

**Some Considerations for Implementing
the SMART Information Retrieval
System Under UNIX**

Edward A. Fox^{*}

83-560
September 1983

^{*}Department of Computer Science
Cornell University
Ithaca, New York 14853

now at:
Department of Computer Science
Virginia Tech
Blacksburg, Virginia 24061

This work was supported in part by the National Science Foundation, under grant IST-81-08696.

TABLE OF CONTENTS

1 Introduction	2
2 History of Implementations	3
2.1 Early Versions	3
2.2 UNIX Implementation	4
3 Design	5
3.1 IBM SMART System	5
3.2 Use of UNIX	6
3.3 Design Principles	8
3.4 Retrieval Capabilities	9
3.5 Data Storage Schemes	12
3.6 INGRES Relations	4
3.7 Program for Search and Ranking	20
3.8 Retrieval Related Programs	2
3.9 Indexing Package	24
4 Creation and Clustering of Extended Vectors	27
4.1 Background	27
4.2 Constructing Bibliographic Submatrices	29
4.2.1 Input Data	29
4.2.2 Initial subvectors	0

4.2.3 Updating Subvectors	2
4.2.4 Construction Algorithm Complexity	35
4.3 Implementation in SMART	5
4.4 Clustering and Searching	41
4.4.1 Background	2
4.4.2 Clustering with Bibliographic Data	43
4.4.3 Algorithms	3
4.4.4 Clustering	44
4.4.4.1 Parameters	4
4.4.4.2 Actual Procedures in Outline Form	47
4.4.4.3 Linearized Cluster Tree	2
4.4.5 Searching	53
5 EVALUATION	5
5.1 Background	55
5.1.1 Recall/Precision	5
5.1.2 Summaries Available	56
5.1.3 Key Issues	7
5.2 Methods Utilized	58
5.2.1 Similarity Based Ranking	8
5.2.2 Query Statistics	59
5.2.3 User Oriented Averages	1
5.2.4 Single Value Measure	61

5.2.5 Statistical Comparison	1
5.2.6 Sample Evaluation Output	62
5.3 S Interface	0
5.3.1 S Package	70
5.3.2 Use of S	0
5.4 Discussion and Conclusion	73
6 Future Plans	5
6.1 Packaging	76
6.2 Cornell Projects	6
6.2.1 UNIX Retrieval Tools	76
6.2.2 Phrase and Thesaurus Construction	78
6.2.3 Probabilistic Retrieval	78
6.2.4 Office Information Systems	9
6.3 Extensions	79
6.4 Conclusion	2
References	83

SOME CONSIDERATIONS FOR IMPLEMENTING THE SMART INFORMATION RETRIEVAL SYSTEM UNDER UNIX

Edward A. Fox*

Abstract

Since the early 1960's the SMART project has tested out new ideas in information science aimed at fully automatic document retrieval. Beginning in 1980 development of an enhanced and generalized version of SMART has progressed at Cornell. The current implementation is in the C language and runs under the UNIX operating system on a VAX 11/780 computer.

The history of SMART is outlined. Considerations that led to the current design are described. Since SMART now allows multiple concept types to be manipulated in connection with an extended vector representation, storage and processing issues are discussed, including use of INGRES relations. Clustering algorithms are presented and run parameters are given for document clustering and subsequent clustered searching. SMART experiments (e.g. with p-norm queries, or probabilistic methods) can be compared using the evaluation package. The S statistical package can be applied to performing other special analysis and descriptive tasks. Finally, to illustrate the usefulness of these facilities, an outline is given of current SMART activities and of future plans.

*Department of Computer Science, Cornell University, Ithaca, NY 14853; now at Dept. of Computer Science, Virginia Tech, Blacksburg, VA 24061. This work was supported in part by the National Science Foundation, under grant IST-81-08696.

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.