseen texts

in general classes and along a small set — smaller than Biber's five — of dimensions, such that they can be explained in intuitively simple terms to the user of an information retrieval application. Our aim is to take a set of texts that *has* been selected by some sort of crude semantic analysis such as is typically performed by an information retrieval system and partition it *further* by genre or text type, and to display this variation as simply as possible in one or two dimensions.

7.2 Method

We start by using features similar to those first investigated by Biber, but we concentrate on those that are easy to compute assuming we have a part of speech tagger (Cutting *et al.*, 1992; Church 1988), such as third person pronoun occurrence rate as opposed to 'general hedges' (Biber, 1989). More and more of Biber's features will be available with the advent of more proficient analysis programs, for instance if complete surface syntactic parsing were performed before categorization (Voutilainen and Tapanainen, 1993).

We then use discriminant analysis, a technique from descriptive statistics. Discriminant analysis takes a set of precategorized individuals and data on their variation on a number of parameters, and works out a set *discriminant functions* which distinguishes between the groups. These functions can then be used to predict the category memberships of new individuals based on their parameter scores (Tatsuoka, 1971; Mustonen, 1965).

7.3 Evaluation

For data we used the Brown corpus of English text samples of uniform length, categorized in several categories as seen in Table 7.1. We ran discriminant analysis on the texts in the corpus using several different features as seen in Table 7.2. We used the SPSS system for statistical data analysis, which has as one of its features a complete discriminant analysis (SPSS, 1990). The discriminant function extracted from the data by the analysis is a linear combination of the parameters. To categorize a set into N categories $N-1$ functions need to be determined. However, if we are content with being able to plot all categories on a two-dimensional plane, which probably is what we want to do, for ease of exposition, we only use the two first and most significant functions.

2 categories

In the case of two categories, only one function is necessary for determining the category of an item. The function classified 478 cases correctly and misclassified 22, out of the 500 cases, as shown in Table 7.3 and Figure 7.4.

Experiment 1	Experiment 2	Experiment 3 (Brown categories)
I. Informative	1. Press	A. Press: reportage
		B. Press: editorial
		C. Press: reviews
	4. Misc	D. Religion
		E. Skills and Hobbies
		F. Popular Lore
		G. Belles Lettres, etc.
	2. Non-fiction	H. Gov. doc. & misc.
		J. Learned
II. Imaginative	3. Fiction	K. General Fiction
		L. Mystery
		M. Science Fiction
		N. Adv. & Western
		P. Romance
		R. Humor

Figure 7.1: Categories in the Brown Corpus.

Variable	Range
Adverb count	19 – 157
Character count	7601 – 12143
Long word count (> 6 chars)	168 – 838
Preposition count	151 – 433
Second person pronoun count	0 – 89
"Therefore" count	0 – 11
Words per sentence average	8.2 – 53.2
Chars / sentence average	34.6 – 266.3
First person pronoun count	0 – 156
"Me" count	0 – 30
Present participle count	6 – 101
Sentence count	40 – 236
Type / token ratio	14.3 – 53.0
"I" count	0 – 120
Character per word average	3.8 – 5.8
"It" count	1 – 53
Noun count	243 – 751
Present verb count	0 – 79
"That" count	1 – 72
"Which" count	0 – 40

Figure 7.2: Parameters for Discriminant Analysis.

Category	Items	Errors
I. Informative	374	16 (4 %)
II. Imaginative	126	6 (5 %)
Total	500	22 (4 %)

Figure 7.3: Categorization in Two Categories.

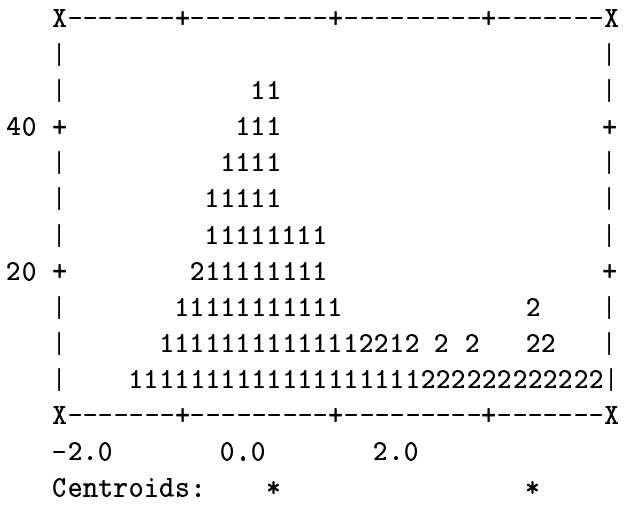

Figure 7.4: Distribution, 2 Categories.

4 categories

Using the three functions extracted, 366 cases were correctly classified, and 134 cases were misclassified, out of the 500 cases, as can be seen in Table 7.5 and Figure 7.6. "Miscellaneous", the most problematic category, is a loose grouping of different informative texts. The single most problematic subsubset of texts is a subset of eighteen non-fiction texts labeled "learned/humanities". Sixteen of them were misclassified, thirteen as "miscellaneous".

15 (or 10) categories

Using the fourteen functions extracted, 258 cases were correctly classified and 242 cases misclassified out of the 500 cases, as shown in Table 7.7. Trying to distinguish be-

Category	Items	Errors
1. Press	88	15 (17 %)
2. Non-fiction	110	28 (25 %)
3. Fiction	126	6 (5 %)
4. Misc.	176	68 (47 %)
Total	500	134 (27 %)

Figure 7.5: Categorization in Four Categories.

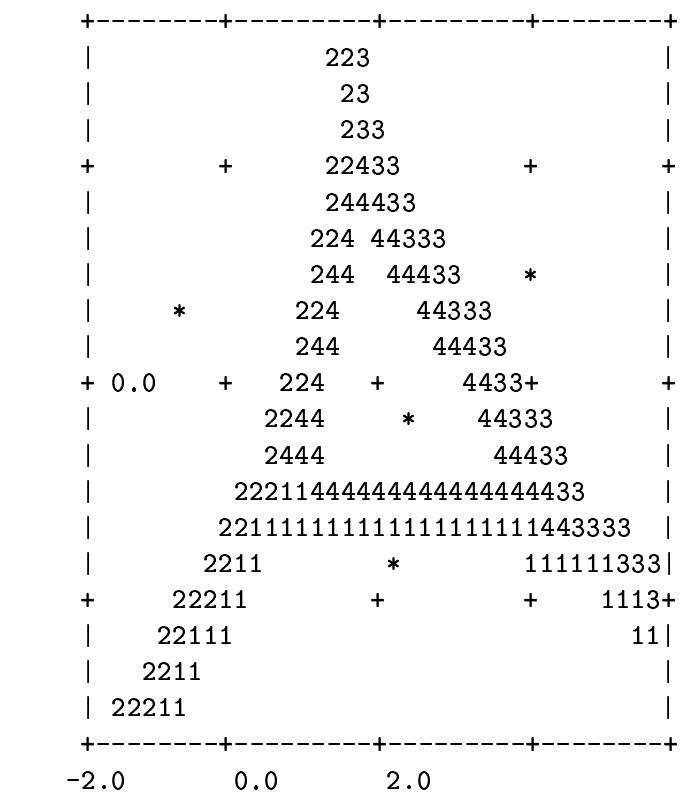

Figure 7.6: Distribution, 4 Categories.

tween the different types of fiction is expensive in terms of errors. If the fiction subcategories were collapsed there only would be ten categories, and the error rate for the categorization would improve as shown in the "revised total" record of the table. The "learned/humanities" subcategory is, as before, problematic: only two of the eighteen items were correctly classified. The others were most often misclassified as "Religion" or "Belles Lettres".

7.4 Validation of the technique

It is important to note that this experiment does not claim to show *how* genres in fact differ. What we show is that this sort of technique *can be* used to determine which parameters to use, given a set of them. We did not use a test set disjoint from the training set, and we do not claim that the functions we had the method extract from the data are useful in themselves. We discuss how well this method categorizes a set text, given a set of categories, and given a set of parameters.

The error rates climb steeply with the number of categories tested for in the corpus we used. This may have to do with how the categories are chosen and defined. For instance, distinguishing between different types of fiction by formal or stylistic criteria of this kind may just be something we should not attempt: the fiction types are naturally defined in terms of their content, after all.

The statistical technique of *factor analysis* can be used to discover categories, like Biber has done. The problem with using automatically derived categories is that even if they are in a sense real, meaning that they are supported by data, they may be difficult to explain for the unenthusiastic layman if the aim is to use the technique in retrieval tools.

Other criteria that should be studied are second and higher order statistics on the respective parameters. Certain parameters probably *vary more* in certain text types than others, and they may have a *skewed distribution* as well. This is not difficult to determine, although the standard methods do not support automatic determination of standard deviation or skewness as discrimination criteria. Together with the investigation of several hitherto untried parameters, this is a next step.

7.5 Readability indexing

Not unrelated to the study of genre is the study of *readability* which aims to categorize texts according to their suitability for assumed sets of assumed readers. There is a wealth of formulæ to compute readability. Most commonly they combine easily computed text measures, typically average or sampled average sentence length combined with similarly computed word length, or incidence of words not on a specified "easy word list" (Chall 1948), (Klare, 1963). In spite of Chall's warnings about injudicious application to writing tasks, readability measurement has naively come to be used as a prescriptive metric of good writing as a tool for writers, and has thus come into some disrepute among text researchers. Our small study confirms the basic findings of the early readability studies: the most important factors of the ones we tested are word length, sentence length, and different derivatives of these two parameters. As long as readability indexing schemes are used in descriptive applications they work well to discriminate between text types.

7.6 Application

The technique shows practical promise. The territorial maps shown in Figures 7.4, 7.6, and 7.8 are intuitively useful tools for displaying what type a particular text is, compared with other existing texts. The technique demonstrated above has an obvious application in information retrieval, for picking out interesting texts, if content-based methods select a too large set for easy manipulation and browsing (Cutting *et al.*, 1992).

In any specific application area it will be unlikely that the text database to be accessed will be completely free form. The texts under consideration will probably be specific in some way. General text types may be useful, but quite probably there will be a domain- or field-specific text typology. In an envisioned application, a user will employ a cascade of filters starting with filtering by topic, and continuing with filters by genre or text type, and ending by filters for text quality, or other tentative finer-grained qualifications.

This type of technique has been used by the IntFilter Project at Stockholm University in some preliminary studies of how texts on the USENET News conferencing systems were understood by readers. Categories such as "query", "comment", "announcement",

Category	Items	Errors	Miss
A. Press: reportage	44	11 (25 %)	F
B. Press: editorial	27	8 (30 %)	A
C. Press: reviews	17	4 (24 %)	B
D. Religion	17	8 (47 %)	G
E. Skills and Hobbies	36	17 (47 %)	J
F. Popular Lore	48	32 (67 %)	G,E
G. Belles Lettres, Biographies etc.	75	49 (65 %)	D,B,A
H. Government documents & misc.	30	9 (30 %)	J
J. Learned	80	32 (40 %)	H,D,G,F
K. General Fiction	29	16 (55 %)	fiction
L. Mystery	24	12 (50 %)	-"-
M. Science Fiction	6	1 (17 %)	-"-
N. Adventure and Western	29	18 (62 %)	-"-
P. Romance	29	22 (76 %)	-"-
R. Humor	9	3 (33 %)	-"-
Total	500	242 (48 %)	
Fiction (From previous table)	126	6 (5 %)	
Revised total	500	178 (35 %)	

Figure 7.7: Categorization in 15 Categories.

"FAQ", and so forth were used, and texts categorized using parameters such as different types of length measures, form word content, quote level, percentage quoted text and other USENET News specific parameters. (Ben Cheikh and Zackrisson, 1994; Hussain and Tzikas, 1995)

7.7 Precision or presentation

This first initial study left me with two obvious further avenues of study. One is to study the effect of using stylistic information to display document variation in an information retrieval interface, in order to aid users in selecting documents for further perusal; the other is to use stylistic information to improve retrieval precision by identifying uninteresting documents early on in the retrieval process. I attempted both: the next two parts of this dissertation text will describe the two sets of experiments.

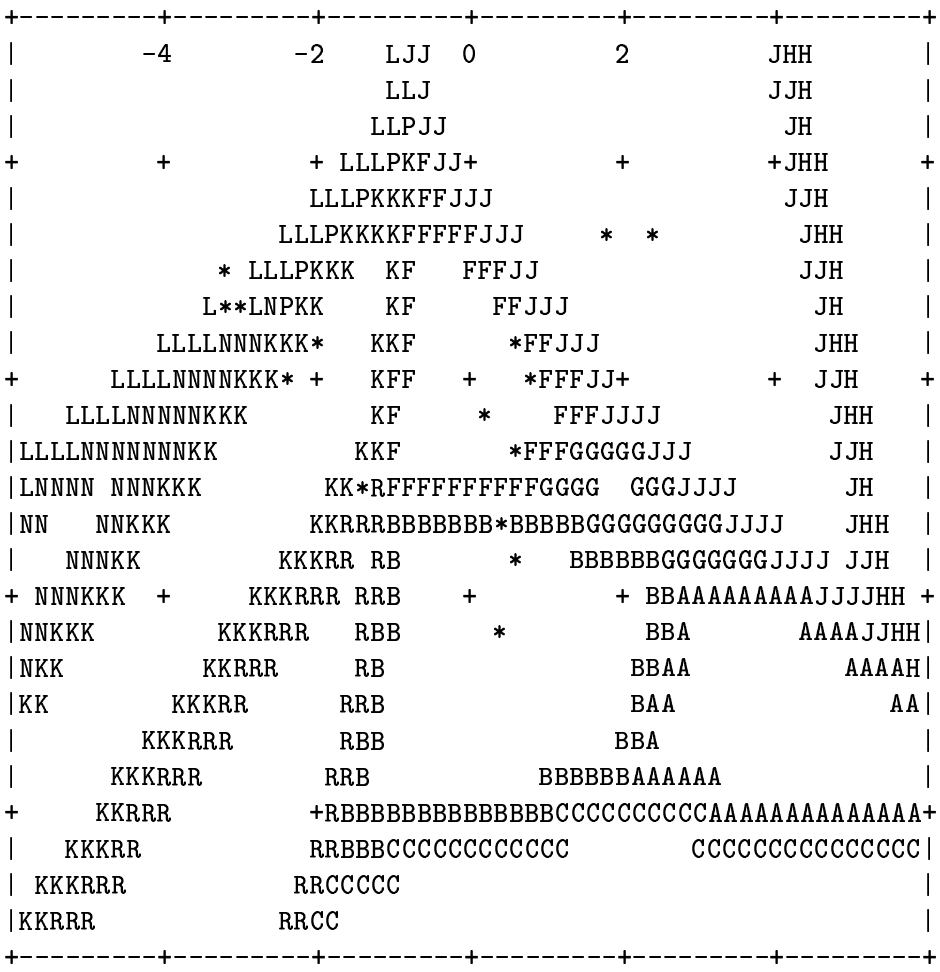

Figure 7.8: Distribution, 15 Categories — * Indicates a group centroid.

Part III

Stylistics and Relevance

These following chapters describe a sequence of experiments where I attempt to make use of stylistic information to improve retrieval precision by identifying uninteresting documents — i.e. documents unlikely to be relevant for a certain type of search context — early on in the retrieval process.

Chapter 8

Stylistic analysis and relevance

The material in this chapter describes work performed during 1996, while I worked in the Proteus project at New York University. Some of the material has previously been published as part of a chapter titled "Stylistic experiments in information retrieval" written by me in the volume "Natural Language Information Retrieval" edited by Tomek Strzalkowski (Karlgren, 1999), some presented as a poster at SIGIR'96 in Zürich (Karlgren, 1996a), and some as part of a paper titled "Natural Language Information Retrieval: TREC-5 report" by Tomek Strzalkowski, Louise Guthrie, myself, Jim Leistensnider, Fang Lin, Jose Perez-Carballo, Troy Straszheim, Jin Wang, and Jon Wilding presented to the fifth Text Retrieval Conference (Strzalkowski *et al.*, 1996).

8.1 Materials and processing

In these experiments I measure several different types of simple stylistic items and combine them using non-parametric multivariate techniques such as decision tree learning techniques. Measurements on these items have all shown differences between categories of text in various stylistic studies (Shaykevich, 1968; Sandell, 1977; Biber, 1988; Biber, 1989; Karlgren and Cutting, 1994, e.g.) and are in these experiments taken as starting points for broadening the scope of information retrieval functions after the fashion outlined in Chapters 3 and 6. There is a considerable range of variation in the material. The experiments investigate the differences between documents that have been judged relevant and non-relevant by human judges, and attempt to utilize the found differences in distinguishing the two categories of documents.

