RESEARCH-PYRAMID BASED SEARCH

TOOLS FOR ONLINE DIGITAL LIBRARIES

By

SULIEMAN BANI-AHMAD

Submitted in partial fulfillment of the requirements

For the degree of Doctor of Philosophy

Dissertation Advisor: Dr. Gultekin Ozsoyoglu

Department of Electrical Engineering and Computer Science

CASE WESTERN RESERVE UNIVERSITY

May, 2008
We hereby approve the dissertation of

SULIEMAN BANI-AHMAD

Candidate for the Ph.D. Degree *.

(signed) Gultekin Ozsoyoglu

(Chair of the committee)

Mehmet Koyuturk

G. Q. Zhang

Frank Merat

Meral Ozsoyoglu

(date) April 2, 2008

* We also certify that written approval has been obtained for any proprietary material contained therein.
Table of Contents

Table of Contents ........................................................................................................................................... i
List of Tables ................................................................................................................................................ iv
List of Figures ................................................................................................................................................ v
Acknowledgements ......................................................................................................................................... viii

Chapter 1. Introduction ................................................................................................................................... 1
  1.1 Problem Statement ................................................................................................................................. 1
  1.2 Contributions of the Thesis .................................................................................................................... 4
  1.3 Developed Tools ................................................................................................................................... 6
  1.4 Thesis Outline ......................................................................................................................................... 6

Chapter 2. Evaluating Publication Score and Similarity Functions in Digital Libraries ... 8
  2.1 Introduction ............................................................................................................................................ 8
  2.2 Evaluation of Score Functions ................................................................................................................ 12
    2.2.1 Paper Score Functions ...................................................................................................................... 12
    2.2.2 Author Score Functions ................................................................................................................... 16
    2.2.3 Publication Venue Score Functions ................................................................................................ 17
    2.2.4 Empirical Evaluation of Score Functions ......................................................................................... 18
      2.2.4.1 Experimental Setup .................................................................................................................... 18
      2.2.4.2 Paper Score Function Evaluation ............................................................................................... 23
      2.2.4.3 Author Score Function Evaluation ............................................................................................. 30
      2.2.4.4 Publication Venue Score Function Evaluation .............................................................................. 34
  2.3 Evaluating Publication Similarity Measures ......................................................................................... 38
    2.3.1 Introduction ...................................................................................................................................... 38
    Similarity Measures between Two Publications ......................................................................................... 40
      2.3.2 Text-Based similarities .................................................................................................................... 40
      2.3.3 Citation-Based similarities .............................................................................................................. 41
        2.3.3.1 Basic Bibliographic Coupling ................................................................................................... 41
        2.3.3.2 Bibliographic Coupling with Reachability Analysis .................................................................... 42
        2.3.3.3 Co-Citation Similarity with Reachability Analysis ................................................................. 45
Chapter 3. Improved Publication Scores via Research Pyramids

3.1 Introduction

3.2 Publication Scores

3.3 Properties of Research Pyramid Model

3.4 Research Pyramid Identification Procedures

3.5 Empirical Evaluations

3.6 Conclusions

Chapter 4. Scalable Content-Driven Search Keyword Suggester

4.1 Introduction

4.2 Problem Statement and Design Principles of the CDSK-Suggester

4.3 Constructing Token Hierarchy

4.4 Topic-Sensitive Token Weight

4.5 Suggesting Search Keywords

4.6 Experimental Results

4.7 Conclusions

Chapter 5. On Popularity Quality: Growth and Decay Phases of Publication Popularities

5.1 Introduction

5.2. Page Quality and Webpage Popularity Evolution Model

5.3 Properties of Publication Citation Graphs and Research Pyramids

5.4. The User Citation Behavior Model

5.5 Publication Quality with Aging Factor

5.6 Experimental Results and Observations

5.6.1 Popularity Growth Curves and Estimating Publication Quality
List of Tables

Table 2.1 Summary of paper, author and publication venue importance measures .................. 18
Table 2.2 Correlation between paper scoring measures .......................................................... 28
Table 2.3 Average overlapping ratio between author scoring measures ............................... 31
Table 2.4 Correlation coefficient between author scoring measures .................................. 32
Table 2.5 Top K overlapping ratios between venue importance measures for different values of K ........................................................................................................................................ 35
Table 2.6 List of top-scored publication venues .................................................................. 37
Table 2.7 Top 50 overlapping ratio between paper similarity measures ............................... 58
Table 3.1 The Means, InterQuartile Ranges (IQR), Skewness, and Kurtosis values of the Publication Score Functions. ........................................................... 79
Table 4.1 The observed linkage types and their percentages ................................................. 95
Table 4.2 The frequency of each observed parts-of-speech .................................................. 96
Table 4.3 The distribution of un-tagged tokens ................................................................. 107
Table 4.4 The number of links generated by two techniques, (i) Text-based adjacency used in the CompleteSearch engine [HI06] and (ii) linkage-based proximity used in our approach. .......... 109
List of Figures

Figure 2.1 Internal citation count Distribution ................................................................. 19
Figure 2.2 Paper importance distribution of citation-based functions ............................... 23
Figure 2.3 Score distribution of PageRank variations ...................................................... 26
Figure 2.4 Scatter plots display the correlation between selected paper importance measures ... 28
Figure 2.5 A scatter between Center of mass and venue importance($V_{\text{Citation Count}}$) .... 29
Figure 2.6 PageRank-based author’s importance distribution ........................................ 30
Figure 2.7 Scatter plots of the correlation between selected author importance measures .... 33
Figure 2.8 Publication venues score distribution .............................................................. 34
Figure 2.9 A scatter between Center of mass and venue importance($V_{\text{Citation Count}}$) .... 36
Figure 2.10 Illustration of citation networks ................................................................. 43
Figure 2.11 Two examples illustrating the use of second-level citation information .......... 44
Figure 2.12 An illustration of third-level citations ....................................................... 44
Figure 2.13 Three-level co-citation similarity illustrations ............................................. 45
Figure 2.14 Similarity score distribution of citation-based and author-coupling based paper similarity measures ................................................................. 55
Figure 2.15 The distribution of similar paper count based on bibliographical coupling ....... 59
Figure 3.1 Skewness of Score distribution of the three main citation-based publication score functions ........................................................................................................... 64
Figure 3.2 Citation age distribution curve of AnthP ....................................................... 66
Figure 3.3 The RP-Based Model ................................................................................. 66
Figure 3.4 SimRank score change with citation age .................................................... 68
Figure 3.5 SimRank score change with citation distance ............................................. 68
Figure 3.6 Functions of LB- and PB-IdentifyRP algorithms ........................................ 73
Figure 3.7 Examples where $PB-FormRP()$ is more successful than $LB-FormRP()$ ....... 75
Figure 3.8 Variance of citation-graph densities in different topics ................................. 77
Figure 3.9 Score distributions of PageRank normalized within RPs ................................ 77
Figure 3.10 Observed RP sizes by LP-IdentifyRP and PB-IdentifyRP ............................ 78
Figure 3.11 Score distributions of CitCnt normalized within RPs ..................................... 78
Figure 3.12 Distribution of no. of RPs annotated with each author ............................................. 80
Figure 3.13 Quality values distribution of the search results........................................................ 80
Figure 3.14 Sample results of the “complexity of join” query. Quality is computed using RP-
based (sample 1) and the globally-normalized PageRank (sample 2) along with the average
relevancy scores as assigned by experts. ...................................................................................... 81
Figure 4.1 Searching CiteSeer for keywords “query processing using query graphs”. ......... 85
Figure 4.2 Google suggest refinements search-keywords for the search keywords “graph th”... 86
Figure 4.3 Link-grammar parser output.......................................................................................... 96
Figure 4.4 levels of linkages and super-linkages......................................................................... 97
Figure 4.5 Layers of the Token Hierarchy................................................................................... 98
Figure 4.6 TextRank Input Graph [RP04], vertices are tokens and links represents linguistic
adjacency.......................................................................................................................................... 99
Figure 4.7 The SK-Suggester interface procedure................................................................. 104
Figure 4.8 Search-Keyword Suggester Query Execution Modules............................................. 105
Figure 4.9 Possible hits of the query “query graph”................................................................. 108
Figure 4.10 Distribution of RP coverage .................................................................................... 110
Figure 4.11 Distribution of token scope (after filtering).............................................................. 110
Figure 4.12 Distribution of specificity values of (a) all tokens (left) and (b) all tokens except
stopwords...................................................................................................................................... 112
Figure 4.13 Distribution of TextRank scores for simple tokens.................................................. 113
Figure 5.1 Searching Google Scholar for “top-k query processing for semistructured data” .... 117
Figure 5.2 Citation count per year for two publications that appeared in 1992 and 1994 (from
CiteSeer) and cited around 300 times each.................................................................................... 118
Figure 5.3 Popularity drop of webpages, as opposed to observed in-citation life cycle of
publications...................................................................................................................................... 118
Figure 5.4 In-citation per research pyramid (inter-research pyramid citations, i.e. citations from
publications outside an RP to ones inside it)............................................................................... 125
Figure 5.5 Inter-pyramid Citation count (x-axis) vs the (average difference in publication dates of
publications in a research pyramid)............................................................................................. 125
Figure 5.6 Empirical citation-age probability distribution curves (i.e., citation age vs frequency of citations with that age) of publications in three datasets (i) Data management (2) life sciences and (iii) computer science. ................................................................................................................................. 128
Figure 5.7 (a) Age vs frequency of citations of publications. (b) Weibull distribution CDF ..... 129
Figure 5.8 Relative error in quality estimation for different choices of the parameter c ........... 132
Figure 5.9 Popularity growth curves for sample publications ................................................. 133
Figure 5.10(a) Pt Popularity growth curve for a typical publication, (b) Citation age curve, the Weibull distribution, (c) the derivative of popularity growth curve $dP_t/dt$ (d) Popularity multiplied by citation age distribution $P_t$. .............................................................................................................................. 134
Figure 6.1 elGiza keyword-based search pipeline ..................................................................... 138
Figure 6.2 General Sequence of Ajax Request ......................................................................... 139
Figure 6.3 Sample search output format .................................................................................. 140
Figure 6.4 elGiza Keyword-based Search Page......................................................................... 141
Figure 6.5 Sample Output for the Search Keyword Suggester .................................................. 142
Figure 6.6 Search Settings Form - Keyword-based Search Tool.............................................. 145
Figure 6.7 Sample (ACM-Like) Search Output for Keyword-based Search Tool (query terms are “query graphs”)........................................................................................................................................... 148
Figure 6.8 Sample (ACM-Like) Search Output for Example-based Search Tool................. 149
Acknowledgements

First of all, I would like to express my sincere gratitude to my advisor Prof. Gultekin Ozsoyoglu for his guidance, support, and encouragement throughout my Ph.D. research. His expertise and excellent advice substantially improved my research skills and prepared me for my future careers. I was very fortunate to work under his supervision.

I want to thank Prof.s Mehmet Koyturk, G. Q. Zhang, Meral Ozsoyoglu, and Frank Merat for serving on my dissertation committee and providing helpful suggestions and comments for my research.

I would like to thank Ali Cakmak, Mustafa Kirac, and Nattakarn Ratprasartporn for their support and helpful comments with my research.

Finally, I would like to thank my parents, sisters, brothers, and my wife for their constant love, support, and encouragement throughout my graduate career.
Research-Pyramid Based Search
Tools for Online Digital Libraries

Abstract

by

SULIEMAN BANI-AHMAD

In any online literature digital library, findability precedes usability: users cannot use what they cannot find. Four research directions that support better findability in digital libraries are

(a) Accurate scoring functions to assign importance/prestige scores to publications,
(b) Accurate similarity measures for publications to locate publications similar to a given publication
(c) Accurate ranking measures to order search results based on their importance and relevance to users’ interests, and
(d) Helping users develop search keywords that lead to successful searches.

The contributions of this thesis are as follows.

1. Propose and comparatively evaluate score functions for publications, authors, and publication venues, as well as similarity measures for publications, towards research direction items a and b.
2. Validate a new model for the evolution of research and citation behavior, namely, the Research Pyramid Model. Then, propose and evaluate two algorithms for identifying research pyramid structures in publication citation graphs, and for research-pyramid-based publication score generation, towards research direction item a.
3. Propose and evaluate a citation-based publication popularity growth and decay model, towards research direction item c.
4. Using the Research Pyramid Model and the identified research pyramid structures, develop two literature digital library searching and ranking tools:
   a. A Research-Pyramid-based ranking tool that assigns accurate scores for publications, and
   b. A scalable content-driven Search-Keyword Suggester that helps users to put together query search terms effectively.
Chapter 1. Introduction

1.1 Problem Statement

An OLDL (Online Literature Digital Library) is a library in which collections, i.e., publications from one or more domains of study, are stored in digital formats (as opposed to print, microform, or other media) and accessible by users through the Internet. Examples of well-known OLDLs are IEEE Explore [IEX], ACM Portal [ACM], CiteSeer [CS], Google Scholar [GS], and PubMed [PMD].

Digital libraries are rapidly growing in popularity. For instance, ScienceDirect [SD], the world’s leading scientific, technical and medical information resource celebrated its billionth article download in November’06 since launched in 1999. Besides usage, digital libraries are also rapidly growing in terms of size and diversity of topics. For instance, (i) in Computer Science, ACM Digital Library [ACM] has close to 1 million full-text publications collected over 50 years, to search and download; (ii) in Electrical Engineering and Computer Science, IEEE Xplorer [IEX], another OLDL, provides users with on-line access to more than 1,700 selected conferences proceedings.

These high growth rates introduced several challenges facing the information access capability of OLDLs. Next we list few challenges that motivated the research conducted in this thesis.

Challenge 1: Large Sizes and Topic Diversity of Search Output Results. Search outputs of OLDLs tend to suffer from the “topic diffusion” problem, where commonly, keyword-based searches produce a large number of publications over a large number of topics, where not all topics are of interest to the user. One way to solve this problem is to assign scores to search
results (i.e., publications). Assigning scores to publications helps OLDLs to present the most important relevant publications to the user first. Citation-based publication score measures (e.g., citation count) are commonly used for ranking publications. At the present time, OLDLs lack effective and accurate publication ranking.

**Challenge 2: Lack of Effective Scoring Functions for Publications.** At the present time, OLDLs lack effective and accurate publication rankings [RE07]. Providing accurate publication scores can help users in reducing the time spent in searching ODLs, and thus enhances the scalability of OLDL usage as users can quickly identify important relevant publications to their topic of interest.

The above two challenges have motivated the work presented in chapters 2 and 3 of this thesis. This part of our work aims at (i) evaluating citation-based publication scoring functions in terms of effectiveness and accuracy, (ii) enhancing the accuracy of publication scoring functions to assign importance/prestige scores to publications.

**Challenge 3: Lack of Effective Scoring Functions for Search Outputs.** In the field of literature digital libraries, citation analysis is employed to order digital library search outputs (e.g., Google Scholar). Examples of citation-based measures are citation-count [SEC07] and PageRank [BP98]. However, as noticed by Cho et. al. [Cho05], citation-based measures compute popularity of publications based on the “current” state of a citation graph that continuously changes and evolves. Thus PageRank is effective in capturing the popularity of publications based on the current citation-graph in-hand. In chapter 5, we show that PageRank may assign inaccurate popularity scores for both old and recent publications. And thus PageRank can not be used to rank OLDL search outputs. We therefore need effective techniques to order search...
results based on their importance and relevance to users’ interests. This idea has motivated our research work presented in chapter 5.