8.1.1 Corpus

The Wall Street Journal portion of the test corpus provided for the TIPSTER information retrieval research program is used as a training set for these experiments. It constitutes 74 516 Wall Street Journal articles from the years 1990, 1991, and 1992. To complement the corpus with some less well edited and more heteregenous material, I also used Excite, a World Wide Web search service, to retrieve material from the Internet using queries from the Text REtrieval Conference (TREC) as search strings. This added 3 493 HTML

documents. This material is in English, and the following experiments are entirely based on previous studies of the English language. There is no reason to assume that results between languages should be substantially different on this fairly simple level of processing, apart from the effort of collecting sets of lexical genre markers — and indeed, Bibers later studies investigate genres across several languages. The major constraint to generalizing the result from these experiments to other collections is not the language bias from using English-language materials, but the genre bias from using mainly well-edited news print.

8.1.2 Variables examined

The choice of stylistic items examined is motivated by intuitions of the implementer, moderated by processing considerations. The items include lexical statistics such as average word length, long word counts, type/token ratios, pronoun counts and digit counts as well as syntactic statistics such as average sentence length and some parsing statistics. In the tables in the following sections, I display averages and value ranges.

Word-based statistics

Word based measures such as average word length, statistics on word length distribution, long word counts, and type/token ratios have been the staple of most readability indexes and authorship determination methods since the dawn of the field of statistical stylistics (Mendenhall, 1887); statistics for several such measures can be found in Figure 8.1 and are used in these experiments under the assumption that variation in lexical choice will reflect interesting variation in overall style.

Corpus	WSJ	WWW
n	74516	3493
Average word length	5.04	5.31
	(2.5 - 7.2)	(3.8 - 21)
Long words	0.195	0.217
	(0.02 - 0.6)	(0 - 1)
Type token ratio	0.599	0.467
	(0.2 - 1)	(0.1 - 1)
Capital type token ratio	0.724	0.617
	(0.2 - 1)	(0.2 - 1)
Digit content	0.00912	0.00575
	(0 - 0.2)	(0 - 0.1)

Figure 8.1: Simple lexical statistics.

The *average word length* has a range from just under four characters to over twenty. The long averages are most likely documents that are lists of technical terms, addresses or WWW URL:s — no text is likely to have over twenty character long words on average. *Long words* are word tokens over seven characters in length; this measure is used in readability metrics mainly because of its simplicity for hand-measurement and is included here because of its great simplicity for computer processed measurement. *Type-token ratio* is an estimate measure of number of different words — the number of different words

("types") divided by the number of word tokens. This measure is not length normalized, so document length will influence this measure drastically. *Capital type token ratio* is the same measure, but calculated only for capitalized words, on the assumption that the number of different proper names in a Wall Street Journal article will vary. *Digit content* is simply the number of digits per character in the text.

Text-based statistics

Texts exhibit considerable variation in syntactic complexity (Losee, 1996; Menshikov, 1974). Word statistics — word length, long word counts, type/token ratios — as measures of terminological complexity have often been paired with sentence length to produce readability scores (Klare, 1963; Lorge, 1959, e.g.). Indeed, most statistical stylistic measures heretofore have included attempts to model syntactic complexity.

Sentence length is one such method, although arguably a blunt one — what syntactic constructions are complex in themselves, and when they are evidence of complexity in an already complex subject matter is a matter of contention (Dawkins, 1975, e.g.). Sentence length is tabulated, together with absolute text length, in Figure 8.2.

As a somewhat deeper approximation of clause complexity, I used the output of a robust parser built for information retrieval purposes (Strzalkowski, 1994a). The parser produces trees, as parsers for strict word order languages are prone to do; it is also set to surrender attempts to parse clauses after reaching a timeout threshold, so as not to choke on syntactic complexities. When the parser skips, it notes it has done so in the parse tree. The average maximum depth of parse trees for a text were counted along with the number of skip marks — both were taken as an indication of clausal complexity. The data for tree depth and skips are both tabulated in Figure 8.2.

TextTiles is a system which cuts up a text into *tiles*, pieces that can be understood as subtopic segments (Hearst and Plaunt, 1993; Hearst, 1997); the Wall Street Journal corpus was processed using TextTiles, and the number of tiles for each document was retained. The text tiling data are also shown in Figure 8.2.

Corpus	WSJ
n	74516
Length	448
	(10 - 10 000)
Average sentence length	19.3
	(3 - 100)
Average parse tree depth	9.6
	(4 - 30)
Number of parser skips per sentence	0.42
	(0 - 3)
Average number of TextTiles	2.4
	(1 - 60)

Figure 8.2: Text level statistics.

Statistics on specific items

In addition to finding global measures of the text, specific items in the text may give strong indications of the stylistic neighborhood the text belongs to. Measures such as various types of pronoun counts, for instance, can be used to predict the formality or informality of the text (Biber, 1988; Biber, 1989; Karlgren and Cutting, 1994),[1], and the presence or absence of certain lexical or ortographical items such as contractions [2] "amplifiers" — a short list of adverbs which mark the certainty of textual propositions[3] or the relative frequency of the verbs "seem", "appear" — which are used in certain types of academic prose as hedges — may be useful in a similar fashion. What cues to pick is naturally very tightly bound to the context the text is found in. For current texts from the World Wide Web it is natural to look for hypertext links and image tags; however, any attempt even to try taking a snapshot of the current state of text production on the Web will never do justice to the variation of text and text types that can and will be found there: trying to codify features of anything as fluid as the HTML standard is setting oneself up for a limitless task. In these first experiments only purely textual features are taken into account.

The intuitions behind these variables are quite shallow from the part of the implementer, and can sometimes give rise to surprises. The presence of personal pronouns, for instance, might easily be understood to signal informal or other highly subjective text. In the Wall Street Journal it mainly signals that the article at hand is an interview, rather than a third party report — not that it is a highly opinionated editorial; while interviews can be more subjective than other Wall Street Journal articles they do not need to be: a high level of first person personal pronouns may turn out to mean that the piece is a rather dry three paragraph report on some industrial sector with a three sentence quote from a Federal Reserve official. In Figure 8.3 I give values for several items, and since a nil score is quite common — meaning no such items were found in the text — in some cases I give the percentage of items that take that value.

8.2 Correlation between variables

The various variables have been tested for correlation using the Spearman rank order correlation coefficient. A sample of the entire data set was used to calculate the correlation coefficients, and the results are shown in Figure 8.4. There are no surprising correlations; the only ones higher than 0.5 are the inverse correlation between type-token ratio and

[1] Douglas Biber has used these and other similar statistics to cluster texts in a multidimensional space, in order to find underlying dimensions of variation (Biber, 1988; Biber, 1989). He defined and used his variables to find general properties of genres and varieties across a wide range of language use situations — he investigated not only English non-fiction prose, but several varieties of spoken language as well, and has lately expanded his experiments to cover other languages, to find commonalities and rule-bound differences between the categories. Among other items, I use several variables Biber proposed, to find differences on a dimension he called "informational vs involved production" — roughly, the textual variation between situationally and subjectively marked text versus more "objective" prose.

[2] "isn't", "doesn't" ...

[3] "absolutely", "altogether", "completely", "enormously", "entirely", "extremely", "fully", "greatly", "highly", "intensely", "perfectly", "strongly", "thoroughly", "totally", "utterly", "very".

Corpus		WSJ		WWW
n		74 516		3 493
Item	% with nil score	average and range	% with nil score	average and range
First person pronouns		0.00451		0.00747
	50%	(0 - 0.14)	30%	(0 - 0.12)
Second person pronouns		0.000658		0.00649
	85%	(0 - 0.09)	45%	(0 - 0.11)
Third person pronouns		0.00929		0.00796
	38%	(0 - 0.12)	20%	(0 - 0.075)
'It'		0.00686		0.00435
	30%	(0 - 0.09)	20%	(0 - 0.046)
Indefinite pronouns		0.000505		0.000796
	80%	(0 - 0.05)	60%	(0 - 0.13)
Amplifiers		0.000602		0.00113
	77%	(0 - 0.05)	50%	(0 - 0.18)
'Seem' and 'Appear'		0.000115		0.000144
	93%	(0 - 0.03)	85%	(0 - 0.008)
Contractions		0.00467		0.00279
	43%	(0 - 0.1)	50%	(0 - 0.07)

Figure 8.3: Specific item statistics.

document length and the correlation between characters per word and the number of long words. I retain all variables for the following experiments.

8.3 The TREC evaluation and relevance judgments

The Text Retrieval Conferences, or TRECs are organized annually by the National Institute of Standards and Technology in the United States. They are set up in the form of a competition, where the participating information retrieval systems each given year are given fifty standardized search queries to retrieve texts from a given large set of documents. The systems are then asked to return ordered lists of retrieved documents for the TREC evaluation, and the top one hundred documents returned by each system for each query are evaluated by human judges — given *relevance judgments* — to assess system performance.

The average 11-point precision (see Section 2.1) varies from query to query — some queries are more difficult than others, either because of the type of topic or because of the relative scarcity of relevant material. For each participating system in TREC an average of the results over each of the fifty queries is calculated, as a summary of the results.

The relevance judgments leave a large portion of the retrieved documents unjudged, but sampling tests seem to indicate that if no participating system ranked a document in the top one hundred it is unlikely it will be relevant for that query. In the relatively well-understood TREC task, information which changes the relative order of documents in a returned set can be valuable even if it only changes the judgment on a few documents, if it is consistently reliable.

	CpW	lw	t/t	C t/t	digs	wds	wps	treed	skips
CpW	1	0.9	0.2	0.0	-0.2	-0.4	0.1	0.2	0.0
lw	0.9	1	0.2	-0.1	-0.3	-0.4	0.1	0.3	-0.1
t/t	0.2	0.2	1	0.4	0.0	-0.8	0.0	-0.1	-0.3
C t/t	0.0	-0.1	0.4	1	0.1	-0.4	0.0	-0.2	-0.3
digs	-0.2	-0.3	0.0	0.1	1	-0.3	-0.1	-0.5	-0.4
wds	-0.4	-0.4	-0.8	-0.4	-0.3	1	-0.3	-0.3	-0.3
wps	0.1	0.1	0.0	0.0	-0.1	-0.3	1	0.2	0.0
treed	0.2	0.3	-0.1	-0.2	-0.5	-0.3	0.2	1	0.2
skips	0.0	-0.1	-0.3	-0.3	-0.4	-0.3	0.0	0.2	1
tile	-0.1	-0.1	-0.3	-0.1	-0.1	0.4	0.0	-0.1	-0.2
p1	-0.2	-0.1	-0.2	-0.2	-0.4	0.0	-0.2	-0.1	-0.1
p2	-0.1	0.0	0.0	0.0	0.0	0.2	0.0	-0.1	0.0
p3	-0.2	-0.1	-0.2	-0.3	-0.5	0.1	-0.3	-0.1	-0.1
it	-0.2	-0.1	-0.1	-0.2	-0.3	-0.2	0.0	-0.1	-0.2
ind	-0.1	0.0	0.0	-0.1	-0.1	0.1	0.0	-0.1	-0.1
amp	0.0	0.0	-0.1	-0.1	-0.2	-0.1	0.0	0.0	-0.1
seem	0.2	0.2	0.3	0.2	0.3	0.2	0.3	0.2	0.1
n't	-0.3	-0.2	-0.2	-0.2	-0.3	-0.1	-0.3	-0.3	-0.2

	tile	p1	p2	p3	it	ind	amp	seem	n't
CpW	-0.1	-0.2	-0.1	-0.2	-0.2	-0.1	0.0	0.2	-0.3
lw	-0.1	-0.1	0.0	-0.1	-0.1	0.0	0.0	0.2	-0.2
t/t	-0.3	-0.2	0.0	-0.2	-0.1	0.0	-0.1	0.3	-0.2
C t/t	-0.1	-0.2	0.0	-0.3	-0.2	-0.1	-0.1	0.2	-0.2
digs	-0.1	-0.4	0.0	-0.5	-0.3	-0.1	-0.2	0.3	-0.3
wds	0.4	0.0	0.2	0.1	-0.2	0.1	-0.1	0.2	-0.1
wps	0.0	-0.2	0.0	-0.3	0.0	0.0	0.0	0.3	-0.3
treed	-0.1	-0.1	-0.1	-0.1	-0.1	-0.1	0.0	0.2	-0.3
skips	-0.2	-0.1	0.0	-0.1	-0.2	-0.1	-0.1	0.1	-0.2
tile	1	0.1	0.3	0.2	0.1	0.3	0.1	0.4	0.1
p1	0.1	1	0.2	0.2	-0.1	0.2	0.1	0.3	0.1
p2	0.3	0.2	1	0.3	0.2	0.4	0.3	0.5	0.3
p3	0.2	0.2	0.3	1	0.0	0.3	0.1	0.3	0.2
it	0.1	-0.1	0.2	0.0	1	0.1	0.1	0.3	0.0
ind	0.3	0.2	0.4	0.3	0.1	1	0.2	0.4	0.2
amp	0.1	0.1	0.3	0.1	0.1	0.2	1	0.4	0.0
seem	0.4	0.3	0.5	0.3	0.3	0.4	0.4	1	0.3
n't	0.1	0.1	0.3	0.2	0.0	0.2	0.0	0.3	1

Figure 8.4: Spearman's rank order correlation between variables.

For the purposes of these experiments, the texts are considered to belong to one of three categories. The systems participating in the competition suggested as most relevant for a query have been read by the judges, and assessed as either relevant or not relevant by the relevance judgments. If a text has not been retrieved and highly ranked by any system for any query, it is *not judged*: no system liked it, and no human judge has seen or read it. If it has been judged, it can either be *relevant* or *non-relevant* for a query. All texts that are relevant for some query will be considered relevant for the purposes of these experiments — for *some* information need, there was *some* interesting material in it. All texts that have been read and judged non-relevant are considered non-relevant in these experiments: some system liked it for some information need, but no human found anything interesting in it. A text that has been judged non-relevant for one query and relevant for some other (and there are several examples!) is relevant. Again, for some information need, there was something interesting in it.

While the TREC procedure has its faults and drawbacks — among other things, it neutralizes much of what makes stylistics potentially useful in an information retrieval situation, by assuming that information needs can be encoded in short exclusively topical statements — it is by far the most comprehensive effort to set up a commonly acceptable test collection of queries and documents to go with them, and provides a clean test bench to experiment with any information retrieval algorithm. In this case my aim is to test if stylistic differences between relevant and non-relevant documents are large enough to motivate automatic rule-based filtering in information retrieval.

For this experiment, I used the Wall Street Journal portion of the TIPSTER corpus: 74 516 Wall Street Journal articles from the years 1990, 1991, and 1992, and TREC queries 202-300 with relevance judgments were used for categorization. Of these documents, 2039 were marked relevant for at least one of the ninety-nine queries; 35 289 were marked non-

relevant — judged, but not relevant for any of the queries; leaving 37 188 articles not judged[4].

8.4 Relevance of stylistics to relevance

There are at least two known facts which give us reason to hope there should be some promise in using stylistic variation to weed out non-relevant documents from the TREC material. Firstly, documents come from various sources, with different stylistic preferences, and varying usefulness for any given topic; secondly, while the TREC judges attempt to neutralize personal preferences, there are limits to how far it is possible to homogenize the retrieval event: the human judges, while well-trained for the task, are likely to exhibit biases for certain types of documents — at a minimum those which are easy to judge as being relevant or not.

The TREC evaluation makes use of several subcollections for retrieval tasks. The Wall Street Journal used in these experiments is only one of them: others include material from the AP newswire and US Patent applications. The subcollections behave differently. Some have more useful documents than others; some are preferred by systems. Weighing them together with this in mind can improve retrieval results (Voorhees *et al.*, 1994). The Wall Street Journal itself is a diverse source of text: while most of the text can be assumed to have been edited by Wall Street Journal standards into a suitable style, genre variation should still be detectable between e.g. human interest stories and stock offering reports.

As an example of how non-topical variation among texts can interact with relevance, document length can be used as a factor for picking out documents. Length normalization of term frequencies in documents has been standard in most term matching procedures: if a document index shows a certain number of hits for a term, the count is normalized to account for differences in document lengths. A number of hits for a term in a document is more significant in a short document. However, some results from the recent TREC evaluations show that length normalization does not always improve results — longer documents seem disproportionately preferred by the judges (Singhal *et al.*, 1995; Buckley *et al.*, 1996).

8.5 Results and discussion

Texts that were found relevant did differ systematically from texts which were not found relevant; for all variables tested, the difference was statistically significant even by univariate tests. In addition, relevant texts and non-relevant texts taken together — i.e. texts *retrieved* by systems participating in the TREC evaluation — differ stylistically from the rest of the corpus in a systematic manner. The difference between relevant and

[4]Relevance judgments for queries 202-250 (TREC 4) gave 1116 relevant documents, and 12 482 non-relevant ones; queries 251-300 (TREC 5) gave 1008 relevant and 31 518 non-relevant. 18 769 articles were retrieved for more than one query; fifteen for more than 25 of 99 queries. 165 were found relevant for more than one query; two for four queries.

	Relevant	Non-relevant	Not Judged	All
n	2039	35289	37188	74516
Average word length	5.07	5.05	5.03	5.04
Long words	0.199	0.196	0.193	0.195
Type token ratio	0.532	0.583	0.619	0.599
Capital type token ratio	0.671	0.713	0.738	0.724
Digits	0.0058	0.00857	0.00982	0.00912

Figure 8.5: Simple lexical statistics by relevance.