**Challenge 4: Supporting Example-Based Search Queries.** In another direction, we have worked on enhancing OLDL user search experience through example-based search. Example-based search refers to finding similar publications to a publication identified by the user as a “good” example, and relevant to his/her search. For that purpose, we have worked on extending and evaluating the existing publication-to-publication similarity measures. Example-based search helps reduce the topic diffusion problem by providing the user with similar publications to the publications that the user finds relevant to his/her topic.

**Challenge 5: Scalable Search-Keyword Suggestion to Users.** Studies show that users spend considerable amounts of time in search sessions to properly select keywords, and to modify their search keywords in order to successfully locate documents that they are searching for. A search-keyword suggester may help users choose keywords properly, and thus, users are less likely to face unsuccessful search attempts. This has motivated the research presented in chapter 4 of this thesis which aims at developing a tool for helping users choose search keywords that lead to successful searches.

To summarize, we list the four research directions taken in this thesis to support better information access in digital libraries.

- Accurate and effective scoring functions to assign importance/prestige scores to publications,
- Accurate similarity measures for publications to locate publications similar to a given publication
• Accurate ranking measures to order search results based on their importance and relevance to users’ interests

• Helping users develop search keywords that lead to successful searches.

1.2 Contributions of the Thesis

The contributions of this thesis are as follows.

(A) Propose and comparatively evaluate score functions for publications, authors, and publication venues, as well as similarity measures for publications.

Using social networks or bibliometrics, a number of publication score functions has been defined [BP98, KL98, SIC05]. Examples are PageRank [BP98] and Authorities scores [KL98], both adopted from the world-wide-web search domain, and citation-count scores from the bibliometrics domain [CS03]. Importances of authors can be computed based on importances of publications that authors wrote. Similarly, importances of publication venues can be computed through the importances of publications that appeared in publication venues.

Existing publication similarity measures fall into two classes: (i) text-based similarity measures from the field of Information Retrieval (IR), such as the cosine similarity and the TF-IDF (term frequency-inverse domain frequency) model [GS98], or (ii) citation-based similarity measures based on bibliographic coupling (i.e., common citations between two publications) [RW98], cocitation (i.e., common citers of two publications) [SM73] or author-coupling (i.e., common authors between two publications).

Contribution 1: (Chapter 2 of this thesis and the reference [SIC05]), we compare and evaluate several publication score functions, including PageRank [BP98], Authorities scores [KL98], and citation-count scores. We also propose and evaluate multiple scoring functions for authors and
publication venues. All score functions are evaluated in terms of accuracy, separability, and independence.

**Contribution 2:** (Chapter 2 of this thesis and the reference [SDE05]). We summarize the existing publication similarity measures, and extend and evaluate them in terms of their accuracy, separability, and independence.

For the evaluations of both publication score and similarity functions, we use the ACM SIGMOD Anthology [ANTH] a digital library of about 15,000 publications in data management.

**(B) Validate a new model for the evolution of research and citation behavior, namely, the Research Pyramid Model. Propose and evaluate two algorithms for identifying research pyramid structures in publication citation graphs, and for research-pyramid-based publication score generation.**

Studies show that research papers in any scientific discipline, such as data management, naturally cluster into ‘research pyramids’, where a research pyramid refers to a small set of publications that deal with the same most-specific research topic [AYA].

**Contribution 3:** (Chapter 3 of this thesis and the reference [SEC07]), we validate the research-pyramid model, and propose and empirically evaluate two approaches to identify research pyramids.

We use the research-pyramid model to assign accurate scores for publications as presented in chapter 3 of this thesis.

**(C) Propose and evaluate a citation-based publication popularity growth and decay model.**

**Contribution 4:** (Chapter 6 of this thesis and the reference [SGECDL08]). For publication citation graphs, we (i) experimentally validate the popularity growth phase model proposed by Cho et. al. [Cho05]. (ii) propose a probabilistic model for domain-specific publication citation behavior, and (iii) propose a new definition and a mathematical model for popularity growth and
decay for publications by coupling Cho et. al.’s model of popularity growth [Cho05] with our probabilistic publication citation behavior model, which we refer to as the publication quality with aging factor. This new definition helps to accurately rank of OLDL search output results, and reduce the inherent bias of citation-based publication score functions.

1.3 Developed Tools

Using the Research Pyramid Model and the identified research pyramid structures, we develop two OLDL searching and ranking tools:

(A) A Research-Pyramid-based Ranking Tool for Publications

This tool basically assigns score publications within their research pyramids, which results in publication scores that are (i) accurate and (ii) separable, i.e., distribute well over a given interval, e.g., [0, 1].

(B) A scalable content-driven Search-Keyword Suggester that helps users to put together query search terms effectively.

Contribution 5 (Chapter 4 of this thesis and references [SGJCDL08, SGSKSTR]). We propose and evaluate a “content-driven search keyword suggester” for keyword-based search in literature digital libraries. Our search keyword suggestion approach is based on an a priori analysis of the publication collection in the digital library at hand, and consists of the following steps.

1.4 Thesis Outline

The rest of this thesis is organized as follows. Chapter 2 consists of two parts; in the first part we present our observations on the drawbacks of the current citation-based scoring functions that are used in ranking publications, authors and publication venues. In the second part we present, extend and evaluate several publication-to-publication similarity score functions that can be used to answer user queries of type query-by-example. In chapter 3, we validate the research pyramid
model and use it to solve the separability and accuracy problems of publication score functions. In chapter 4, we propose and evaluate a “content-driven search keyword suggester” for keyword-based search in literature digital libraries. In chapter 5, we (i) experimentally validate that PageRank score of publications change over time and follow the logistic growth equation [Cho05], (ii) empirically model one aspect of researchers citation behavior in technology-driven fields of study such as computer science, this model capture researcher tendency to cite old publications, and (iii) extend the popularity growth model developed by Cho et. al. [Cho05] to capture publication popularity decay.
2.1 Introduction

Web-based (literature) digital libraries are now commonplace, providing access to publications of each discipline. Examples are ACM Digital Library [ACM] and IEEE Xplore [IEX] in computer science, PubMed [PMD] (largest with 14 million publications) in biomedical sciences, CiteSeer [CS] with 600K publications in computer science, and the recently launched Google Scholar [GS].

This chapter has two parts. In the first part, we deal with the issues of defining score functions for publications, authors, and publication venues in digital libraries, and evaluating how good they are. Presently, digital libraries do not assign scores to publications, even though they are potentially useful for (a) providing comparative assessment, or "importance", of papers, (b) ranking papers returned in search outputs, and (c) using scores in locating similar papers. Using social networks or bibliometrics, one can define a large number of publication score functions.

To assess the "goodness" of such a score function, next we list three measures.

- **Accuracy**. By measuring recall at rank k, and precision at rank k [CS03], we can decide how well a score function assigns scores. However, this approach has its disadvantages: (a) it requires expert identification of scores as "correct" scores, (b) it is manual; (c) experts may vary in their judgments; and estimates of expert judgments may not be consistent [CS03]. We propose the following two approaches which do not rely on the existence of an expert.
• We assume that multiple score functions indicating that a paper is an important one is a strong indication that the paper is important. Based on this assumption, we can say that score functions that divert from or do not agree with the majority can be identified and excluded. Applying this idea via "top-k papers", if a number of score functions agree in their top-k score rankings for sets of papers, where k is properly chosen, they are perhaps consistent with each other and "accurate".

• We employ a consistency-based correlation experiment to check the accuracy of paper scoring and similarity functions as follows: Publication venues and the set of papers published in them exhibit a mutually reinforcing relationship; in general, papers from the most important publication venues (like ACM SIGMOD Conf., VLDB Conf., and ACM TODS journal in AnthP) are expected to be of higher quality than papers published in less important venues. Based on this assumption, we assess paper scoring functions by checking the consistency between the publication venue score and the scores of the papers published in it. For paper similarities we check the consistency between the proposed citation-based similarity functions and the text-based similarities that are studied heavily in the literature on the other hand.

• Separability. To provide a comparative assessment of papers, authors, and publication venues, it is desirable to have a score distribution which is not highly skewed. Citation count distributions (as paper score functions) in general decay very fast [RS04], and, most citation-based score functions do not separate scores well over a given range. One approach we propose and evaluate in this work is to use nonlinear normalizations to correct this problem.
• **Independence.** We would like to employ score functions that are noncorrelated with each other.

Next we briefly summarize the currently-used or proposed score and paper similarity functions. For assigning (importance) scores to publication venues, a number of studies in biomedical sciences assign scores ("journal quality indicators") to biomedical journals [KL02]. We are not aware of any studies that compute scores for authors. Existing citation-based publication score functions are all based on the notion of prestige in social networks [WF94] and bibliometry [CS03]. The well-known PageRank [BP98] algorithm determines the importance of a web page (in our case, paper) by the number and importances of pages with links to it (in our case, citing papers). The Hyperlink Induced Topic Search (HITS) algorithm [Kl98] is similar to the PageRank algorithm in that HITS involves computing two scores for each page; hub and authority scores. Authorities represent high-prestige pages, whereas hubs are pages that have links to authorities. Other citation-based score functions can be derived as follows. (a) Use normalized citation count (i.e., how many times a paper is cited by other papers) as the basis for a score function. (b) Revise the score of a paper using the score of its publication venue (conference or journal). (c) Add weights to citations, e.g., citations by an "important" author's work are more significant. (d) Revise the score of a paper using temporal distributions of citations; e.g., citations in the last 10 years are more significant than earlier citations. (e) Revise a paper score using the score of its citation venue; that is, capture the notion of a hub or an authority, e.g., survey journal represents a hub, whereas a research paper represents an authority. (f) Revise a paper score by the score of its author. One can also combine the score functions above.
Text-based similarity functions are based on information retrieval methodologies [GS89,C98]. As an example, using the vector space model of IR and the TF-IDF weighting scheme [GS89], similarity between two papers may be measured by using cosine, jaccard, dice or other document measures.

CiteSeer [CS] is a literature search system for searching (presently) about 600,000 computer science and bioinformatics papers. CiteSeer uses three document measures, namely, word vectors, LikeIt string distance [LI97], and the Common Citation, Inverse Document Frequency [Giles98]; it also uses the "citation count" paper score function to rank its output. Google Scholar is a specialized version of the popular Google search engine to search "scholarly literature, including peer-reviewed papers, theses, books, preprints, abstracts and technical reports from all broad areas of research" [GS]. While Google Scholar provides similarity scores, the similarity function it uses is not known.

Processing ranked queries is studied in several data retrieval applications. Basically, the data objects are ranked using some ordering criteria, and top-k ranked objects are returned. [CK97, CK98, CG99, F01].

Our major findings in this work are as follows.

- Citation-count-based scoring is the best in terms of separability. PageRank-based scoring is the best in terms of accuracy.
- Authorities scores of HITS and PageRank scores are highly correlated.
- Separability and accuracy of PageRank-based paper scores can be enhanced by (a) weighing citations (b) weighing the "Future Citation Probabilities" represented by the $E$ parameter of PageRank (c) postprocessing PageRank raw scores by (i) nonlinear
normalization, (ii) linear normalization by a properly selected percentile score or (iii) combining PageRank-based paper scores and publication venue scores.

- Author scores based on author's top K-scored or top-K% scored papers accurately capture author scores.
- Citation-count-based publication venue scores are more accurate than author-score-averaging publication venue scores published in publication venues.

By evaluating "multiple levels" of paper similarities based on bibliographic-coupling, cocitation and author-coupling, we observe that: (a) similarity value distribution curves are similar within the same group of similarity functions, (b) citation-based and author-coupling based similarity functions are more separable than bibliographic-coupling-based functions, (c) top-K overlapping ratio between paper similarity functions increases as we move to "higher levels" of similarity functions as more papers appear to be similar. (d) Text-based similarity function show very low overlapping with citation-based and author-coupling-based functions.

2.2 Evaluation of Score Functions

In this section we present, extend, and evaluate citation-based score functions for Papers, Authors and Publication Venues.

2.2.1 Paper Score Functions

A. PageRank. Importances of papers that cite a particular paper determines its importance. PageRank [BP98] and HITS [KI98] were designed based on this assumption. PageRank scores is computed recursively using the formula

\[ P_{i+1} = (1 - d) M^T P_i + E \]
Where $P_i$ and $P_{i+1}$ are the current and next iteration PageRank vectors respectively. $M$ is a matrix derived from the citation matrix $C$ by normalizing all row-sums in $C$ to 1. $C$, in turn, is the adjacency matrix of the graph $G$ formed as follows; the papers represent the graph nodes, and the citation relationships between these papers represent the edges. $C$ is of size $N \times N$, where $N$ is the total number of papers in the system. Finally, $d$ and $(1-d)$ are the future citation probability.

Given that an author $A$ who is writing a new paper and already cited paper $u$ which in turn cites paper $v$, and let $w$ be a paper in AnthP selected randomly. The parameter $d$ represents the probability that $A$ will cite $w$, and $(1-d)$ is the probability that $A$ will cite $v$.

To guarantee the algorithm convergence, it is assumed to have a hidden link between each pair of the graph nodes. This link is represented by the user-defined parameter $E$. A variation of $E$ is simply $E_1=d$. Another variation of $E$ that is used in [PB98] is

$$E_2 = d / N \left[ \frac{1}{N} \right] P_i.$$ 

Where $1_N$ is a vector of $N$ ones. Next we list changes to PageRank.

- We use the following score mapping schemes to normalize raw PageRank scores to a given interval:
  - Linear normalization maps PageRank scores to the interval $[0, 1]$ by dividing each score by the selected raw PageRank score $PR_{\text{norm}}$ where $PR_{\text{norm}}$ is selected in two different ways: (i) absolute maximum PageRank score: the disadvantage of this choice is that the outliers will negatively affect the score distribution, (ii) $n^{\text{th}}$ percentile PageRank score: this choice reduces the effect of outliers by reducing the threshold by which we linearly normalize. All papers with score higher than $PR_{\text{norm}}$ receive the score of 1. This means that we
sacrifice separability of these papers, which are very few as it turns out, to gain separability for the rest of the paper set. We normalize by the 99th and 95th percentiles in our experiments.

- Nonlinear normalization reduces the effect of outliers by resisting the increase in the PageRank scores of papers as they receive more and more citations. Google uses this approach to map PageRank scores to the interval \([0..10]\) [Rid]. In experiments, we use the logarithm function for this purpose.

• Enhancing PageRank by preprocessing its inputs: We list the following variations for the citation matrix and the parameter \(E\):
  
  - Weighted citations (versus Uniform citation weight): (i) use the number of times a paper \(x\) is cited within the body of paper \(y\) as a weight for the citation from \(y\) to \(x\), and (ii) use publication venue ranking of a paper as a weight for the citation. We employ two alternatives for venue ranking: (1) Expert ranking, and (2) Citation count of venues as venue ranking.
  
  - Weighted future-citation probability (WFCP). To weigh future-citation probability, we use the score of the publication venue where the paper is published as a representative of the future-citation probability. The motivation is that researchers tend to cite papers published in premier venues more than others.