	Relevant	Non-relevant	Not Judged	All
n	2039	35289	37188	74516
Length	757	538	347	448
Average sentence length	19.7	19.4	19.2	19.3
Average parse tree depth	10	9.7	9.5	9.6
Number of parser skips per sentence	0.50	0.43	0.40	0.42
Average number of TextTiles	4.0	2.9	1.8	2.4

Figure 8.6: Text level statistics by relevance.

non-relevant texts is much smaller than the difference between either of them and the non-judged portion of the corpus.

In summary, the results of this experiment show that retrieved highly ranked texts — both relevant and non-relevant — are longer, with a more complex sentence structure than the rest of the corpus, with longer words, fewer digits, and more personal pronouns, and that relevant texts differ from non-relevant in that they tend to be even more marked on most measures.

Relevant texts have longer words and fewer digits — see Figure 8.5 — and more personal pronouns and other indicators of more involved text — see Figure 8.7. On average, they also are longer than other texts — which also has been observed, pointed out, and utilized by the very successful Cornell research group at the latest TREC conference (Buckley *et al.*, 1996). Figure 8.6 contains a summary of the text-based statistics. Relevant texts, besides being longer, also have longer sentences, deeper parse trees, and more skips.

Longer texts are more likely to be relevant at least partly due to the fact that longer texts range over several topics, and thus there is a chance that a long text will touch a relevant topic. In this material, not only are relevant documents longer, but all documents retrieved by systems, even those assessed by human judges as irrelevant, also are longer than the average document. Not only will longer texts touch relevant topics — but apparently they may well touch irrelevant but confusingly similar topics. The non-retrieved portion of the corpus turns out to contain large numbers of very short items, and large numbers of tables and numerical information, both short and long, which the retrieval systems have not proffered to the assessors for consideration. These texts presumably simply have less topical information, and thus are hit less often by the retrieval systems used. Examining the TextTiles scores for the articles yields the expected result: relevant

	Relevant	Non-relevant	Not Judged	All
n	2039	35289	37188	74516
First person pronouns	0.0077	0.0052	0.0037	0.00451
Second person pronouns	0.00078	0.00083	0.00049	0.000658
Third person pronouns	0.011	0.010	0.0081	0.00929
'It'	0.0064	0.0066	0.0071	0.00686
Indefinite pronouns	0.00066	0.00061	0.00040	0.000505
Amplifiers	0.00078	0.00067	0.00053	0.000602
'Seem' and 'Appear'	0.00016	0.00014	0.000092	0.000115
Contractions	0.0054	0.0049	0.0044	0.00467

Figure 8.7: Specific item statistics by relevance.

texts do indeed contain larger numbers of tentative subtopic segments. The number of tiles was higher for the relevant documents than for the non-relevant ones.

The differences between relevant and all other documents, and between relevant and non-relevant, are all significant in a Mann-Whitney test on a 95% confidence level for all statistics.

8.6 Conclusions

Texts differ in style. In this experiment, automatically retrieved texts differed from non-retrieved texts along several simple stylistic metrics. This shows that either 1) retrieval mechanisms are biased for style, or more likely, 2) style and topic go hand in hand. Neither of these results are surprising. Nonetheless, they may be a useful point to note for information retrieval system designers. What is more interesting, and a good starting point for user-oriented information retrieval studies is utilizing this type of measure in distinguishing *interesting* texts from *less interesting* ones. The following chapters continue along this path.

It is important here to warn from generalizing these results in the wrong way. The differences between relevant and non-relevant texts found should not be taken as general results: while useful in a TREC context, as shown by results from the Cornell design team (Singhal *et al.*, 1995; Buckley *et al.*, 1996), they are clearly an effect of the task, corpus, and assessors, not definitive words on text qualities.

Chapter 9

Genres and relevance

Some of the material in this chapter has previously been published as part of a paper titled "Stylistics and Relevance" presented to the 2nd International Conference on New Methods in Natural Language Processing (Karlgren, 1996b). It describes work done during 1996, while I worked in the Proteus project at New York University.

9.1 Stylistic variation and genres

The measures of stylistic variation found and reported in Chapter 8 showed difference between relevant and non-relevant documents as relevance is understood in the TREC task. All of these variables need to be combined in some way to predict the relevance of documents new to us. The aim is to make a decision for a large number of documents relatively rapidly and with little effort.

There are standard methods for extracting combinations of several variables that vary over a set of objects of study; using *principal components analysis* we could weight together sets of variables in linear combinations after their respective merit in dispersing the documents in display space. This I have done myself in an experiment reported in Chapter 14. The problems with principal components analysis are twofold. Firstly, it makes too strong assumptions about the variables' respective distributions. This is actually not too problematic, since the risks involved are small: the weighting chosen might not be the best one, but the results will be displayed to the user for further processing, and a less than optimal solution may well be sufficient. Secondly, and more importantly, principal components are even less perspicuous to the untrained user than are the bare variables, and thus less than useful for practical information retrieval purposes. The dimensions of variation are not readily translatable to plain English descriptors. This is where *genres* come in handy.

9.2 What is a genre?

Genre is a vague term. It is used in many ways in many different fields loosely related to each other, some more formal than other. I use a linguistically and functionally descriptive

notion of genres as groupings of documents that are a) stylistically consistent and b) intuitive to accomplished readers of the communication channel in question. Genres are dependent on context: for a business newspaper such as the Wall Street Journal the genre palette will be different from texts found on the World Wide Web or distributed by the Book of the Month club.

In most computational stylistics, genre has mostly been equated or based on text source: texts from some organization are categorized together with texts from similar organizations, with little or no regard for text usage. Examples are categories such as Wall Street Journal text archive, personal letters, technical documentation, cookbooks, (Kraus and Polák, 1967; Francis and Kučera, 1982; Källgren, 1990; Karlgren and Cutting, 1994); sometimes even on the history of a specific collection, and further down to the level of individual choice, as in authorship studies of various types (Mendenhall, 1887, e.g.). Stylistic variation can be based on the *functional styles* that occur in the collection at hand — as opposed to *individual styles* or sources of text (Vachek, 1975). Functional style can be used as the basis of genre analysis as well, necessarily dependent on some steps of subjective categorization, and my experiments here will be based on subjective categorization of an otherwise homogenous material.

9.3 Striking a balance between text function and stylistic description

Using source or individual choice as a basis for text categorization is unsatisfying from a *linguistically descriptive* point of view, and it is desirable to find a better foundation for analysis. Source or origin of text is not necessarily a linguistically salient categorization scheme: this information can usually be found in metatextual information without linguistic processing, and conversely, a shared source does not necessarily imply shared stylistic or functional characteristics. In effect, linguistic analysis is not *necessary* to categorize texts by source, nor is it *reliable* for this purpose.

My interest is in finding relatively stable characteristics of text: this is what stylistic analysis can be useful for. These characteristics, as far as I dare assume, are intentional: the author adheres to the conventions of a genre — or diverges from it — for a purpose, consciously or not, with varying degrees of success. The reason to chart stylistic characteristics of a text in an information retrieval context is to predict the usefulness of a text for a reader expressed as values along salient dimensions of textual variation or membership of the text in a category. I believe reader perceptions of *text functions* are central to this task: what readers believe about texts is what underlies categorization schemes and authorship alike. Similar methods have been used for Usenet News materials to determine a genre palette by a combination of text and reader study (Ben Cheikh and Zackrisson, 1994; Hussain and Tzikas, 1995). Robert Sigley uses statistical methods to define an index of formality for texts, and argues in a similar vein that text categories — similarly to styles and genres — must be defined in functional categories, without allowing the automatic means of detection to obscure the "meaning" of the category (Sigley, 1997).

In Chapter 15 I will describe an example of using a method involving user interviews, where genres are defined for the express purpose of interaction between reader and text collection. We built a genre palette through interviewing users, trying to define genres that are reasonably consistent with the vaguely expresssed user attitudes and expectations, and yet conveniently computable using observable measures of stylistic variation as outlined in the previous chapters. Finding this balance is a two-way process, trying to find a set of genres that both a) users understand and b) can be distinguished using automatic means — and letting the processes influence each other.

In the present experiment the stylistic variation investigated is taken from a fairly well-edited body of text — the Wall Street Journal — where presumably most writers conform to the expected norms of writing, and if not, the texts will be edited by professional editors to conform to them. With a common source for all texts, the categorization used below is based on inferred function of the text. The Wall Street Journal has a limited palette of self-labeled subgenres the texts adhere to. This experiment does not intend to communicate the genre palette immediately to a reader or user: the genres will be related to relevance in a TREC experiment, not used in an interactive setting.

9.4 Corpus and statistical measurements

For this experiment, a part of the TREC test corpus was selected: 74 516 Wall Street Journal articles from years the 1990, 1991, and 1992. Of these documents, 1116 were marked relevant for at least one of fifty queries (TREC queries 201-250) and 12 482 marked non-relevant, i.e. judged, but not relevant for any of the queries[1]. Initially, the documents were analyzed to obtain simple sentence statistics and to obtain simple measures for syntactic complexity.

9.5 Hypotheses

The hypotheses of the experiment were 1) that certain genres or types of text would be more likely to provide the answers the human judges would prefer, and 2) that this preference is clear enough to be detectable even using the fairly simple mechanisms tested in this experiment.

The stylistic variation was expected for two reasons. Firstly, by the likelihood that the corpus contains materials that will never be useful in a generally framed information retrieval task such as TREC: stock report tables and the like; secondly, by the fact that the human judges, while well-trained for the task, are likely to exhibit preferences for certain types of documents, namely those which are easy to judge as being relevant or not.

[1] 3 493 articles were retrieved for more than one query. Three articles were retrieved for more than 25 of 50 queries.

N	Genre	Category	Tree Depth	Skips	Words	Typ /Tok	Chars /Word	Digits /kChars	Words /Sent
11 331	A								
330	(2.9%)	rel	10.2	0.527	980	0.478	5.09	3.13	19.4
3094		non-r	10.0	0.511	988	0.476	5.05	3.01	18.9
7907		not j	9.86	0.498	782	0.503	4.99	3.44	18.2
209	B								
5	(2.3%)	rel	8.40	0.360	6717	0.323	4.74	32.7	21.1
70		non-r	8.40	0.341	3933	0.346	4.82	21.5	17.7
134		not j	8.20	0.165	1481	0.335	4.62	38.9	27.1
13 669	C								
309	(2.2%)	rel	9.89	0.502	677	0.511	5.08	7.38	20.3
2516		non-r	9.87	0.482	656	0.504	5.09	7.73	20.7
10844		not j	9.44	0.459	528	0.516	5.00	10.4	20.5
6006	D								
124	(2.0%)	rel	9.63	0.480	1009	0.495	4.95	5.25	18.2
1278		non-r	9.53	0.477	1075	0.484	4.94	4.89	17.9
4604		not j	9.38	0.464	835	0.514	4.90	6.00	17.8
2613	E								
49	(1.8%)	rel	10.3	0.516	1249	0.442	4.89	2.97	19.5
604		non-r	9.91	0.503	1228	0.446	4.90	3.06	18.9
1960		not j	9.95	0.499	855	0.486	4.86	3.24	18.7
3187	F								
48	(1.5%)	rel	10.6	0.580	597	0.554	5.17	4.32	21.2
707		non-r	10.1	0.484	503	0.577	5.20	3.49	19.8
2432		not j	9.78	0.434	367	0.600	5.12	4.53	19.6
21 941	G								
183	(0.8%)	rel	10.3	0.452	241	0.629	5.18	6.25	20.8
2526		non-r	9.90	0.409	189	0.644	5.17	7.29	20.3
19232		not j	9.55	0.388	169	0.651	5.10	8.73	19.6
3539	H								
21	(0.5%)	rel	9.24	0.397	535	0.588	4.91	21.1	13.7
490		non-r	8.83	0.402	643	0.543	4.85	27.0	11.7
3028		not j	8.17	0.331	467	0.566	4.83	27.1	14.5
1096	I								
6	(0.5%)	rel	9.12	0.460	377	0.603	4.35	51.1	18.7
145		non-r	8.33	0.275	677	0.610	4.29	77.2	17.0
945		not j	7.12	0.150	250	0.691	4.67	65.2	24.9
10 925	J								
41	(0.3%)	rel	10.1	0.476	107	0.743	5.23	6.31	22.4
1052		non-r	10.4	0.330	75	0.800	5.24	7.25	20.2
9832		not j	10.1	0.328	70	0.805	5.15	8.14	19.5

Figure 9.1: Clusters based on stylistic data, and their proportions of relevant documents.

9.6 A quick and dirty genre categorization

Stylistic variation, besides individual choice, is partly an effect of genre variation. To get closer to the genres one can expect to find in the corpus text one issue of Wall Street Journal (910102) was categorized manually into ten rough categories: articles, business news with and without tables, lists of paragraph length items, editorials, letters, paragraph-length items, "What's News" (menu-type lists of one-sentence items), tables, and single one-sentence items. This was used as a training set to categorize the entire corpus: simple stylistic measurements for the hand categorized data — as shown in the previous chapter[2] — were used in a discriminant analysis, and the resulting discriminant functions were used to automatically categorize the entire corpus. The details of the method are not important: the result is sloppy in any case. No checking was made to see how well and consistently the articles were categorized in the genres given; the idea was simply to have a seed set to cluster the documents around. In Figure 9.1 some statistics for each category are shown; the category names have been replaced with letters so as not to imply the categories are consistent with real life genres.[3]

The hypothesis was that a simple stylistic clustering might well prove useful thanks to its anchoring in genre, and in spite of this anchoring being quite tentative. Figure 9.1 shows that there are considerable differences between the categories in stylistic metrics — unsurprisingly, since they have been clustered to maximise that difference — but more importantly, the categories show considerable differences in how large a proportion of the documents is relevant, and most importantly, in *how* the relevant documents differ from the non-relevant ones stylistically. For instance, whereas in category A, relevant documents will have longer sentences on average than non-relevant and non-retrieved documents, in categories C and H the relevant documents will have shorter sentences; and whereas most categories prefer documents with a low type-token ratio, category H prefers documents with a high ratio.[4]

9.7 Conclusions

These results show that the stylistic difference between relevant and non-relevant found in the previous chapter indeed can be found to relate to somewhat functionally defined text categories. My claim is that understanding relevance will entail analyzing the tasks and expectations of users; this experiment shows that for a certain set of users and for a certain scenario a clear bias towards certain types or genres of text can be found. The experiment also shows that stylistically determined genres or functional styles are different as regards potential usefulness for the queries tested, and that the distinctions between relevant and non-relevant differ between genres. These results should be taken as a starting point for further formal study of how situations affect measures of stylistic variation.

---

[2] With one extra measure added: digits per character, multiplied by 1000.

[3] The determined reader will be glad to know that the order of the document clusters in the figure is the same as in the listing of the hand-assessed seed categories in the beginning of the paragraph.

[4] These differences are significant on a better than 95% level, by Mann-Whitney U (see Section 10.2.1; most numbers in Figure 9.1 have not been checked, however.

Chapter 10

On non-parametric multivariate statistics and non-linear combinations

10.1 Underlying assumptions of multivariate tools

Much of today's exploratory linguistics is data-intensive, gathering large numbers of text samples from large numbers of languages and large numbers of genres.[1] These data are rather less than illuminating unless treated and collated in some useful way. This is the purpose of statistical analysis. There is a large number of textbooks, toolkits, and general-purpose statistical data processing systems available, and many data-oriented linguists happily make use of clustering, factor analyses, discriminant analysis, and various types of significance- and hypothesis-testing tools available in the systems.

But there are risks involved in using ready-made tools. There may be underlying assumptions about the data that linguistic data do not fulfil. Most processing algorithms that are built to handle several variables at once and check for interactions between them — *multi-variate* methods — presuppose that all the variables involved have similar distributions; some even presuppose normal distributions.

But common distribution assumptions cannot be expected to hold for language data: there is no reason to believe that the presence of a certain word order or stylistic item or their relative frequencies would follow specifically a normal distribution, or any other distribution modeled on variables from engineering or biology. In fact, some linguistic items do follow normal distributions, at least close enough for multivariate methods not to fail.

There are excellent arguments that using standard packages is risky: David Lee, e.g., argues that Biber's results are based on faulty methodology and shoddy statistics (Lee, 2000). Biber's work formed the basis for the experiments Douglass Cutting and I performed as reported in Chapter 7 and Chapter 8, but for the purposes of the present set of experiments, Biber's calculations and specific finds are less interesting than the general tenor of his work: texts can be distributed over a set of genres using a large set

[1] "Large" in comparison with previous research, that is.

of interrelated but weak variables. This, coupled with the fact that most variables in use here are strong enough to withstand some rough handling in statistics packages, gives purchase to further experimentation.