Note that 46% of the AnthP papers receive no citations from AnthP papers. Thus, if no corrective actions are taken, all of these papers receive the minimum PageRank score. To enhance the separability of paper scores, we compute the final (importance) score of a paper \(P\) as

\[
P_{\text{Score}}(P) = W_{\text{PageRank}} \times P_{\text{ScorePageRank}}(P) + W_{\text{venue}} \times V_{\text{ScoreCitationCount}}(v)
\]
where \( v \) is the publication venue of \( P \), \( P_{\text{scorePageRank}}(P) \) is the PageRank score of \( P \), \( V_{\text{scoreCitationCount}}(v) \) is the citation count score of venue \( v \), and the sum of the two weights is one, i.e., \( W_{\text{PageRank}} + W_{\text{venue}} = 1 \). In experiments, we use 0.7 for the PageRank score weight and 0.3 for the venue citation count score weight.

**B. Hubs and Authorities.** Authority score of paper \( P \) is computed by summing up the hub scores of the papers citing \( P \). Hub score of \( P \) is computed by summing up the authority scores of the papers that \( P \) cites. Computation is recursive until results converge after a number of iterations. One difference between HITS and PageRank is that the first one works on papers in the result set of a query, while the latter considers all the papers independent of the query [C03]. We use authorities as paper scores.

**C. Citation Count.** A paper, normally, does not cite another paper unless the cited paper is relevant. And, large number of citations to a paper gives an indication that the paper is important. Based on this fact, we can use citation count as a measure for paper importance. For a given paper \( P \), let \( \text{CitationCount}(P) \) be the number of times paper \( P \) is cited by other papers. Using the number of citations, paper \( P \) is as important as those papers that have the same number of citations and more important than those papers that have fewer citations. let \( \text{PapersWithCitations}(i) \) be the number of papers that are cited \( i \) times. The importance of a given paper \( P \) with respect to its citation count can be computed as follows[AA03]:

\[
P_{\text{Citation Count}}(p) = \sqrt{\frac{\text{CitationCount}(p)}{\sum_{i=0}^{\text{CitationCount}(p)} \text{PapersWithCitations}(i) / \text{No. of Papers}}}
\]

We will refer to this chapter ranking measure as \( P_{\text{Citation Count}} \).
2.2.2 Author Score Functions

We compute author importances in four different ways:

1. [Top-K-Avg]Weighted average of the most important k papers of author A:

   \[
   \text{Top-K-Avg}(A,k) = \frac{\sum_{i=1}^{k} \text{Imp}(P_i)}{k}
   \]

   where \( P_i \) is the \( i \)th most important paper of author A.

2. [Top-K%-Avg]Weighted average of the top \( k\% \) papers among author A’s papers.

   \[
   \text{Top-K%-Avg}(A,k) = \frac{\left[\sum_{i=1}^{\lfloor k\% |S_A|\rfloor} \text{Imp}(P_i)\right]}{k\% |S_A|}
   \]

   where \( S_A \) is the set of papers written by author A, and \( P_i \) is the \( i \)th most important paper of author A.

3. [Top-K-Avg-TD]Weighted average of the \( k \) most-important papers of author A in every \( t \)-year durations,

   \[
   \text{Top-K-Avg-TD}(A,k) = \max \left\{ \frac{\sum_{i=1}^{k} \text{Imp}(\text{Sort}_{\text{Desc}}(P_{t_i}))}{k} : \text{where } P_{t_i} \text{ are papers of } A \text{ published in } t_i, t_i = [t_{i1}, t_{i2}] \text{ and } t_{i1} - t_{i2} = t \right\}
   \]

4. [Top-K-Avg-TV]Weighted average of the \( k \) most-important papers of author A published in the top \( t \) highest-scored publication venues.

   \[
   \text{Top-K-Avg-TV}(A,k) = \max \left\{ \frac{\sum_{i=1}^{k} \text{Imp}(\text{Sort}_{\text{Desc}}(P_{v}))}{k} : \text{where } P_{v} \text{ are papers of } A \text{ published in } v \in \{\text{top } t \text{ publication venues}\} \right\}
   \]
2.2.3 Publication Venue Score Functions

We propose the following measures to compute the importance score of a given conference, journal, workshop $V$:

1. $[\text{VCitation\_Count}]$ use the number of citations to the papers published in $V$ as an importance measure as follows:

$$V_{\text{Citation\_Count}}(V) = MinV + (1.0 - MinV) \times \left(\frac{CC[V]}{PC[V]} \right) / MaxV$$

where $CC[V]$ and $PC[V]$ are the number of citations to the papers published in $V$ and the number of papers published in $V$ respectively. $MinV$ is a minimum-value weight, and $MaxV$ is the highest publication venue importance score. We use $MinV = 0.4$ in the experiments.

2. $V_{\text{Auth\_Avg}}$ use the average of author’s importance scores for the authors who have published in $V$:

$$V_{\text{Auth\_Avg}}(V) = \frac{\sum_{A \in S} \text{ImpAuthor}(A)}{MaxV}$$

where $S$ is the set of authors published in $V$.

Table 2.1 summarizes the paper, author and publication venue importance measures.
Table 2.1 Summary of paper, author and publication venue importance measures

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P\textsubscript{PageRank}</td>
<td>Paper importance- PageRank</td>
</tr>
<tr>
<td>P\textsubscript{Authorities}</td>
<td>Paper importance – Authority score of HITS</td>
</tr>
<tr>
<td>P\textsubscript{Citation Count}</td>
<td>Paper importance - Citation count</td>
</tr>
<tr>
<td>Top-K-Avg</td>
<td>Author importance – Weighted average of the most important k papers of author A</td>
</tr>
<tr>
<td>Top-K%-Avg</td>
<td>Author importance – Weighted average of the most important k% of the author A’s papers</td>
</tr>
<tr>
<td>Top-K-Avg-TD</td>
<td>Author importance – Weighted average of the k most-important papers of author A in every t-year durations</td>
</tr>
<tr>
<td>Top-K-Avg-TV</td>
<td>Author importance – Weighted average of the k most-important papers of author A in the top t highest-scored publication venues</td>
</tr>
<tr>
<td>V\textsubscript{Citation Count}</td>
<td>Publication venue importance – the number of citations to the papers in the venue</td>
</tr>
<tr>
<td>V\textsubscript{Auth Avg}</td>
<td>Publication venue importance – the average importance scores of authors published in the venue</td>
</tr>
</tbody>
</table>

2.2.4 Empirical Evaluation of Score Functions

2.2.4.1 Experimental Setup

Sample set selection. For each paper in the ACM Anthology, DBLP bibliography [DBLP] is used to extract the titles, authors, publication venue (conference or journal), and publication year info. Information extracted about each paper is the paper’s publication venue, the publication year, authors, and citations. Our experimental dataset includes: (a) 106 conferences, journals, and books, (b) 14,891 papers, and (c) 13,208 authors. We refer to the citations from any paper in the AnthP set to a paper in the same set as Internal citations. Citations from any paper in AnthP to any paper P where outside AnthP but within DBLP papers are referred to as External citations. We refer to the citation from paper P to other papers as out-citation of P. And for the citation from a paper to paper P as in-citation to P. By default, the work “citation” covers in-citations.
The average number of citations of SIGMOD Anthology papers was 20. We excluded citations to any papers outside AnthP or DBLP. The average number of in-citations, internal and external, per Anthology paper is 4.289. The average internal in-citation count per Anthology paper is 2.066. This means the average citation reduction due to external citation removal was 48.2%.

Figure 2.1 Internal citation count Distribution

Figure 2.1 shows some statistics of ACM SIGMOD Anthology. Figure 2.1(a) displays the citation count distribution over years, Notice that the most recent papers are not cited yet, this means that their score will be very low although we do not know how important they are for sure. Same comments apply to the papers published before 1974; we do not have information as to which papers cite them. The papers published before 1974 and after 2000 were very few as shown in Figure 2.1(b).
Figure 2.1(c) displays the distribution of internal in-citation counts for the papers in AnthP. After excluding external citations, 6851 papers received no internal citations (46%). The number of papers that received one and two internal citations are 2141 (16%), and 1514 (11.2%) respectively. These figures indicate that the citation graph represented by the AnthP papers is highly sparse.