10.2 Variables with unknown distribution

In any case, some of the stylistic variables I have been studying in the application of stylistic metrics to information retrieval systems patently and unsurprisingly do not follow a normal distribution. This realization can be handled in several ways:

1. Cross fingers and hope that the discrepancy does not affect the results. This is what Douglass Cutting and I did in the first experiment, and what Biber does in his work.

2. Recode the variables to follow a normal distribution, whereupon standard packages can be used without loss of peace of mind. There are standard methods for doing this, and some are displayed by Rolf Sandell (Sandell, 1977) who employs this workaround specifically in order to work with stylistic variables.

3. Try to understand the distribution of the underlying terms. This is the approach followed by Slava Katz (Katz, 1996) who has other applications in mind, but is unhappy with some of the assumptions made about how words appear in texts. As is clear from his work, building models of word distributions is non-trivial.

4. Not rely on statistical methods that make assumptions on the behavior of variables. This is the route I have chosen for the present experiments. I use standard methods developed for *non-parametric* variables, i.e. variables with no *a priori* approximation of their value space.

   There are no standard tests of multivariate significance for non-parametric variables. This means that the significance tests referred to in these experiments will not address interaction between variables, but will proceed by testing the variables one by one. This means that the results may miss some positive effects of variable interaction, but does not risk a false positive result from using a multivariate test based on false assumptions. Eventually, non-parametric multivariate tests need to be developed specifically to suit the purposes of linguistics.

10.2.1 Mann-Whitney U

Measuring some feature of members of two sets may show differences between the sets, but establishing if these differences are due to random variation or a systematic difference between the sets is not always easy. Both types of error are common: finding that a clear and convincing difference in measurement in fact could be random, or finding that a small and obviously insignificant error in fact absolutely cannot be random.[2]

[2]Rather arbitrarily, most tests are conventionally used to establish "significance" — meaning that there is a risk of one-in-twenty of error in trusting the test. I will follow that convention here.

Rank	Set membership	Some measurement
1	A	179
2	B	170
3	B	170
4	A	169
5	A	169
6	A	169
7	B	152
8	B	150
	Rank sum	Average
Set A:	16	160,5
Set B:	20	171,5

Figure 10.1: Calculation of rank sums. The difference in average measurement value is large, but the difference between the rank sum of set A and set B is small.

The Mann-Whitney U rank sum test is one of two equivalent formulations[3] for calculating significant differences between some measurement in two sets. Each member of the two sets is measured. The sets are then sorted together into one list by the result of the measurement, and then the measurement is discarded — all calculations are performed on *rank* in the list rather than the measurement itself. This means that the test relies on fewer properties of the measuring process and underlying variable.

Rank	Set membership	Some measurement
1	A	172
2	A	171
3	B	170
4	A	169
5	B	168
6	A	167
7	B	166
8	B	165
9	B	165
	Rank sum	Average
Set A:	13	169,75
Set B:	32	166,8

Figure 10.2: Calculation of rank sums. Here set A has a much lower rank sum than set B; the difference in average measurement is small.

For each set, the sum of ranks is calculated, with special rules applied to ties. If the members of the sets tend to be found towards different ends of the list the rank sums differ: one set will have a high rank sum; the other, a low. The difference is calculated with respect to the relative sizes of the sets, and if this difference is over a threshold for significance we may infer that the sets indeed are different. For this threshold a normal distribution *is* used as a yardstick. The two examples in Figure 10.1 and

[3]The other is Wilcoxon's rank sum test.

Figure 10.2 show where rank sum and average measurements give different impressions whether the sets differ or not. The Mann-Whitney U test does not make assumptions about which distribution the sampled variables follow, but if the difference in distribution shape between the variables is very large, may give a false indication of difference.

10.2.2 Spearman's ρ (rho)

To test for correlations between variables I use the Spearman rank order correlation coefficient (also known as Spearman's ρ or rho) which just like the Mann-Whitney U test is nonconfessional as regards distributions. Spearman's rho is a measure of the linear relationship between two variables. The most common correlation measure — Pearson's correlation coefficient — is similar to Spearman's rho: in fact they are calculated using the same computations. The only difference is that in Spearman's rho, the values of the variable are first converted to ranks, thus obscuring variability in distribution.

Measurement for some variable a	Rank for a	Measurement for some other variable b	Rank for b	Squared rank difference
182	2	75	4	4
169	6	66	7	1
185	1	85	1	0
152	7	61	8	1
173	5	80	3	4
175	4	82	2	4
180	3	70	6	9
150	8	72	5	9
				Sum: 32

Figure 10.3: Calculation of rank difference.

The difference between ranks is calculated, squared, summed, transformed through a calculation related to the size of the data set, and finally compared to a threshold table, which relates the rho value to reasonable values for the sum of squared differences. In the example in Figure 10.3 the squared sum of differences is 32, which after a few transformations and given the number of items and so forth may give reason to assume that the example variables indeed are correlated.

10.3 Combining several measurements — multivariate analysis

Given that we have a set of measurements that we believe give a reliable model of whatever we are working with — in the case at hand, stylistic measurements that we believe indicate differences in genre — we must combine these measurements into decisions about each item under study: deciding relevance, in this case.

10.3.1 Linear combinations

Many knowledge-based systems combine information from different sources by *weighting* the sources using some training scheme based on examining historical data. These weights are then used in some usually unspecified way to combine information. Most often the method used will be a linear combination of measurements: "add five times the value of variable x to three point fourteen times the value of variable y to obtain the score on scale z." But there is no reason to assume that variables engage in a relationship of a type that is suitable for linear combination.

Linguistic data are a case in point. Some variables may have a *polar*, or binary, distribution: "If singular first person pronouns are present, a text is not a legal text." or the relationship may be more complex: "If there are numerous tense shifts and a relatively high incidence of personal pronouns the text is an interview." which could contrast with "If there are plenty of pronouns in the text it is fiction."

The experiments by Douglass Cutting and me reported in Chapter 7, and further continued by me in Chapter 8 made use of linear algebraic combinations in form of discriminant analysis. Linear algebra is a robust way of combining evidence, and seems to work even transplanted to this type of unexpected scenario. This is theoretically unsettling in the sense that one would expect more complex dependencies between knowledge sources from vastly different levels of abstraction. It is also wasteful of linguistic knowledge in the sense that that linguists *know* or should know about interrelations between linguistic variables and not need to throw that information out in order to rediscover some of it in probablilistic formulae. Most importantly, results of this form have low or no explanatory power. Better ways of combining evidence, through production rules, decision trees, general pattern matching techniques, algebraic techniques, and combinations thereof are necessary to be able to make use of and understand linguistic data.

10.3.2 Knowledge-based classification schemes

There are numerous general concept learning schemes that have been developed for classification tasks irrespective of domain. In these experiments I have used the classification package C4.5 (Quinlan, 1993) which is freely available for experimentation.

Classification systems ingest precategorized training data and deliver sets of criteria whereby test data can be categorized into given classes. The requirement for using this type of scheme is that there are predefined discrete categories that the training data belong to. The classification system then successively partitions the training data into smaller and smaller subsets using a single attribute at a time, until a subset is deemed suitable homogenous. The art of building a useful classification system of this sort hinges on a) choice of which partition of many possible ones to make at any given point, and b) knowing when further partitioning of the training data results in more noise than real knowledge.

As a result of examining the training data, C4.5 builds classifiers based on the data in the form of *decision trees*, which is a tree-like algorithm structure with nodes either *leaves* indicating class membership, or internal *decision nodes* with an attached test to be made on some attribute value of the case at hand and one branch and subtree for

Outlook	Temp (deg C)	Humidity (%)	Windy?	Class
sunny	25	70	true	Play
sunny	28	90	true	Don't Play
sunny	30	85	false	Don't Play
sunny	21	95	false	Don't Play
sunny	18	70	false	Play
overcast	21	90	true	Play
overcast	29	78	false	Play
overcast	15	65	true	Play
overcast	28	75	false	Play
rain	21	80	true	Don't Play
rain	16	70	true	Don't Play
rain	25	80	false	Play
rain	17	80	false	Play
rain	20	96	false	Play

Figure 10.4: Example data (from Quinlan).

each possible outcome of the test. A case is classified by traversing a tree from the root outward, moving through it until a leaf is encountered. These trees can be reexpressed by C4.5 as *if-then* rules.

In Figure 10.4 example data taken from Quinlan[4] will help clarify the type of analysis made by the classification tool. The example data are partitioned by the classification tool into subsets by testing on one of the four variables available. The aim is to find a subset where all members are of one single class. In this case, a test on the variable "outlook" has three outcomes: "sunny", "overcast", and "rainy". The items in the middle group, with value "overcast", are all of class "Play", but the other two groups still have mixed classes. If the first subset is further divided by a test on "humidity" and the third subset by a test on "Windy?" each of the subsets will only contain items from a single class. This classification can be formulated as a decision tree, as shown in Figure 10.5.

For the purposes of the experiment at hand, C4.5 has advantages: above all, it is easy to use. It has the drawback that it does not allow for interactions between variables: it processes them one by one. C4.5 rules are of the form "if variable X has value more than a and variable Y has value less than b for an item then the item is of category C" — what one would wish for is a mechanism of combining sums and products of variables for more complex interaction: "if variable X has a higher score than variable Y and variable Z has value more than a and there are no other items that have a sum of X, Y, and Z higher than this item then set the threshold for membership in category D to be b". However, C4.5 does allow for interesting non-linear interactions in the form of multi-part conditions — a condition can specify several variables in conjunction.

[4] As a localization attempt, the temperatures in the example have been converted to Celsius or Centigrade degrees, but the underlying behavioral model is incomprehensible for Swedes and defies translation. *Not* play when the sun is out?

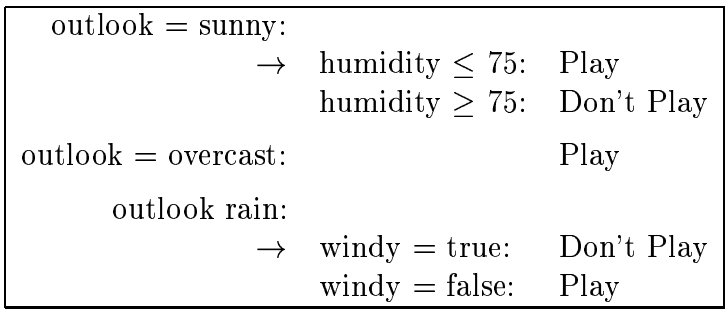

Figure 10.5: Decision tree corresponding to example data in Figure 10.4.

Chapter 11

Stylistics and precision

The material in this chapter describes work performed by Troy Straszheim and me during 1996, while I worked in the Proteus project at New York University. Some of the material has previously been published as part of a chapter titled "Stylistic experiments in information retrieval" written by me in the volume "Natural Language Information Retrieval" edited by Tomek Strzalkowski (Strzalkowski, 1999) and some as part of a paper titled "Natural Language Information Retrieval: TREC-5 report" by Tomek Strzalkowski, Louise Guthrie, myself, Jim Leistensnider, Fang Lin, Jose Perez-Carballo, Troy Straszheim, Jin Wang, and Jon Wilding presented to the fifth Text Retrieval Conference (Strzalkowski *et al.*, 1996).

11.1 Stylistics and relevance, again

The preceding chapters established that topical and stylistic variation are not independent, and thus that relevant texts differ from non-relevant. The following experiment tries to establish a method to utilize the difference predictively the likely relevance of a new unseen text. To make use of the stylistic differences between relevant and non-relevant documents, I use the C4.5 classification tool as described in Chapter 10 (Quinlan, 1993). C4.5 takes multivariate material and produces simple rules to partition items into classes using the variable values.

The two rules shown in Figure 11.1 are examples learned from relevance judgments for TREC topics 202-250. They identify likely candidates for non-relevant documents. The working hypothesis was that typical non-relevant documents share characteristics that may be easier to identify than typical relevant documents, which may be of appropriate style but of irrelevant topic. There were several similar rules learned from the data — these were chosen for display because of their relative transparency and clarity: in essence both are rules to disfavor stock market listings in tabular form in preference to text.

As described in Section 8.3, systems that participate in the TREC evaluation searched the TREC document space and retrieved 1000 documents for each query. Our project group participated in TREC with several submissions, and I took our best submission as a baseline result. This system had an average 11 point precision, as described in Section 2.1, over all queries of 0.1460. Using this list, and the stylistic measurements

∘	•	baseline
	Word count ≤ 1308	
	Chars per word ≤ 5.10619	
Digits > 0.0179775	Digits > 0.0119782	
	Words per sentence > 17.4583	
	1st person pronouns ≤ 0.0166667	
	Amplifiers ≤ 0.00234467	
$\rightarrow$ class non-relevant	$\rightarrow$ class non-relevant	
Average precision:		
0.1459	0.1452	0.1460

Figure 11.1: Example rules and their overall effect on average precision.

from the previous chapters, I tried to find methods that would identify non-relevant documents in the list. These documents would then be moved to the end of the list, hopefully improving the evaluation results for the query.

The rules in Figure 11.1 were tested — along with other similar ones — on the ranked list. The rules rearranged the list, not throwing out any documents, but moving likely irrelevant documents to the end. In spite of this cautious approach, the results are not conclusive improvements. The TREC average precision is a brittle measure: two or three relevant documents mistakenly rearranged to come towards the end of the list may completely swamp the effects of several dozens of irrelevant documents that were correctly demoted.

Figure 11.1 shows the combined average precision, and the graph in Figure 11.2 shows the results query by query: in the graph, the results are expressed as compared to the original result. Overall, the results are slightly worse than if the rules had not been in effect, but a look at the graph reveals that the rules have varying utility for the queries. In face of the overall result, this is encouraging — one might think that a rule such as the one marked with ∘ in Figure 11.1 only should be brought to apply if the query does not ask for figures; while the two extreme cases are not easily identifable as different from each other — queries 272 and 274, as shown in Figure 11.3 — one might expect that the differences in query expressions may be useful in some way.

In conclusion, this set of experiments shows that stylistic rules of thumb may indeed improve the performance of automatic retrieval systems, for some queries. They also show that stylistic rules of thumb are too specific to be useful across the board, in all situations, and that this sort of information will be best used in a case-by-case manner. The consequence is that to make use of stylistic variation for reliable relevance grading we need a query typology: each query must be identified for likely style preferences.

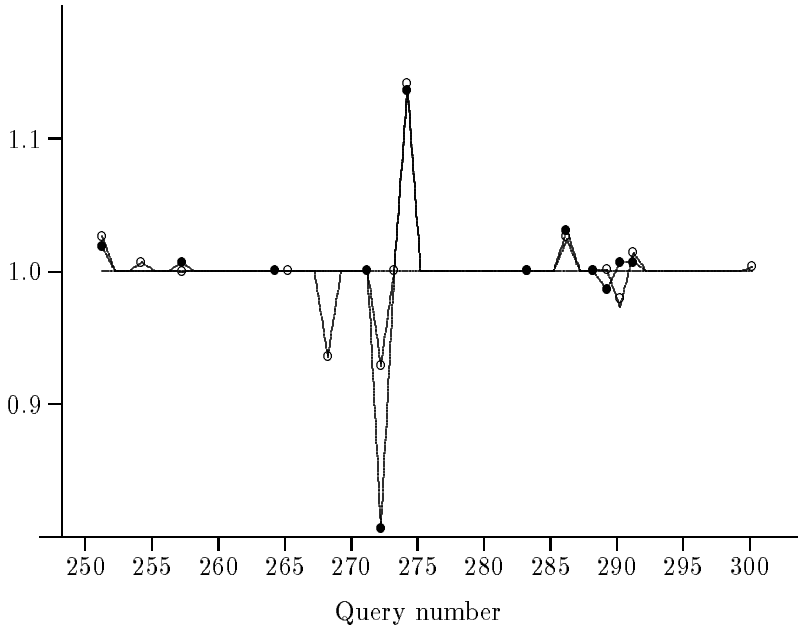

Figure 11.2: Average precision change query by query.

Disaster query: Query 272

```
Medically, is outpatient surgery
more prevalent now than ever before?
```

Best query: Query 274

```
What are the latest developments in
production of electric automobiles?
```

Figure 11.3: Query examples.

Chapter 12

Experiments in query categorization

This chapter describes work done by me during 1996, while I worked in the Proteus project at New York University. It was supported by the National Science Foundation under grant IRI-93-02615 and by the Defense Advanced Research Projects Agency under contract 94-F157900-000 from the Office of Research and Development.

12.1 Queries are different

Previous experiments have shown that it is possible to find rules that pick out non-relevant documents from sets of ranked documents. The success rate for such rules varies across queries, much as can be expected: queries are different from each other, which is reflected in all TREC submissions.

From the experiments reported in preceding chapters we now know that there are statistically significant differences between relevant and non-relevant documents in the TREC Wall Street Journal material as a whole. So far, so good, but the problem comes when trying to apply these results on a query-by-query basis. The sets retrieved for each query are different stylistically; the genres and styles vary from topic to topic and thus from query to query. Rules to distinguish relevant from non-relevant for the entire corpus cannot be indiscriminately applied across queries. Often performance is degraded, or at least not improved significantly: most queries are either unaffected or slightly improved. A few are disasters, percentage-wise. A disaster is a case of wrongly reranking four or five documents: the margins are slim when using TREC measures. Especially if a highly ranked document is demoted to the end of the list the average precision by TREC measures falls noticeably.

The case of the two example rules from the previous section (◦ rule and • rule — shown again in Figure 12.1) is an illustrative example. They both work quite well for their respective training corpora and when tested on the following year's TREC material improve results for many queries — as shown in Figure 11.2. However, both rules suffer breakdowns on at least one query; this reduces the advantage gained from the other queries so that average precision is somewhat lower than for the unprocessed set. They get disastrous results on query 272, where the • rule mistakenly demotes three relevant documents, one of them from rank number one, and the ◦ rule demotes two. In all, the

∘	•	baseline
	Word count *leq* 1308	
	Chars per word *leq* 5.10619	
Digits > 0.0179775	Digits > 0.0119782	
	Words per sentence > 17.4583	
	1st person pronouns *leq* 0.0166667	
	Amplifiers *leq* 0.00234467	
rightarrow class non-relevant	*rightarrow* class non-relevant	
Average precision:		
0.1459	0.1452	0.1460
w/o "numbers" queries		
0.1465	0.1465	0.1460

Figure 12.1: Example rules and their overall effect on average precision for a subset of the queries.

∘ rule improves scores of ten queries and lowers scores of four, with a resulting combined average precision (as described in Section 8.3) of 0.1459 (99.9% of the baseline); the • rule improves ten and lowers two queries with an average of 0.1452 (99.4%).

12.2 Typologizing queries by query appearance

The question is, if queries differ, can their differences be found and observed systematically? Can queries be categorized for expected stylistic preferences solely from their form or from the set of documents retrieved for it?

The ∘ rule is a case in point. Quite clearly, it disfavors numerical data. Looking at the queries in the test set and trying to identify which queries are numbers-oriented and which are not, it is clear that some queries indeed explicitly ask for documents which "... reveal the number of jobs lost ..." (251), include "... comparative studies which show any disparity in longevity ..." (254), note that the objective is to find "basic data for ... comparison" (257), or that "... to be relevant a document should show cost figures ..." (268), to pick a few examples somewhat less than randomly.

So, if I only apply the rules to a hand-filtered set of queries that make no mention of "figures", "numbers", "increased rate", or "basic data" and other like terms, there indeed is a slight improvement: the average precision improves, to become slightly better than baseline as shown in Figure 12.1. But not all queries work that way. Of the four examples mentioned above, both 251 and 254 improve average precision *using* the ∘ rule, in spite of being "numbers" queries. This is the sort of information that really should be elicited from the user in some explicit way rather than being second-guessed from the rather bare form of the TREC queries.

Reranking rule	Number of queries	Number of improved queries	Number of degraded queries	Average precision change
Number	24	14	6	0.955
Complex	24	11	10	0.988

Figure 12.2: Average result change per query type.

12.2.1 Relevant and non-relevant documents in subsets of the corpus

In the first experiment in this chapter I used rules which were found by searching for differences between relevant and non-relevant documents over the entire set of retrieved documents from TREC-5, and then tried to see if they could be better applied to a subset of the queries that seemed to fit them. This seemed to work, although the gains were modest.

My next experiment tests for differences between subsets of the corpus. As shown in Chapter 9, the differences between relevant and non-relevant documents are larger in stylistically homogenous subsets of the corpus — which would seem to be useful in this experiment. I identified two categories of query: the "Number" category as given above, of 24 queries, and another similarly vague category of 24 "Complex" queries — queries where the language of the query seemed to suggest that the retrieved documents could be complex or abstract, such as scientific text.

I then applied C4.5 to learn rules specifically for the documents retrieved in the TREC-5 data for each of the two categories: my hypothesis was that if the queries seem to favor numbers or complex text, this should shine through in the retrieved material, and in the relevance judgments made on them. I found rules to distinguish non-relevant documents for each of these training sets. I then reranked the output of previously submitted TREC-6 data for these queries using the rules, so that documents identified as non-relevant by the rules are demoted to the end of the list. These results are shown in Figure 12.2. As before results are inconclusive but encouraging: while the average precision change, by the finicky TREC metric, shows some loss, most queries show improvement.

12.3 Cluster queries by retrieved set

As a second experiment, I clustered queries not by their appearance but by examining properties of the documents retrieved for it. The hypothesis was that even if the difference in appearance between TREC queries is too small to cluster queries systematically, the types of document they retrieve may show enough differences to be useful for clustering the queries. A certain query might retrieve more numerical data than another query; that tendency may be reason enough to characterize this query as a "Number" query — which information can be used to bias the ranking of the retrieval results.

For each query, I took the top hundred documents and then extracted those documents which were from the Wall Street Journal. Using stylistic characteristics of these articles, as measured by the variables described in the previous chapter, I ran a clustering algorithm[1] and found reasonably dissimilar retrieval sets — one large: b, and some smaller ones: a, c, d, and e. Using these retrieval set clusters I now could categorize each test query into one of five categories.

12.3.1 Find non-relevant documents for each category

Using C4.5 to examine the training set yields classification rules to identify non-relevant documents for clusters b (3 rules), d, and e (2 rules). No relevance rules were found for categories a and c, and the queries in these categories were removed from further processing: the remaining 43 queries were in clusters b (24 queries), d (10 queries), and e (9 queries).

12.3.2 Rerank previous runs

Finally, as in the experiment above, I reranked the output of previously submitted TREC-6 runs (Strzalkowski *et al.*, 1996) using the rules, so that documents identified as non-relevant by the rules are demoted to the end of the list. Examples of quite disappointing results are shown in Figure 12.3. The average precision is at times improved using cluster-specific rules, but does not improve on the overall scores. The net result is worse than without intervention.

Rule	n	All queries	n	Cluster only
b1	50	0.998	24	1.002
b2	50	0.942	24	0.966
b3	50	0.999	24	1.001
d	50	0.937	10	0.900
e1	50	0.927	9	0.898
e2	50	0.981	9	0.964

Figure 12.3: Average precision change per rule.

12.4 Conclusions

Clustering queries post-hoc with no information from the user is not possible in a systematic use. Stylistic analysis of retrieved documents seems to be of potential use, but the bottleneck clearly is in how useful information such as stylistic preferences can be elicited from the user without adding undue work load in the information retrieval situation.

TREC test queries are expressly designed to hide other than precisely topical information from the system; to have this sort of mechanism work, the system must elicit

[1] PROC FASTCLUS of the SAS package.

more and other types of information from the user — either by expressly querying the user or by inferring information from the query or topic matter itself. Both approaches need investigation, both by further textual experimentation and by user studies.

Part IV

Stylistics in Interactive Information Retrieval Systems

These following chapters describe a set of experiments to use stylistic information to display document variation in an information retrieval interface. The aim is to utilize more knowledge about documents to improve user control over the information access process. Current systems have little knowledge of text, and interaction with them is designed after that fact. With more knowledge of documents, such as stylistic analysis can provide, the interaction can be richer. The experimental designs shown in the next few chapters utilize this.

Chapter 13

Visualizing Stylistic Variation

The material in this chapter has previously been published as part of a paper by Troy Straszheim and me titled "Visualizing Stylistic Variation" in the Proceedings of the 30th Hawaii International Conference on Systems Sciences (Karlgren and Straszheim, 1997) and in a paper titled "Stylistic and Relevance" presented to the 2nd International Conference on New Methods in Natural Language Processing (Karlgren, 1996b). It describes work done by Troy and me during 1996, while I worked in the Proteus project at New York University. The work presented here was planned and supervised by me; Troy Straszheim did the first implementation of the system design and named the tool IKSVIT.

13.1 Aim of these experiments

As discussed in Chapter 5 and Chapter 6, one of the bottlenecks of information access systems is the information flow from system to user: systems present retrieval results in a list, possibly ordered by likely relevance. If more information is available about the items under consideration, this presentation can be improved considerably: systems widely in use today give a short two-line summary of the text, identify which language it is written in, and possibly some forms of contextual information.

Variable name	Statistic	Typical Range
WORDS	Text length in words	31-9228
TT	Type token ratio	0.13-0.89
CPW	Average word length in characters	4.59-9.95
WPS	Average sentence length in words	2.45-63.1
P1	Proportion first person pronouns of words	0-0.105
P2	Proportion second person pronouns of words	0-0.020
P3	Proportion third person pronouns of words	0-0.060
IT	Proportion 'it' of words	0-0.044
NT	Proportion contractions: I'll, you're, etc.	0-0.033

Figure 13.1: Stylistic items under consideration.

This chapter will describe some experiments made as a groundwork to build a tool to use the stylistic cues found in electronically published texts as described in preceding chapters. The tool is intended to categorize or sort documents in an interactive information retrieval setting.

13.2 Text materials and stylistic items

To build a text corpus for use in this experiment we ran a set of typical TREC queries on the Altavista search engine and retrieved the top 60 returned pages[1]. These vary considerably in style. We will use one of those queries: "What is the economic impact of recycling tires?" as an example in the following discussion. Each text in the test material is processed to obtain — among others — the statistics for the items listed in Figure 13.1. In general, the items under consideration reflect variation of various kinds: lexical — where texts about the same subject can treat it with technical or lay vocabulary; syntactic — complex syntax may reflect more complex ideas or reasoning about a given subject; textual — texts can be in-depth treatments of a topic or overviews over several topics. An item, naturally, may, and most often will, relate to variation of several kinds simultaneously: "therefore" signifies a certain lexical choice as compared to "thus" but also a certain textual progression as compared to "and"; "tortious interference" not only has different flavor than "bad influence" but may suggest a different genre.

So, how can the variation of these items be combined to distinguish interesting types of document from each other? Simply picking a couple of parameters from the table, and plotting them against each other will give a first picture of the distribution of items. For this purpose, we developed Iksvit, a simple visualization tool written in Tcl/Tk.

13.3 Issue 1: Which items to display?

The first issue to address is to select which stylistic items to display: a useful strategy might be to pick a couple of parameters with a seemingly high spread, and see what the graph looks like. For the dimensions at hand there were some examples which seem to disperse the material quite well, as in Figure 13.2, and some which conversely let the texts stick together into a corner of the graph to a much higher extent, as in Figure 13.3.

13.4 Issue 2: How combine variables?

Now, we know that each of the variables in Figure 13.1 is of little consequence taken alone: even when they may have quite high descriptive power, using them for diagnostics may be risky. Random variation, and more distressingly, non-random intentional variation may obscure or obfuscate the variation we are interested in. Thus, using a combination of factors may be a better idea.

[1] This is a very small text corpus for this sort of experiment, and the results should be understood to illustrate the techniques used, rather than provide any information about texts on the Internet.

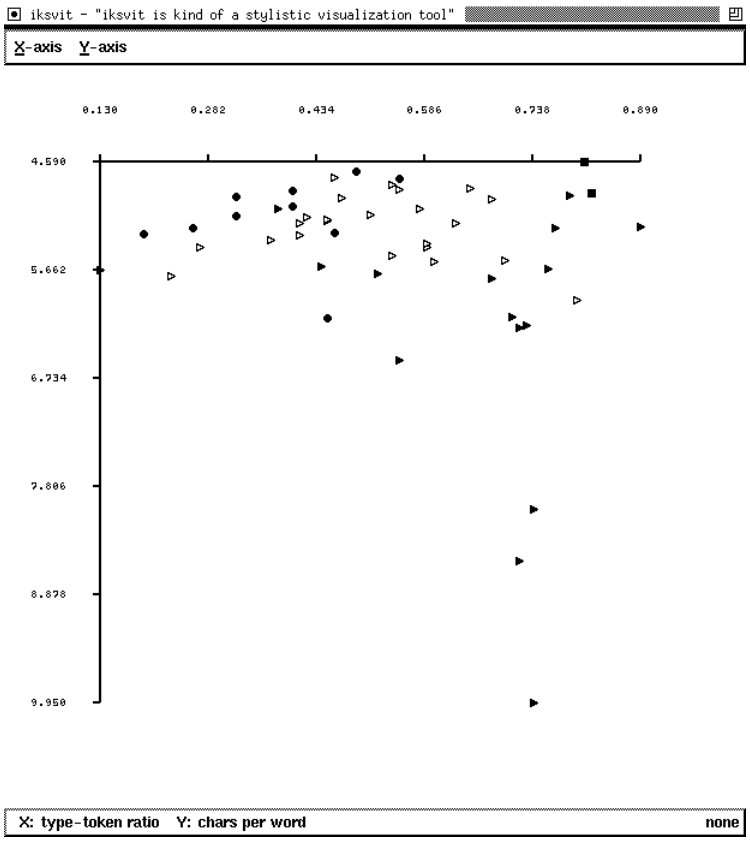

Figure 13.2: Average word length vs. type-token ratio.

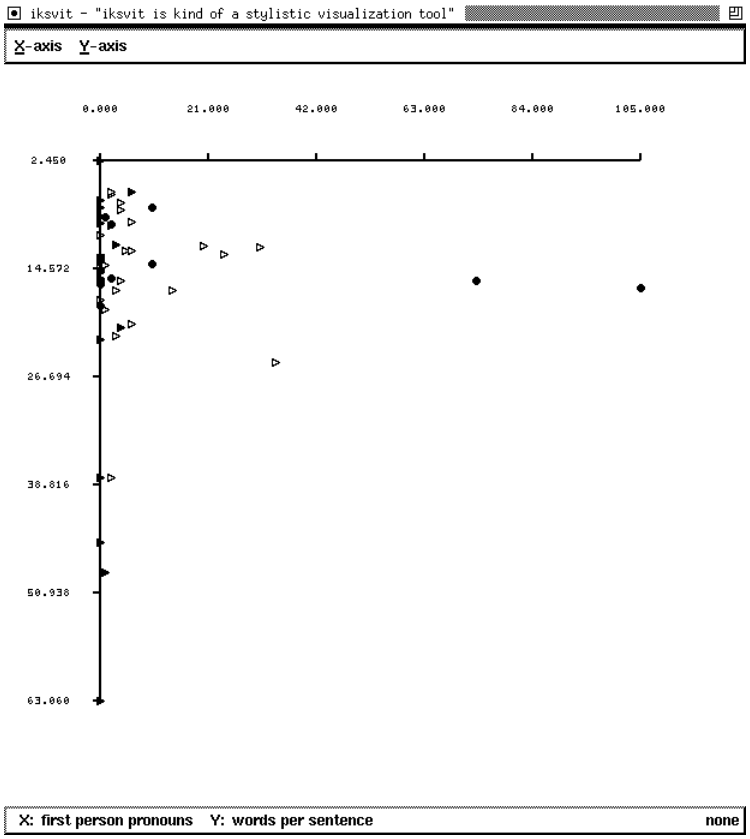

Figure 13.3: Proportion of first person pronouns in the text vs. average sentence length.

To reduce the dependence on any one of the parameters, common practice is to *weigh* them together in a linear sum. Weighing together information from a large number of *stylistic items* or *style markers* is straightforward but risky, as discussed in Chapter 10. Combining parameters by weighing them together is a common problem in many branches of science, and there is a battery of algorithms to do so automatically — but these algorithms tend to presuppose similar or identical distributions for each individual variable. In this case, somewhat riskily, I elected to test a standard technique for finding combinations of several variables that co-vary over a set of objects of study. While it is debatable whether simple linear score combinations of textual measurements capture the rather complex underlying interdependencies in text, this first test will help investigate the power of the variables.

13.5 Principal components analysis

The standard matrix algebraic technique *principal components analysis* is designed to reduce the number of variables if a large number of measurements is made on items under study, without omitting any variables. A smaller set of variables is used to approximate the original set. The new variables in the smaller set are called *principal components* and are designed to carry most of the information from the original set. The principal components are linear combinations of the various variables under study: a standard analysis for the data in this experiment yields the relative variable weightings displayed

Variable	PRIN1	PRIN2	PRIN3	PRIN4	PRIN5
WORDS	0.39	0.19	-0.32	0.14	-0.21
TT	-0.34	0.035	0.45	-0.0099	0.54
CPW	-0.12	0.53	0.36	0.70	-0.25
WPS	-0.075	0.63	0.19	-0.69	-0.30
P1	0.40	0.21	0.096	-0.0035	0.42
P2	0.27	-0.39	0.43	0.0032	-0.51
P3	0.45	0.13	-0.0043	-0.020	0.17
IT	0.44	0.18	0.026	0.053	0.22
NT	0.30	-0.22	0.58	-0.13	0.015
Proportion	0.50	0.14	0.12	0.09	0.08

Figure 13.4: First principal components.

in Figure 13.4 — in this case I used the SPSS package and its standard settings (SPSS, 1990). The weights indicate the relative importance of the variables — the variables are normalized first, so that their scale of variation will be similar.

The "proportion" row indicates how much of the total variance this component carries: in our case, the first component covers half of the total variation, and the second 14 per cent. The first two components thus cover about two thirds of the variation of the material; this is a way of guaranteeing a better dispersal than any of the single items under study. Using the two first components as dimensions IKSVIT yields the graph in Figure 13.5.