Figure 2.1(d) shows top 10 venues in term of their citation count. This matches the experts ranking of these venues. These venues are known to be the best in the computer science community.

In our experiments, we considered only the AnthP papers. We also excluded papers published after 2000 and before 1974 because their citation-based score does not reflect its quality as we explained before. After these selections, AnthP includes 13,493 papers.

We evaluate the importance measures in term of their independency, accuracy and separability.

**Comparing Score Functions.** For comparison purposes, we start with paper importance measures, because, as we mentioned before, they form the base for computing authors and venues importance values.

To evaluate the *separability* of the measures, we use histograms to explore the distribution of the scores over the interval [0, 1]. To check the measures *independency* we perform two experiments:

- *Top K overlapping*: we pick the Top scored K papers, authors and publication venues based on each importance measure. Having done that, we measure the percentage of overlapping papers, authors and publication venues between these sets. The measures
whose top-K highly overlap are of higher quality. Exclude other measures that deviate from all the others.

- **Correlation coefficient computation**: we compute the statistical correlation coefficient between different importance measures. Keeping the scores of two highly correlated measures in our database is redundant.

The motivation behind these experiments is that if multiple measures indicate a paper is an important one, then this gives a strong indication that the paper is perhaps important. Based on this assumption we can say that measures that divert from or do not agree with the majority of measures can be excluded.

The majority of the importance measures we propose are all citation based. We expect to have some degree of correlation among them. If two measures highly agree, that is, the correlation coefficient between them is high, then keeping both of them in our database is not important.

To check the **accuracy** of the importance measures we perform two experiments:

- Consistency-based correlation: Venues and the set of papers published in them exhibit what could be called a mutually reinforcing relationship; papers from most important publication venues (like SIGMOD, VLDB and TODS) are expected to be of higher quality than papers published in less important venues. Based on this fact we study the consistency between a particular venue importance and the importance of the papers published in it. To avoid cyclicity, we consider \( V_{\text{CITATION,COUNT}} \) as a measure for venue’s importance, because \( \text{AUTH_AVG} \) is based indirectly on paper’s importance. Based on this we may:
  
  - Group the AnthP papers by the publication venue where they were published.
Measure the “distance” among the groups. If -based on \( V_{\text{CITATION_COUNT}} \) venue \( u \) is more important than venue \( v \), then the distance between \( u \) and \( v \) should be positive.

We suggest using the *Center of mass (CoMass) difference* as distance function as follows:

- Compute the average paper importance of each group.
- Compare the CoMass values. Venues that are more important should receive a higher average.

If venues that are more important receive a higher CoMass value, then the importance measures really reflect the importance of venue. This also supports the paper impotence measures that we used.

- Human judgment of paper importances and venues ranking
  Finding a 100% reliable automated benchmark to evaluate the ranking measures of papers, authors and publication venues - without human judgment may be misleading. The quality of a querying system necessarily requires human evaluation, due to the subjectivity inherent in *importance* notion. To support the previous approaches of comparing, we can utilize human experience in this field as follows,

  - pick \( n \) papers that deals with the same research problem is some field in computer science,
  - give them to an expert person to rank them,
  - compare his with the ranking you get from different importance measures by computing the sum of paper’s displacement as follows, assume \( A \) and \( B \) are these
set of papers ranked by the expert and by scoring measures we use (PageRank, Authorities …) respectively:

\[
disp(A, B) = \left(\sum_{i=1}^{n} \text{abs}[i - P_{i_a}]\right) / \sum_{i=1}^{n} i
\]

Where \( P_{i_a} \) is the rank of the \( i^{th} \) paper of A in B.

For venues ranking, we can compare the ranking that we obtain based on the proposed venue’s importance measures and the experts ranking.

2.2.4.2 Paper Score Function Evaluation

Separability

![Figure 2.2 Paper importance distribution of citation-based functions](image-url)
The AnthP graph is highly sparse. This explains the fact that most of the papers received very low PageRank (with weighted citations – WC_CC) and Authority scores. This phenomenon is visualized in the graph of Figure 2.2.

The $P_{\text{citation\_count}}$ curve, as it is clear from the graph of Figure 2.2, spreads better than the other curves over the $[0,1]$ interval. The reason is that the score of paper $P$ is computed by the following equation:

$$P_{\text{Citation\_Count}}(p) = \sqrt{\left( \sum_{i=1}^{\text{CitationCount}(p)} \frac{\text{PapersWithCitations}(i)}{\text{No. of Papers}} \right)}$$

Assuming that paper $P$ receives $C$ citations, the term under the square root is the ratio of two numbers; $n$ and $N$; $n$ is the number of papers that receives $C$ citations or less. $N$ is the total number of papers in AnthP. This means that for the papers that are cited the most their score will be 1.0. The majority of PageRank and Authorities scores cluster around the 0.1 value. This is because we normalize by maximum raw PageRank score. In AnthP, 73.2% of the papers received two citations or less. Only two papers receive 34 citations. Thus, the majority of the papers received PageRank, and Authority scores that cluster around the 0.1 value.

**Observation 1:** $P_{\text{Citation\_Count}}$ showed better separability than PageRank and Authority that are normalized by the maximum raw score.

**Observation 2:** Normalizing by maximum score (outlier) negatively affects the distribution of the measure over a given interval.

To make PageRank more separable we propose the following mapping schemes:

- Linear mapping by the $n^{th}$ percentile
• Nonlinear mapping

The first enhances separability by avoiding normalizing by extreme outliers. And the second imitates Google’s approach of mapping the raw PageRank scores to the set [0..10]. In our experiments, we use 99\textsuperscript{th} and 95\textsuperscript{th} percentiles for normalization. For nonlinear normalization, we use the logarithm function.

To make PageRank more realistic, we propose the following variations that involves pre-processing the inputs, namely the citation matrix and user-defined parameter E:

• Pre-processing the citation matrix: since the number of times a paper x is cited within the body of paper y reflects stronger relationship, we use this number as a representative of the citation strength. Yet another possible citation weight is the importance of the venue of the citing paper; that is, citations from premier publication venues weights more than citation from others.

• Weighted future-citation probability WFCP. We use the importance of the venue where the paper is published as E. the motivation behind this, as we mentioned before, is the fact that researchers tends to cite papers published in premier venues more than others.

We have two alternatives to use venue’s importance as citation weight; expert’s ranking and the $V_{\text{Citation\_count}}$ score of the publication venue. We use $V_{\text{Citation\_count}}$ as WFCP.
Figure 2.3 displays score distribution of PageRank variations. Figure 2.3(a) shows the score distribution of the original form of PageRank; that is, with uniform citation weight and future citation probability. The figure indicates that taking the logarithm on the scores meets the nonlinearity scaling and distributes the scores better. From Figure 2.3(b) we notice that using the $V_{\text{Citation Count}}$ score as a measure of citation weight enhances the separability and distribution of the scores. The number of papers with minimum score was more than 14,000 in the original form of PageRank. The number of papers with minimum score was 9,500 in PageRank with weighted citation; this indicates that the separability is better when weighting the citations.
**Observation 3:** Nonlinear normalization using LG function and weighting citation by the citing paper venues importance enhances separability.

**Observation 4:** The effect of using expert's ranking on separability matches the effect of using citation weight and $V_{\text{Citation\_count}}$.