13.6 Issue 3: What should dimensions be called?

The problem with this otherwise interesting plot is that it may not be immediately useful as an information retrieval display: the dimensions are not readily translatable to plain English descriptors. Besides dubious methodological assumptions, this is another drawback of using statistical tools: the categories found and developed do not always conform to categories common in human usage. This gives us the third issue to think about: how to judiciously name the dimensions of variation for best possible effectiveness in communicating the structure of the collection to the user. This is where *genres* come in handy. The next chapter investigates the possibility of adding genre information to IKSVIT.

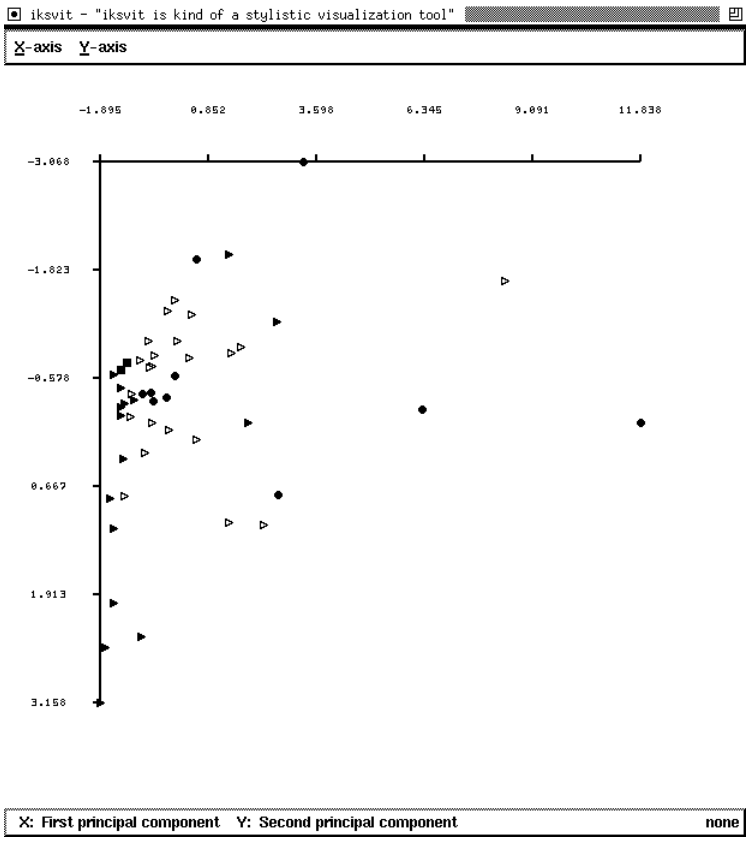

Figure 13.5: First two principal components.

Chapter 14

Genres and Visualization

The material in this chapter has previously been published as part of a paper by Troy Straszheim and me titled "Visualizing Stylistic Variation" in the Proceedings of the 30th Hawaii International Conference on Systems Sciences (Karlgren and Straszheim, 1997) and in a paper titled "Stylistic and Relevance" presented to the 2nd International Conference on New Methods in Natural Language Processing (Karlgren, 1996b). It describes work done by me during 1996, while I worked in the Proteus project at New York University.

14.1 Aim of these experiments

Plotting the texts with the two variables against each other we get the graph given in Figure 13.5. The problem with this otherwise interesting plot is that it may not be immediately useful for information retrieval: as discussed in Chapter 9 and Chapter 13, the dimensions found by statistical data organization methods are typically not readily translatable to plain English descriptors.

14.2 Genres and stylistic items in scatterplots

There are no objectively defined genres for our type of material; what genres we want to make use of will depend on the domain of discourse, the data we have recourse to, and what stylistic items we have chosen. Above all, they will depend on reader preferences or our perception of the readers' information needs.

For the purposes of this experiment I made a rough hand-categorization of the texts used in Chapter 13.[1] In the material there were database listings, error messages, technical texts, journalistic texts, commercial texts, legal texts, announcements, forms, and various other textual and non-textual material. Since the material is small, I divided it

[1]Conceivably I could have used automatic methods such as *clustering techniques* to do the same. I would then have ended up with the same problem as for factorial analysis or principal components analysis: I would have had descriptively interesting categories which would be difficult to explain to the reader. Instead, I went with sloppily manually defined categories based on *text function*.

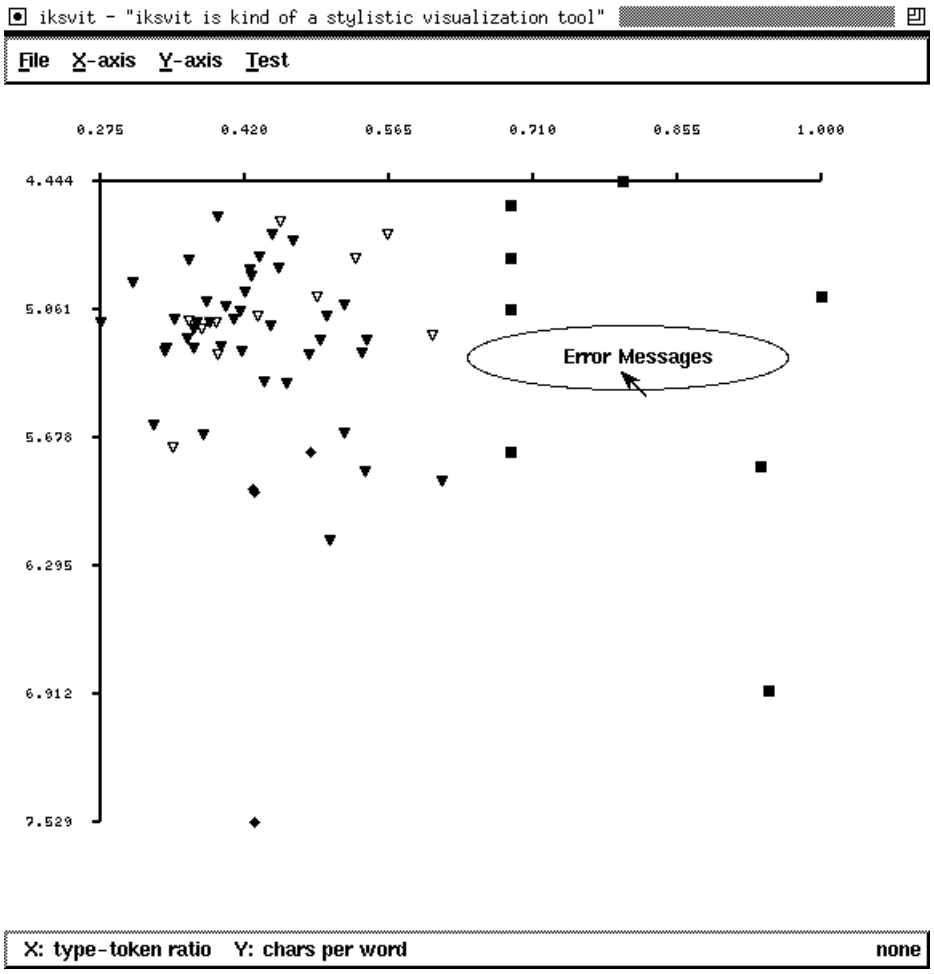

Figure 14.1: Plot of type-token ratio vs word length.

into four quite broad categories: proper text (white triangles), database listings and lists of links (black triangles), governmental announcements (black circles), and error messages (black squares) as shown in the figures of this chapter.

The graphs displayed in the previous chapter show how the genres emerge quite nicely in figure 13.2, whereas the pattern is much less clear in the other two figures.

14.3 Balloon help

If the dimensions used are chosen appropriately, regions of the plotted area will show consistency genre-wise. This is true for some combinations of dimensions, and Iksvit can show pop-up help in some areas, to display which type of text is prevalent there. A mouse click towards one end of the plotted area will give an indication of which types of document can be found there. In the example shown in Figure 14.1 error messages have a larger type-token ratio than other texts — meaning they are short. A different collection, composed of articles from the Wall Street Journal, clusters "Business Briefs" — one paragraph notes on some current issue — similarly, as shown in Figure 14.2: they are short with exceptionally many digits.

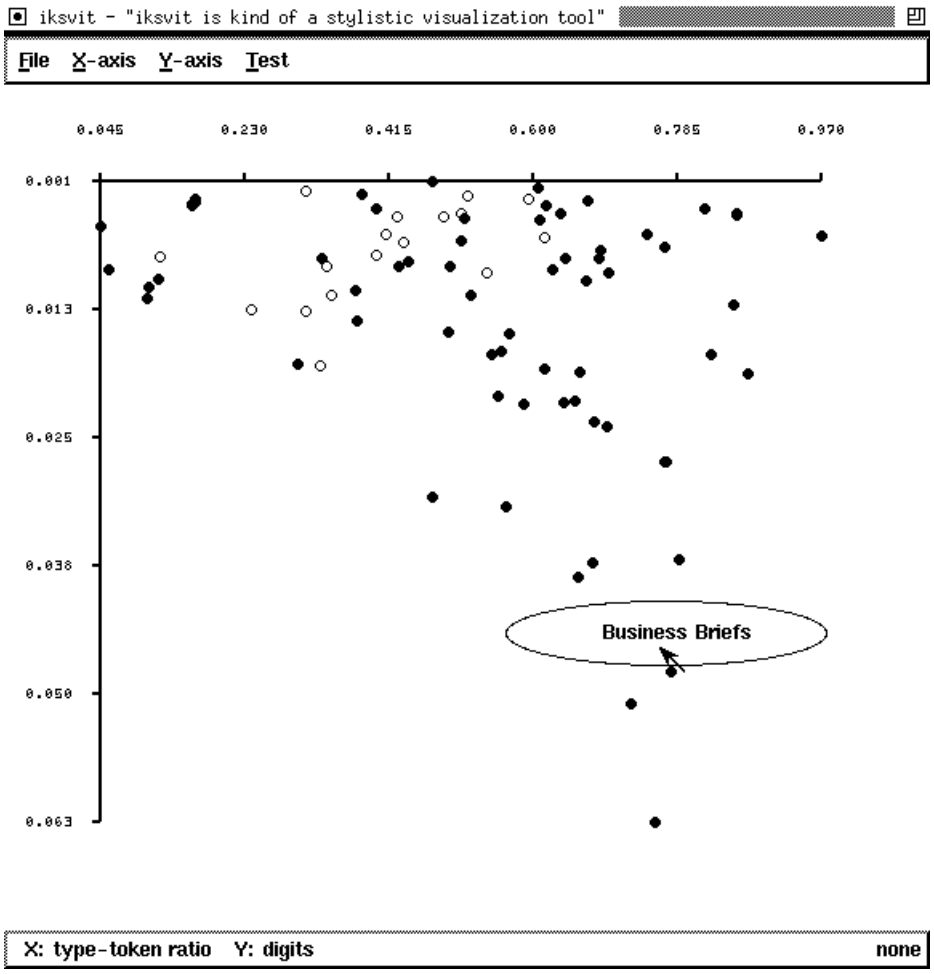

Figure 14.2: Plot of type-token ratio vs number of digits in text.

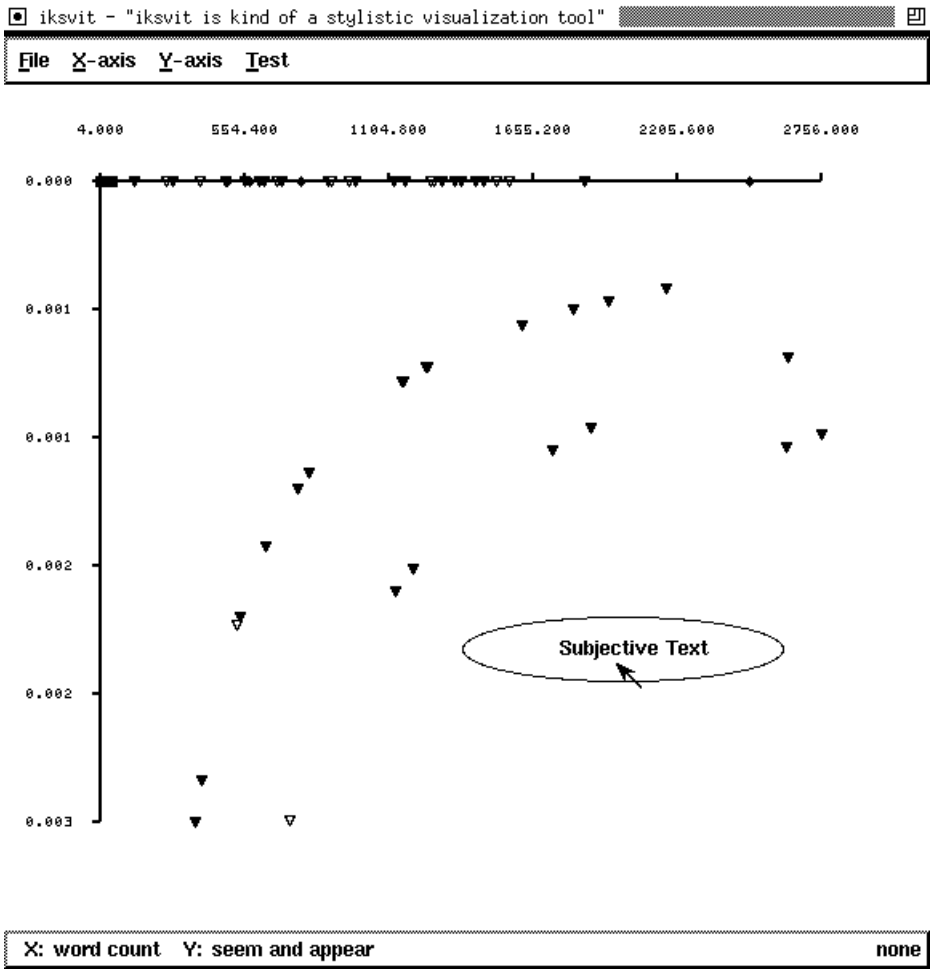

Figure 14.3: Plot of "seem" and "appear" vs text length.

14.4 Genre determination and hierarchies

The previous two genre examples were defined by *text function*, but genres are multi-faceted things. The plot in Figure 14.3 shows how text length and the presence of the "subjective marker" verbs *seem* and *appear* can be used to identify in some sense "subjective" texts.

This is an illustration of the problem mentioned above in Section 9.2, namely of what a genre is. Genres can be defined simply by text source, functionally by intended or observed purpose, linguistically descriptively by style, by channel or by any number of criteria. The next chapter will describe an empirical method of gathering user impressions and collating them informally into a useful palette of genres.

14.5 Conclusions

It is clear from experiments so far that the notion of collecting documents into *informative* categories will be useful for the purposes of interactive information retrieval — the reader's

task of selecting some documents for further processing and perusal needs tools that do not involve wading through lists of tens of thousands of documents. One such handy categorization, as has been discussed in several chapters so far in this text, is that of *genre*.

To get explanatory power, a genre analysis of the exemplified kind must be designed to make use of the most informative dimensions of textual variation. The algorithmically best choices may be too dependent on the variables chosen to give a useful and explanatorily powerful display of the textual material at hand.

But the most important issue is what genres one can assume are *salient* in the collection at hand, and how to *name* them. The next chapter tries to determine this for one specific combination of reader category and document collection.

Chapter 15

Assembling a Balanced Corpus from the Internet

The material in this chapter has previously been published as part of a similarly titled paper by Johan Dewe, myself, and Ivan Bretan, presented to the 11th Nordic Conference on Computational Linguistics (Dewe *et al.*, 1998). It describes work done in the DropJaw project, initiated and supervised by Ivan Bretan at Telia Research. The work presented here was planned and supervised by me; most of the data collection was done by Johan Dewe as part of his M Sc thesis project.

15.1 Balanced corpora for textual research

For empirically oriented textual research it is crucial to have material available for extraction of statistics, training probabilistic algorithms, and testing hypotheses about language and language processing in general.

In recent years, the awareness that text is not just text, but that texts come in several forms, has spread from more theoretical and literary subfields of linguistics to the more practically oriented information retrieval and natural language processing fields. As a consequence, several test collections available for research explicitly attempt to cover many or most well-established textual *genres*, or *functional styles* in well-balanced proportions (Francis and Kučera, 1982; Källgren, 1990).

The creation of such a collection is a complex matter in several respects. The object of the next few chapters is to build retrieval tools for the Internet, and thus, for our purposes, the choice of genres to include is one of the more central problems: there is no well-established genre palette for Internet materials. To have materials to experiment with, we need to find them, and then collect and collate them in a form suitable for our purposes. This collation task has to establish categories based on vaguely expressed user expectations, and define recognition measures using large numbers of computable features which taken singly have low predictive and explanatory power. This chapter gives an outline of the methodology we use for determining which genres to include.

15.2 Establishing genres

15.2.1 Method

In previous similar studies the categories of test corpora were less suitable for our purposes, as outlined in Section 9.2. For this study, we wished to have a firm foundation for our genre palette. Our basic souce of knowledge is interviewing users about their perceptions of what types of material they find and interact with online. We collate the impressions and try to define genres that are both reasonably consistent with what users expect and observable and conveniently computable using measures of stylistic variation.

15.2.2 Questionnaire

The questionnaire in Figure 15.1 was sent to 648 computer users — students, researchers, and teachers at Stockholm University and the Royal Institute of Technology. We received 7 error messages and 67 responses, which gives a response rate of 10 per cent.

> Hi. I need two minutes of your time.
> For my M Sc project I will classify WWW documents by genre.
>
> What is a genre? A genre is a group of documents with similarities as regards form. Journalistic material, for instance, gives us several examples of genres. We find scientific materials, short stories, news items, advertisments, and so forth. In a larger perspective a newspaper itself is a genre, as compared to crime fiction, parliamentary records, and chat group text.
>
> Similarly, it should be possible to categorize materials from the WWW in genres. The obvious ones I can figure out myself, but I do not want to constrain myself to a single perspective. So I need your help to gain a wider view:
> * What genres do you feel you find on the WWW?
> Take a minute to think about the question, and send me a list of the genres that occur to you. All replies are useful to me!
>
> Thank you for your time,
> /Johan Dewe, d92-jde@nada.kth.se

Figure 15.1: An English translation of the genre questionnaire.

15.2.3 Compiling the results

The answers ranged from very short to extensive discussions — some examples are shown in Figure 15.2. It was very clear to us that most readers conflated genre and form on the one hand with content and topic on the other: "tourism", "sports", "games", "adult pages". This is not surprising. Genre and topic are not independent dimensions of variation, and a typical library categorization reflects both dimensions simultaneously. Several respondents did give examples of more cleanly form-oriented genres as well: "home pages", "data bases", "FAQs", "search pages", "reference materials". Some respondents gave explicit references to *paper genres* — one lengthy quote is given among the examples in Figure 15.2. The *intention* of the information provider showed up as a genre formation

- Science, Entertainment, Information
- Here I am, Sales pitches, Serious material
- Home pages
- Data bases
- Guest books
- Comics
- Pornography
- FAQs
- Search pages
- Corporate info
- Product info
- Reference materials
- My immediate reaction is that genres from general society will be found on the WWW as well. We get stuck in old conventions. ... e.g. e-mail conventions follow paper letter conventions. I would start by using genres from ordinary life and see if they are applicable to WWW.
- Public info
- Non-government organization info
- Corporate info
- Informative advertisements
- Non-informative advertisments
- Research materials
- Games and pornography
- News
- Economic info
- News
- Tourism
- Sports
- Games
- Adult pages
- Science
- Culture
- Language
- Public documents, Internal documents, Personal documents
- Information
- "Check out what a flashy page I can code"
- "I guess we have to be on the net too"

Figure 15.2: Some translated excerpts from the answers to the questionnaire

criterion in several responses: "here I am", "sales pitches", "serious material"; or, as an alternative formulation of the same criterion, the type of author: "commercial info", "public info", "non-governmental organization info". Some responses explicitly brought up *quality*: "boring home pages" and text *ecology* or intended environment: "public documents", "internal documents", "personal documents".

We have attempted to systematize some of the user-perceived distinctions, namely those that are predictable enough to be modeled with simple metrics, in the genre palette shown in Figure 15.3.

Informal, Private Personal home pages.

Public, commercial Home pages for the general public.

Searchable indices Pages with feed-back: customer dialogue; searchable indices.

Journalistic materials Press: news, reportage, editorials, reviews, popular reporting, e-zines.

Reports Scientific, legal, and public materials; formal text.

Other running text

FAQs

Link Collections

Other listings and tables

Asynchronous multi-party correspondence Contributions to discussions, requests, comments; Usenet News materials.

Error Messages

Figure 15.3: The experimental genre palette.

When trying to assign textual materials to the various categories automatically we expected to find that some genres would be less easy to recognize than others. This hypothesis was confirmed in further experiments; we indeed found that the categories are useful only as a starting point and for any practical application or even further experimentation they will need to be adjusted — merged, split, or redefined. The categories shown in Figure 15.3 are starting points for research: the method for defining them is general; the results are not.

15.3 Finding samples

We used three methods to collect data from the World Wide Web.

First, we took fifty TREC queries (queries no. 251-300; fields "topic" and "description") and ran them through standard WWW search services. We then collected the top ten hits for each query. This gave us 386 documents. Second, we collected sixty queries from Magellan, another search service on the Internet. Magellan provides a "voyeur page" which displays real user queries in real time. We ran the sixty queries through Magellan,

and similarly obtained about 500 documents. Thirdly, we used colleagues' history files from to retrieve about 500 additional documents for a total of 1 358.

	Genre	URL source			
		TREC	Magellan Voyeur	History List	Total
1	Informal, Private	11	67	50	128
2	Public, Commercial	23	87	87	197
3	Searchable indices	4	14	55	73
4	Journalistic materials	50	28	16	94
5	Reports	106	5	2	113
6	Other running text	73	49	38	160
7	FAQs	0	4	8	12
8	Link Collections	31	50	67	148
9	Listings, Tables	17	138	70	225
10	Discussions	16	0	8	24
11	Error Messages	55	36	93	184
	Total	386	478	494	1358

Figure 15.4: The composition of the training corpus.

15.4 Evaluating the choice of genres

To evaluate the genre palette we sent out the list of genres we settled on to the same recipients we originally solicited the genre distinctions from, with a question if they understood what the genres represented and if any obvious genre was missing. We received 102 responses. Most respondents claimed to understand what type of text our genre labels were intended to cover, and while most categories got some comments of one form or another, most comments were caused by our giving too few examples of what the genres were intended to cover. Most comments concerned the category "Interactive pages". Many respondents were annoyed by the fact that the category was of another type than the others. Some respondents objected or did not understand the labels — e.g. "FAQ" or "Listings, tables" or "Error messages"; many asked for a download page or FTP database category; some wondered about the all-inclusiveness of "Other running text"; several asked for a specific category for "Search engines"; several suggested more content-based genres.

Many pointed out that some of the categories were less suitable for search in that they did not imagine themselves ever searching for "Error messages" or "Interactive pages" specifically. Several respondents pointed out that the categories were not mutually exclusive. In summary, the most central objections were either such that would be remedied in an interactive situation where examples are readily available, or requests for more flexible genre assignment. We decided to disregard the comments, and considered the genre palette suitable for the purposes of our experiment.

15.5 Conclusions

We found Internet users do have a vague sense of genres among the documents they retrieve and read. The impressions users have of genre can be elicited and to some extent formalized enough for genre collection. The names of genres should be judiciously chosen to be on an appropriate level of abstraction so that mismatches will not faze readers. Most likely, *consistency* and *formality* are less useful quality criteria for the categorization than is *apparent clarity*.

Chapter 16

Stylistic analysis integrated into an interactive system

The material in this chapter has previously been published as part of a paper by Ivan Bretan, Johan Dewe, Anders Hallberg, Niklas Wolkert and myself titled "Web-Specific Genre Visualization" presented to the Webnet conference in Orlando, Florida in November 1998 (Bretan *et al.*, 1998) and also as part of a paper by the same set of authors titled "Iterative Information Retrieval Using Fast Clustering and Usage-Specific Genres" presented to the tenth DELOS workshop on Digital Libraries in Stockholm in October 1998. (Karlgren *et al.*, 1998) It describes work done by the Dropjaw project group at Telia Research which was initiated, supervised, and organized by Ivan Bretan and consisted of me, in the role of technical expertise, Anders Hallberg and Johan Dewe who as part of their M Sc projects (Dewe, 1998; Hallberg, 1999) implemented the system described here and did the user evaluation, and Niklas Wolkert who did the graphic design of the interface. The project was awarded *Framsteget* for 1998, an annual prize given for the most innovative Swedish research project in information technology by *Ny Teknik*, Sweden's leading engineering weekly.

16.1 An integrative prototype design

This chapter describes how genre prediction — defined empirically for a specific collection and user population as described in the previous chapter — can be used in an interactive interface when it is integrated with other types of search functionality.

Stylistic analysis and genre recognition as described in the experiments in the previous chapters show promise of being useful in information retrieval tasks, but not on their own. The information a stylistic analysis tool can provide must be integrated in a retrieval interface and in the information seeking task environment. In general, interface designers try to shield users from excess information and the complexity of tasks, and build their interfaces accordingly. Adding stylistic information to vanilla information retrieval interfaces would clutter up the display: they are not built to handle complex information and multi-faceted tasks. If complex information is presented in a manner well founded in the task and usage context, users will be able to use and accept it. To

be able to include stylistic analysis and genre categorization in an interface, it must be designed with this in mind.

16.2 Addressing two bottlenecks

The DropJaw project focussed on two of the specific drawbacks of the traditional information retrieval search process discussed in prevous chapters: Firstly, searches are seldom one-shot affairs. Typically a search is improved and refined iteratively, until the retrieved set seems good enough, by some metric. Thus, the interface should support persistence and incremental refinement. Secondly, as has been indicated in previous chapters, the objects of study are more complex than usually is assumed: information access systems can learn more about documents than term frequencies and must be built to utilize this knowledge.

Easify is an interface built in the Dropjaw project, for the express purpose of utilizing a richer representation of retrieval results in the search interface. Fundamental to the interface are the notions of *document grouping*, *incremental search*, and *æsthetic design*. Easify uses Chunkify, a rapid topical clustering tool integrated with the stylistic genre-based document categorization, and displays retrieval results in a highly interactive visualization front-end.

The DropJaw prototype bases its searches for web documents on the user entering terms, as in a traditional system. Rather than producing ranked lists of output based on term occurrence, Easify displays the distribution of the resulting set over two dimensions: dynamically generated topical clusters and user-defined, document-base oriented genre. The two-dimensional document space is displayed on a work board or matrix for further user processing. The work board is a surface for the display of conventionally determined and represented document categories, labeled with genre and topic — in contrast to map display approaches where the information space is mapped directly to the display space (Lin, 1997, e.g.). IKSVIT, as described in Chapter 13 is an example of a map display, and exemplifies the drawbacks of the approach: the space is difficult to interpret for the untrained user.

The DropJaw prototype is implemented in C++ and runs as a stand-alone application under Microsoft Windows. It does not include a indexing or search engine — currently it makes use of any of several commercial search engines as indexers. DropJaw retrieves the documents the indexing service returns and reanalyzes them locally. The architecture is a pipeline model, where different processing components in Chunkify can be connected or disconnected — by user requests to Easify — to the pipeline which delivers a stream of documents from the World Wide Web to the user.

16.3 Appearance

Information visualization tools tend to be designed for specialist users, using elements from standard programming tools and little or no involvement of graphic or industrial designers in the design process. IKSVIT, in Chapter 13, is an example result. By contrast,

information visualization before the computer era (Tufte, 1983), in many cases is designed to be æsthetically interesting and pleasant to look at — without compromising clarity or usability. There is no reason why the introduction of computers into the information visualization process should compromise æsthetics. Lack of æsthetic qualities may distract and bore the user and indirectly cause the loss of potentially useful information; usability is a function of both underlying functionality and æsthetic quality.

For this reason, the DropJaw project put effort into the design of Easify: it is designed to feel playful and fun but without sacrificing a sense of strict reliability. Because of the high information density in the interface, soft sober colors and simple shapes built from non-standard graphic elements have been used to help user choice, without dominating the interface look.

16.4 Information seeking dialog

As has been discussed in Chapter 5, seeking information is seldom done with a clear picture of the goal in one's mind. Typically users will need to familiarize themselves with the available material to find out how they will express their information need to the system — by some preliminary searches, some back-tracking, and some reformulation of previous tries. An information retrieval interface should support persistent and manipulable dialog objects which represent the system's understanding of the search.

Easify has a dialog designed to transcend the typical one-shot dialogs of most information access systems available today: "enter your search terms" — "scan the resulting list". The Easify dialog builds on incremental specialization. The system information flow is based on a pipeline: a first query retrieves a set of documents, and users can specialize the query by working on subsets of the resulting set, even before the system has completed its first search.

Users initiate the interaction by entering a query and clicking the search button. Easify requests Chunkify to start the background pipeline. Chunkify consults the indexing engine for a list of likely documents, and starts retrieving candidate documents from the Internet for categorization. After clustering and categorization, Chunkify starts delivering documents to Easify for presentation.

The pipeline design leaves the user in control of the interface at all times instead of locking the interaction: there are indicators to show that the classification engine is running in the background. A stop button lets the user halt the background processes; a clear button clears the current document set from the display.

The query and the resulting clusters are represented continuously: retrieved documents are inserted in the appropriate cluster irrespective of ranking. The clusters represent topic and genre by position on the work board: each individual cluster can be inspected to see which documents it includes, and each document can be selected for display in a standard HTML browser. Specialization of a query can be done without formulating new search terms, since the clusters can be used instead: each cluster can be dragged to a regroup panel together with other clusters for reclustering at any stage in the processing — even before the entire set has been retrieved and processed.

1. Private, informal text
2. Public, commercial text
3. Discussions
4. Journalistic text
5. Reports — Technical and scientific text
6. Other text
7. Lists and tables
8. Interactive pages and forms
9. Link collections
10. FAQs

Figure 16.1: The genre palette.

The first filter in the pipeline continues processing: DropJaw simply adds a finer-grained Chunkify filter to the end of the pipeline. The first set keeps accruing, and can be returned to if the finer analysis turns out less useful.

16.5 Document representation

Easify represents documents as members of topical clusters, defined for the document set at hand, and stylistically homogenous genre clusters. The genre palette is based on the study from Chapter 15 and shown in Figure 16.1. The genre recognition rules are built using C4.5 as exemplified in Chapter 10 and some examples are listed in Figure 16.2. We attempted to define genres both reasonably consistent with what users expect, and conveniently computable using measures of stylistic variation as outlined in the previous chapters.

16.6 Fast clustering by content

The similarity measure for comparing topical document representations with each other and with cluster centroids is in most respects based on a standard term frequency metric, as defined by Salton and Buckley (Salton and Buckley, 1988): standard term frequencies, cosine length normalization, and a standard collection frequency measure to factor in collection and domain specific terminology variation.

Since the emphasis is on a high degree of interactivity, a quick and dirty clustering must be used for the initial document sets. The assumption of the project is that a low number, up to 5, of clusters in the interface is desirable (Hearst and Pedersen, 1996).

Since the search engine itself is not included in the system setup, there is no time to wait for all the data to arrive. If the algorithm would have recourse to the entire document set, the initial clusters could be formed from a random set, as in Scatter/Gather (Cutting *et al.*, 1992); in our case the first clustering must proceed on the assumption that the first documents to arrive are a representative subset of the entire retrieved set. This is a daring assumption and most likely overly optimistic, but enables us to start clustering sooner, and to restrict the use of the computationally expensive — on the order of N^2 — hierarchical clustering by defining the first clusters on a small number of documents: the first 10-50 documents, which number is adjustable in the interface. The clustering itself is a variant of the standard metrics: a hierarchical agglomerative group-average algorithm (Tatsuoka, 1971, e.g.).

16.7 Genre recognition rules

The genre palette, besides being intuitively understandable, needs to be workable for automatic analysis. We calculate a quite large number of textual features for each individual text and work them together for a categorization decision using a machine learning algorithm.

The stylistic items under consideration in this prototype are based on the experimental data described in previous chapters, with extra features added, vectored specifically to the Internet material we have been using for experimentation: number of images or number of HREF links in the document, for instance. We normalize the measurements by mean and standard deviation, and combine them — 40 of them, at present — into simple if-then categorization rules of the type found in Figure 16.2.

We have a few dozen rules to categorize texts into one of the eleven genres defined in Chapter 15. The genres partition into two major hypercategories: textual and non-textual; each of them in turn splits to one of the sub-categories. These splits are of varying quality: the textual vs. non-textual split does quite well, something like a ninety per cent success rate, while the subsplits make the wrong choice somewhere between once in three or four times. The example rules in Figure 16.2 show why: some of the specific subcategories are defined relatively closely after the training corpus, and are too specific. While the second rule makes sense to some extent: technical text can be understood as having fewer pronouns and more suasive verbs than other texts — the third does not seem work out as reasonably: journalistic texts do not necessarily have longer words and less links than other texts on the WWW. These rules need to be retrained frequently: besides relying on the test corpus, they grow obsolete fast. With the advent of new HTML authoring software, difference in HTML coding standards cannot be relied on for distinguishing between private and public pages.

16.8 Evaluating method and design

Evaluating interactive information access systems is known to be difficult (Belkin, 1998). There are no standard metrics and variation in real-time factors can completely swamp any beneficial effects a presentation prototype might engender.

```
If there are
- more "because" than average,
- longer words than average,
- more different words than average,
then
- the document is of class Textual.

If there are
- less capitalized words than average,
- fewer first person pronouns than average,
- fewer second person pronouns than average,
- more suasive verbs than average
then
- the document is of class Reports

If
- there are less WWW links than average,
- longer words than average,
- more different words than average,
then
- the document is of class Journalistic text
```

Figure 16.2: Example rules.