Figure 2.3(c) that we can avoid the effect of the outliers by reducing the threshold by which we normalize. The Figure indicates that there are 9,500 papers with very low scores. As we know, 46% of the AnthP papers receive no citations. This means that all these papers will receive the minimum score. To gain some separability for these papers, we involve the importance of the publication venue, where the paper is published, to compute the final score of the paper. Figure 2.3(d) displays the score distribution in this case.

**Observation 5:** Reducing the threshold by which we normalize enhances separability. Involving the publication venue importance score in computing paper importance enhances the separability even for papers with no citation (recent papers for instance).

**Independence**

We have chosen three different K values, namely, 10, 20, and 30, for the Top-K-overlapping experiments over AnthP. We have observed that $P_{\text{PageRank}}$, $P_{\text{Auth}}$, and $P_{\text{Citation\_Count}}$ completely agree in their top-K paper rankings.

**Observation:** $P_{\text{PageRank}}$, $P_{\text{Auth}}$, and $P_{\text{Citation\_Count}}$ perfectly overlap.

In a recent work, Diang el. al. gave a detailed analysis of HITS algorithms based on a combination of probabilistic analysis and matrix algebra [Diang03]. They showed that ranking by HITS is the same as ranking by counting inbound and outbound hyperlinks. In our case, "hyperlinks" indicate the citation relationship between papers. PageRank, on the other hand, is
also highly correlated with ranking by indegree (counting inbounds). This goes in parallel with what we have observed.

We used the complete set of AnthP papers to compute the correlation between different score functions. Table 2.2 shows the correlation between each pair of paper score functions. The high correlation coefficient values indicate that \( P_{\text{PageRank}} \) and \( P_{\text{Auth}} \) are highly correlated. This relationship came from the fact that both are based on the same idea of computing paper scores recursively until they converge. Figure 2.4(b) visualizes this observation.

**Observation:** (Table 2.2 and Figure 2.4): \( P_{\text{PageRank}}, P_{\text{Auth}} \) score functions are highly correlated and dependent.

<table>
<thead>
<tr>
<th></th>
<th>Correlation coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_{\text{PageRank}} - P_{\text{Auth}} )</td>
<td>0.992</td>
</tr>
<tr>
<td>( P_{\text{PageRank}} - P_{\text{Citation Count}} )</td>
<td>0.574</td>
</tr>
<tr>
<td>( P_{\text{Auth}} - P_{\text{Citation Count}} )</td>
<td>0.553</td>
</tr>
</tbody>
</table>

Table 2.2 Correlation between paper scoring measures

![Scatter plots](image)

**Figure 2.4** Scatter plots display the correlation between selected paper importance measures

**Accuracy**

We used the complete set of AnthP papers to check the consistency between paper importance and the importance of the venue where they published in. We grouped them according to their
venues. The relation between the center of mass of each group and the venue importance was not clear, correlation is 0.44. The scatter plot of Figure 2.5(A) gives clear picture of that.

![Figure 2.5 A scatter between Center of mass and venue importance(V_Citation_Count)](image)

As it is clearly appear from the scatter plot of Figure 2.5 (A), there is a positive correlation between CoMass and V_{CITATION_COUNT} score of venue importance. We expected to get a straight or at least almost straight line, in which case, a venue with importance \( r \) has a CoMass value that is greater than the CoMass values of all less important venues. What appears from the graph of Figure 2.5 this is the case. For instance, the venue with maximum V_{CITATION_COUNT} score has CoMass that is greater than the CoMass of most of the other venues.

We extracted the papers that have at least 75% of the papers they cite available in AnthP. The number was 2135 papers. We believe that this sample represents the real complete set. We re-conducted some of the same experiment over this paper set. The trend became clearer. Figure 2.5(B) shows the scatter plot of CoMass and the venues’ importance (V_{Citation_Count}). The correlation between CoMass and V_{Citation_Count} values was relatively high. The correlation coefficient was 0.71.
2.2.4.3 Author Score Function Evaluation

Separability

We base author’s Top-K-Avg importance on top-scored 20 papers written by him. We base author Top-K%-Avg importance on top scored 40% of all papers he published. We avoided using AUTH_AVG (publication venue score) because we have already used author’s importance scores in computing publication venue importance.

As it is clear from Figure 2.6, most of the scores were low; this is because we compute all of them based on PageRank paper’s importance values. Papers’ scores were low because all paper’s measures are citation-based ones, and the graph available for our experiments is sparse.

Notice that TOP-K-AVG scores are higher than TOP-K%-AVG at the beginning. TOP-K-AVG considers top 20 important papers while TOP-K%-AVG considers top 40% of an author’s papers. Notice that for authors with less than \( n_f = 20/0.4 \) papers, the size of the number of papers
involved in computing Top-K-Avg is larger than the number of papers involved in computing Top-K%-Avg. This means that those authors receive higher Top-K%-Avg scores. For authors with number of papers higher than \( n_t \), their Top-K-Avg score are higher than their Top-K%-Avg because fewer number of low-scored papers involved in calculating Top-K-Avg author score.

Also, from Figure 2.6, Top-K-Avg-TD has the highest scores. The reason for that we involve very few papers in computing the author’s importances. That is we include the best scored of his papers. The worst curve was that of Top-K-Avg-TV because it is not easy for majority of the authors to have papers accepted in top venues. The competition between authors to publish in these venues is very high.

**Observation 6:** Separability of author importance scores is low because the separability of PageRank scores is low.

**Independence**

<table>
<thead>
<tr>
<th>Scoring Measures</th>
<th>Average overlapping ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top-K-Avg - Top-K%-Avg</td>
<td>0.97</td>
</tr>
<tr>
<td>Top-K-Avg - Top-K-Avg-TD</td>
<td>0.0</td>
</tr>
<tr>
<td>Top-K-Avg - Top-K-Avg-TV</td>
<td>0.97</td>
</tr>
<tr>
<td>Top-K%-Avg - Top-K-Avg-TD</td>
<td>0.0</td>
</tr>
<tr>
<td>Top-K%-Avg - Top-K-Avg-TV</td>
<td>0.54</td>
</tr>
<tr>
<td>Top-K-Avg-TD - Top-K-Avg-TV</td>
<td>0.0</td>
</tr>
</tbody>
</table>
For different values for K, namely; 10, 20, 50 and 100, we form sets of top scored K authors based on different scoring functions, and compute the overlapping ratio. Table 2.3 displays the average overlapping ratios of the four cases.

**Observation:** (Table 2.3): Author scoring functions Top-K-Avg and Top-K%-Avg highly overlap, and thus dependent. And, Top-K%-Avg and Top-K-Avg-TV highly overlap. Correlation between each pair of author scoring functions as shown in Table 2.4 is not conclusive.

<table>
<thead>
<tr>
<th>Scoring Measures</th>
<th>Average overlapping ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top-K-Avg - Top-K%-Avg</td>
<td>0.99</td>
</tr>
<tr>
<td>Top-K-Avg - Top-K-Avg-TD</td>
<td>-0.03</td>
</tr>
<tr>
<td>Top-K-Avg - Top-K-Avg-TV</td>
<td>0.41</td>
</tr>
<tr>
<td>Top-K%-Avg - Top-K-Avg-TD</td>
<td>0.23</td>
</tr>
<tr>
<td>Top-K%-Avg - Top-K-Avg-TV</td>
<td>-0.16</td>
</tr>
<tr>
<td>Top-K-Avg-TD - Top-K-Avg-TV</td>
<td>-0.01</td>
</tr>
</tbody>
</table>

From Figure 2.7(a), authors with nonzero Top-K-Avg-TV scores (i.e., those who have published papers in top t venues) show high correlation between their Top-K-Avg and Top-K-Avg-TV scores. This is expected because top conferences usually accept only good papers. This means that, in general, the top-k papers of an author -assuming that he has papers published in top t venues- will be his papers that are published in that venue. Thus for both Top-K-Avg and Top-K-Avg-TV, we are averaging the scores of the same set of papers for each author.
**Observation:** Top-K%-Avg and Top-K-Avg highly overlap, and are not independent.