In the case of Easify, the interactivity of the system suffers from network time lag: the system now consults some of the major WWW search engines, retrieves the results, retrieves the documents from the WWW one by one, processes them, and only then can start displaying the results. This makes comparison based on measures of *interaction time* quite difficult. Other measures could include *number of interface actions* or *number of backtracking actions* to attain a goal, but since we wished to encourage interactive experimentation we wanted these measures to give high scores, whereas most interfaces are built to minimize clicks and backtracks. We performed a very small semi-formal evaluation, to gather user impressions of the interface.

16.8.1 Subject satisfaction

A small number of subjects were given two simple retrieval tasks each, one using Easify and the other using Altavista, one of the standard WWW information retrieval services. The order between questions and interfaces was varied between subjects. 12 subjects were chosen to be a representative test population. 6 subjects were male, 6 female; all of age 25-30; and while averages are a poor measure for small populations, in this case the subject pool hit the mark quite well: they assessed themselves as averagely experienced internet users who use search services occasionally, and have a reasonable understanding of how search services work.

The tasks were

- Find an album or a concert review about Oasis.
- Find a list of hotels on Malta.

The subjects were given an introduction to the main ideas behind Easify, and shown an example search with the system: all users had used Altavista previously. They were then given the tasks. The experiment supervisor gave tips on query formulation if the subjects seemed to get stuck. The subjects used about 5 minutes on each task.

Most subjects used the interface as intended, and many searched for documents in the genres the results could be expected to show up in. The subjects liked and understood the interface prototype, with some remarks: some subjects were confused by the changing cluster texts, but were comfortable with them after a short explanation; some wanted the genres to be refined as well, to sub-genres. In spite of long response times, all but one of the subjects liked the interface in itself and preferred it to the standard Altavista view.

16.8.2 Technical quality

The genre determination algorithm in its current state makes a mistaken choice too often for some of the genres: the hit rate must be raised to nine out of ten for the concept to work better. Flexible genre determination would help here — a document should be allowed to fall into several genres rather than exclusively one, and the genres must be revised for this purpose.

But more importantly, Easify highlighted a major weakness in information access presentation: clusters of documents need to be described in some compact manner. Multi-document summarization is a necessary technology for this type of interface: providing a few cluster centroid keywords is not sufficient for information access purposes. Multi-document summarization is a task very different from single-document summarization — rather than asking the question "what is special about this document?" it asks the question "what do these documents have in common?" and there are no tools to address this task at present.

Current work with implementing an Easify-like interface in an existing partly hand-categorized Swedish language digital collection investigates the possibility of capitalizing on the qualitative work done by manual indexing by primarily clustering the hand-categorized documents in a suitable number of clusters, using the manually assigned keywords as representation of the topical clusters, and directing uncategorized documents to the cluster nearest them in terms of term frequency.

16.8.3 Developer satisfaction

With rather small effort, this both user- and technique-centered development effort has pinpointed certain weaknesses in the design while at the same time encouraging the development team to pursue the strong sides of the design further. Since most components are based on empirically evaluated knowledge, they proved immediately useful.

Part V

Concluding Remarks

Chapter 17

Conclusions, Encouraging Observations, and Shortcomings

17.1 Summary of results

Typical bare bone term-based information retrieval has weak points. As outlined in the initial chapters on the basics of information retrieval, I feel strongly that

1. text can yield more information than word statistics — the hindrance to understanding text better is not faulty statistics but faulty understanding of what text is and why; and
2. the information flow between user and system is too poor.

The line of research I have pursued in the reports presented in this dissertation text has addressed these informational bottlenecks. This I have done with reasonably encouraging results — they certainly warrant some thought about further application.

17.1.1 Bottleneck 1: The system's view of text

On the first count, the experiments show that it is possible to extract non-topical information from texts with comparatively little bother: stylistic analysis can be done on the cheap. While it is true that you get what you pay for, even shoddy analysis is informative: texts can be categorized in genres, provided the genres in question are well chosen.

Furthermore, stylistic information has been used to improve the results of ranking retrieved documents through inferences by which types of document are likely to be preferred by users. This procedure has been formally evaluated and shows great gains for some types of query; neutral results for most queries; and a clear loss for some. Whether the total adds up to gain or loss is a question of which metric is put into play.

17.1.2 Bottleneck 2: The system's presentation of search results

On the second count, this information has been used in a design for better presentation of information retrieval results. The underlying rule set is less than perfect, the design has not been formally evaluated — and the prototype certainly is only a first sketch of how such information could be utilized. However, test users are satisfied with the system, and prefer it to traditional list presentations. This in fact is an implicit argument against the traditional view of relevance: texts are not sorted in order of falling presumed relevance, but by various other categorizations.

It is clear at this point that the information flow from user to system needs to be similarly broadened to make use of this type of information appropriately. Users will need to get unobtrusive help in posing a more elaborate expression of information need.

17.2 More than a thousand words

The present experiments are designed not to be a definitive word on the retrieval systems of tomorrow, but a first step towards a broader view of what text is and why we bother writing it. This is accomplished here by opening another window to information about text and using it in a context where normally text content would be understood as the sole useful factor for classification and retrieval.

But there is a larger picture. Indeed, in my mind, the picture is larger than more and finer statistical methods for text categorization: I believe text is not the definitive word on information dissemination.

A picture says more than a thousand words, and for me the issue is how to listen to the picture text paints, without being distracted by its words.

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.

Douglas Biber. 1988. *Variation across speech and writing*. Cambridge University Press, Cambridge, UK.

Douglas Biber. 1989. "A typology of English texts". *Linguistics*, 27:3–43.

Ivan Bretan, Johan Dewe, Anders Hallberg, Niklas Wolkert, and Jussi Karlgren. 1998. "Web-Specific Genre Visualization". In *Proceedings of the Webnet World Conference on the WWW and Internet*, Orlando, Florida.

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

Chris Buckley, Amit Singhal, Mandar Mitra, and Gerard Salton. 1995. "New Retrieval Approches Using SMART: TREC 4". In Donna Harman, editor, *Proceedings of the 4th Text Retrieval Conference*, National Institute of Standards and Technology, Gaithersburg, Maryland, November.

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

Jeanne S. Chall. 1948. *Readability.* Ohio State University.

Naoufel Ben Cheikh and Magnus Zackrisson. 1994. "Genrekategorisering av text för filtrering av elektroniska meddelanden (Genre Classification of Texts for Filtering of Electronic Messages)". Bachelor of Art Thesis, The Royal Institute of Technology and Stockholm University, Dept. of Computer and Systems Sciences, Stockholm, Sweden. (in Swedish).

Kenneth Church. 1988. "A Stochastic Parts of Speech and Noun Phrase Parser for Unrestricted Text". In *Proceedings of the 2nd Conference on Applied Natural Language Processing*, Austin, Texas, February. ACL.

Douglass Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. 1992. "A Practical Part-of-Speech Tagger". In *Proceedings of the 3rd Conference on Applied Natural Language Processing*, pp. 133–140, Trento, Italy, April. ACL.

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.

John Dawkins. 1975. *Syntax and Readability.* International Reading Association, Newark, Delaware.

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.

Johan Dewe. 1998. "En prototyp för att klassificera dokument från WWW med avseende på genre och ämne (A prototype for classifying WWW documents in terms of genre and topic)". Master of Science Thesis, The Royal Institute of Technology, Department of Numerical Analysis and Computing Science, Stockholm. (in Swedish).

Johan Dewe, Jussi Karlgren, and Ivan Bretan. 1998. "Assembling a Balanced Corpus from the Internet". In *Proceedings of the 11th Nordic Conference of Computational Linguistics*, University of Copenhagen, Copenhagen, Denmark, January.

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.

W. N. Francis and F. Kucera. 1982. *Frequency Analysis of English Usage.* Houghton Mifflin.

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.

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.

Anders Hallberg. 1999. "Visualisering och manipulering av genreklassificerade sökresultat (Visualization and manipulation of genre classified retrieval results)". Master of Science Thesis, The Royal Institute of Technology, Department of Numerical Analysis and Computing Science, Stockholm. (in Swedish).

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.

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.

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

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

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.

Marti Hearst and Jan Pedersen. 1996. "Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results". In *Proceedings of the 19th International Conference on Research and Development in Information Retrieval*, Zürich, Switzerland, August. ACM SIGIR.

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.

Fahima Polly Hussain and Ioannis Tzikas. 1995. "Ordstatistisk kategorisering av text för filtrering av elektroniska meddelanden (Genre Classification of Texts by Word Occurrence Statistics for Filtering of Electronic Messages)". Bachelor of Art Thesis, The Royal Institute of Technology and Stockholm University, Dept. of Computer and Systems Sciences, Stockholm, Sweden. (in Swedish).

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.

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.

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.

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.

Jussi Karlgren. 1990b. "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.

Jussi Karlgren. 1996a. "Assessed Relevance and Stylistic Variation". In *Proceedings of the 19th International Conference on Research and Development in Information Retrieval*, Zürich, Switzerland, August. ACM SIGIR.

Jussi Karlgren. 1996b. "Stylistic Variation in an Information Retrieval Experiment". In *Proceedings of the 2nd International Conference on New Methods in Natural Language Processing*, Bilkent University, Ankara, Turkey, September.

Jussi Karlgren. 1999. "Stylistic Experiments in Information Retrieval". In

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

Jussi Karlgren, Ivan Bretan, Johan Dewe, Anders Hallberg, and Niklas Wolkert. 1998. "Iterative Information Retrieval Using Fast Clustering and Usage-Specific Genres". In Preben Hansen, editor, *Proceedings of Eighth DELOS Workshop on User Interfaces in Digital Libraries*, pp. 85–92, Långholmen. ERCIM.

Jussi Karlgren and Douglass Cutting. 1994. "Recognizing Text Genres with Simple Metrics Using Discriminant Analysis". In *Proceedings of the 15th International Conference on Computational Linguistics*, volume 2, pp. 1071–1075, Kyoto, Japan, August. ICCL.

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

Jussi Karlgren and Troy Straszheim. 1997. "Visualizing Stylistic Variation". In *Proceedings of the 30th Hawaii International Conference on Systems Sciences*, Maui, Hawaii, January. IEEE.

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

Ferenc Kiefer, editor. 1980. *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.

George R. Klare. 1963. *The Measurement of Readability.* Iowa University Press, Iowa.

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.

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.

Jiři Kraus and Josef Polák. 1967. "Text factors and characteristics". In *Prague Studies in Mathematical Linguistics*. University of Alabama Press.

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.

Gunnel Källgren. 1990. "'The first million is hardest to get': Building a Large Tagged Corpus as Automatically as Possible". In Hans Karlgren, editor, *Proceedings of the 13th International Conference on Computational Linguistics*, volume 3, pp. 400–404, Helsinki, Finland, August. ICCL.

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*.

David Lee. 2000. "Modelling Variation In Spoken & Written English: The Multi-Dimensional Approach Revisited". Doctor of Philosophy Thesis, Lancaster University, Lancaster, England.

Xia Lin. 1997. "Map Displays of Information Retrieval". *Journal of the American Society for Information Science*, 48:40–54.

Irving Lorge. 1959. *The Lorge Formula for Estimating Difficulty of Reading Materials*. Teachers College Press, Columbia University, New York.

Robert M. Losee. 1996. "Text Windows and Phrases Differing by Discipline, Location in Document, and Syntactic Structure". *Information Processing and Management*, 32 (6):747–767.

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.

T.C. Mendenhall. 1887. "The Characteristic Curves of Composition". *Science*, 9:237–249.

I. I. Menshikov. 1974. "K voprosu o zhanrovo-stilevoy obuslovlennosti sintaksicheskoy struktury frazy ("On genre-dependent stylistic variation of the syntactic structure in the clause")". In Golovin et al., editor, *Voprosy statisticheskoy stilistiki*. Naukova dumka; Akademia Nauk Ukrainskoy SSR, Kiev, Ukraine.

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.

Seppo Mustonen. 1965. "Multiple Discriminant Analysis in Linguistic Problems". *Statistical Methods in Linguistics*, 4:37–44.

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.

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.

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

Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A comprehensive grammar of the English language.* Longman, London, England.

J. Ross Quinlan. 1993. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, 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 Douglass R. Cutting. 1996. "Ranking for Usability: Enhanced Retrieval for Short Queries". Technical Report Apple Technical Report number 163, Apple Computer Inc., Cupertino, California.

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.

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

Rolf Sandell. 1977. *Linguistic Style and Persuasion, European Monographs in Social Psychology 11.* Academic Press, London, England.

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

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.

A. Ya. Shaykevich. 1968. "Opit statisticheskogo vydeleniya funktsionalnykh stiley (The practice of statistical discrimination of functional styles)". *Voprosy yazykoznaniya*, 7.

Robert Sigley. 1997. "Text Categories and Where You Can Stick Them: A Crude Formality Index". *International Journal of Corpus Linguistics*, 2 (2):199–237.

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.

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

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.

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.

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.

SPSS Inc., Chicago, Illinois. 1990. *The SPSS Reference Guide.*

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

Tomek Strzalkowski. 1994b. "Robust Text Processing in Automated Information Retrieval". In *Proceedings of the 4th Conference on Applied Natural Language Processing*, Stuttgart, Germany, October. ACL.

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

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.

Tomek Strzalkowski, Jussi Karlgren, Jose Perez-Carballo, Pasi Tapanainen, and Ngn Till. 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 and Jin Wang. 1996. "A Self-Learning Universal Concept Spotter". In *Proceedings of the 16th International Conference on Computational Linguistics*, København, Denmark, August. ICCL.

Jr. Edward H. Sussenguth. 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.

M. M. Tatsuoka. 1971. *Multivariate Analysis.* John Wiley & Sons, New York.

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.

Edward R. Tufte. 1983. *The Visual Display of Quantitative Information.* Graphics Press, Cheshire, Connecticut.

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, Narendra K. Gupta, and Ben Johnson-Laird. 1994. "The Collection Fusion Problem". In Donna Harman, editor, *Proceedings of the 3d Text Retrieval Conference*, National Institute of Standards and Technology, Gaithersburg, Maryland, November.

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.

Atro Voutilainen and Pasi Tapanainen. 1993. "Ambiguity resolution in a reductionistic parser". In *Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics*, Utrecht University, Utrecht, Holland, April. ACL.

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.

Swedish Institute of Computer Science
SICS Dissertation Series

01: Bogumil Hausman. *Pruning and Speculative Work in OR-Parallel PROLOG.*

02: Mats Carlsson. *Design and Implementation of an OR Parallel Prolog Engine.*

03: Nabiel A. Elshiewy. *Robust Coordinated Reactive Computing in SANDRA.*

04: Dan Sahlin. *An Automatic Partial Evaluator for Full Prolog.*

05: Hans A. Hansson. *Time and Probability in Formal Design of Distributed Systems.*

06: Peter Sjödin. *From LOTOS Specifications to Distributed Implementations.*

07: Roland Karlsson. *A High Performance OR-parallel Prolog System.*

08: Erik Hagersten. *Toward Scalable Cache Only Memory Architectures.*

09: Lars-Henrik Eriksson. *Finitary Partial Inductive Definitions and General Logic.*

10: Mats Björkman. *Architectures for High Performance Communication.*

11: Stephen Pink. *Measurement, Implementation and Optimization of Internet Protocols.*

12: Martin Aronsson. *GCLA:*
The Design, Use, and Implementation of a Program Development System.

13: Christer Samuelsson. *Fast Natural-Language Parsing Using Explanation-Based Learning.*

14: Sverker Jansson. *AKL—A Multiparadigm Programming Language.*

15: Fredrik Orava. *On the Formal Analysis of Telecommunication Protocols.*

16: Torbjörn Keisu. *Tree Constraints.*

17: Olof Hagsand. *Computer and Communication Support*
for Interactive Distributed Applications.

18: Björn Carlson. *Compiling and Executing Finite Domain Constraints.*

19: Per Kreuger. *Computational Issues in Calculi of Partial Inductive Definitions.*

20: Annika Wærn. *Recognising Human Plans:*
Issues for Plan Recognition in Human – Computer Interaction.

21: Björn Gambäck. *Processing Swedish Sentences:*
A Unification-Based Grammar and Some Applications.

22: Klas Orsvärn. *Knowledge Modelling with Libraries of Task Decomposition Methods.*

23: Kristina Höök. *A Glass Box Approach to Adaptive Hypermedia.*

24: Bengt Ahlgren. *Improving Computer Communication Performance*
by Reducing Memory Bandwidth Consumption.

25: Johan Montelius. *Exploiting Fine-grain Parallelism in Concurrent Constraint Languages.*

27: Ashley Saulsbury. *Attacking Latency Bottlenecks in Distributed shared Memory Systems.*

28: Kristian Simsarian. *Toward Human Robot Collaboration.*

SICS Dissertation Series 26
ISSN 1101-1335
ISRN SICS-D–26–SE
ISBN 91-7265-058-3

Jussi Karlgren • Stylistic Experiments for Information Retrieval