**Observation:** Positive correlation between publication venue scores and the paper scores published in it explains the positive correlation between the author scoring functions Top-K-Avg and Top-K-Avg-TV.

**Observation:** Top-K-Avg and Top-K%-Avg are independent of Top-K-Avg-TV.

![Figure 2.7 Scatter plots of the correlation between selected author importance measures](image)

**Accuracy**

Top-K-Avg-TV is an inaccurate author scoring function because not all the authors have papers published in the selected top t publication venues. While choosing higher t may give better results, it most probably leads to the same scores obtained by Top-K-Avg. We also notice that since Top-K-Avg-TD diverts from all other scoring functions we conclude that is inaccurate.
**Observation** Table 2.3: Top-K-Avg-TD diverts from all other functions, and is viewed as inaccurate.

2.2.4.4 Publication Venue Score Function Evaluation

**Separability**

For venue’s importance computations, we used TOP-K-AVG author scores to compute the AUTH_AVG scores of publication venues. As it is clear from the graph of Figure 2.8, AUTH_AVG scores – as expected – were very low. V\textsubscript{Citation_Count} scores, on the other hand were concentrated around 0.4, which is the minimum score we assign to a venue.

![Author score distributions](image)

**Figure 2.8 Publication venues score distribution**

The venue with maximum number of citations received 6865 citations, while the number of venues with less than 110 citations is 40 (39% of the venues). This can be noticed at the peak of
$V_{\text{Citation Count}}$ curve of Figure 2.8. Normalizing the score by the maximum-number of citation venue caused the phenomena that most of the venues received low scores.

**Observation 7**: $\text{AUTH\_AVG}$ is indirectly citation based, $V_{\text{Citation Count}}$ is citation-based too. This explains the low separability of both measures.

**Independence**

For Top K overlapping experiment between venues importance measures, we choose different values for K. The overlap sizes where as follows.

<table>
<thead>
<tr>
<th>K</th>
<th>Overlapping size</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.4</td>
</tr>
<tr>
<td>10</td>
<td>0.4</td>
</tr>
<tr>
<td>15</td>
<td>0.4</td>
</tr>
<tr>
<td>20</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Authors who are ranked high do not keep publishing in top conferences all the time. This is because that they work as supervisor for their research works. The research work, usually, is performed by their graduate students. These students vary in their capabilities, this makes the papers they produce vary in level, this leads to the fact that good authors do not always publish in top venues. Moreover, since such less important papers of good authors may affect their scores, which, in turn, causes the low overlapping between Auth_Avg and $V_{\text{Citation count}}$ as venue’s importance measures. The correlation coefficient between the two measures is 0.53.
**Observation 8**: Auth\_Avg does not reflect the real venue importance. $V_{\text{Citation\_Count}}$ better reflect venue’s importance than Auth\_Avg.

**Accuracy: Consistency-based correlation**

We used the complete set of AnthP papers to check the consistency between paper importance and the importance of the venue where they published in. We grouped them according to their venues. The relation between the center of mass of each group and the venue importance was not clear, correlation is 0.44. The scatter plot of Figure 2.9(A) gives clear picture of that.

![Figure 2.9 A scatter between Center of mass and venue importance($V_{\text{Citation\_Count}}$)](image)

As it is clearly appear from the scatter plot of Figure 2.9(A), there is a positive correlation between CoMass and $V_{\text{Citation\_Count}}$ score of venue importance. We expected to get a straight or at least almost straight line, in which case, a venue with importance $r$ has a CoMass value that is greater than the CoMass values of all less important venues. What appears from the graph of Figure 2.9 this is the case. For instance, the venue with maximum $V_{\text{Citation\_Count}}$ score has CoMass that is greater than the CoMass of most of the other venues.

We extracted the papers that have at least 75% of the papers they cite available in AnthP. The number was 2135 papers. We believe that this sample represents the real complete set. We re-
conducted the same experiment over this paper set. The trend became clearer. Figure 2.9(B) shows the scatter plot of CoMass and the venues’ importance ($V_{\text{Citation_Count}}$). The correlation between CoMass and $V_{\text{Citation_Count}}$ values was relatively high. The correlation coefficient was 0.71.

Venues and the set of papers published in them exhibit what could be called a *mutually reinforcing relationship*.

**Observation 9:** $V_{\text{CITATION_COUNT}}$ score of a particular publication venue is consistent with the paper importance average of the papers published in that venue.

Table 2.6 displays top 10 venues obtained from the importance measures and the expert’s ranking. For venues ranking, Notice that the ranking based on the proposed measures highly matches the expert’s ranking.

<table>
<thead>
<tr>
<th>$V_{\text{Citation_Count}}$</th>
<th>Auth_Avg</th>
<th>Expert</th>
</tr>
</thead>
<tbody>
<tr>
<td>TODS(journal)</td>
<td>SIGMOD(conf)</td>
<td>PODS(conf)</td>
</tr>
<tr>
<td>SIGMOD(conf)</td>
<td>VLDB(conf)</td>
<td>ICDE(conf)</td>
</tr>
<tr>
<td>VLDB(conf)</td>
<td>ICDE(conf)</td>
<td>EDBT(conf)</td>
</tr>
<tr>
<td>CSUR(journal)</td>
<td>SIGIR(conf)</td>
<td>DPD(journal)</td>
</tr>
<tr>
<td>PODS(conf)</td>
<td>SIGMOD(journal)</td>
<td>SIGMOD(journal)</td>
</tr>
<tr>
<td>ICDE(conf)</td>
<td>PODS(conf)</td>
<td>SIGMOD(conf)</td>
</tr>
<tr>
<td>JACM(journal)</td>
<td>ER(conf)</td>
<td>VLDB(conf)</td>
</tr>
<tr>
<td>CACM(journal)</td>
<td>CIKM(conf)</td>
<td>VLDB(journal)</td>
</tr>
<tr>
<td>IBM(journal)</td>
<td>TODS(journal)</td>
<td>ICDT(conf)</td>
</tr>
<tr>
<td>POPL(conf)</td>
<td>TKDE(journal)</td>
<td>IS(journal)</td>
</tr>
</tbody>
</table>
2.3 Evaluating Publication Similarity Measures

2.3.1 Introduction

Searching publications based on keywords is common in digital libraries. While useful in many circumstances, the success of locating related publications based on keywords depends on the choice of keywords [JM99]. Example-based searching, i.e., locating similar/related publications to a given publication is also becoming a common search query type in digital libraries [CLK98]. In this work, we deal with the quality of publication similarity measures used for locating related-or similar-publications of a given publication. Existing publication similarity measures fall into two classes: (i) text-based similarity measures from the field of Information Retrieval (IR), such as the cosine similarity and the TF-IDF (term frequency-inverse domain frequency) model [GS98], or (ii) citation-based similarity measures based on bibliographic coupling (i.e., common citations between two publications) [RW98], co-citation (i.e., common citers of two publications) [SM73] or author-coupling (i.e., common authors between two publications). In this work, we summarize the existing publication similarity measures, and extend and evaluate them in terms of their accuracy, separability, and independence. For evaluation, we use the ACM SIGMOD Anthology [ANTH], referred to as AnthP here, a digital library of about 15,000 publications in data management.

Text-based similarity measures are based on information retrieval methodologies [GS98, C98]. As an example, using the vector space model of IR and the TF-IDF weighting scheme [GS98], the similarity between two publications may be measured by using Cosine, Jaccard,
Dice or other document measures [K97].

CiteSeer [CS] is a literature search system for searching (presently) about 730,000 computer science and bioinformatics publications, and uses three document similarity measures, namely, word vectors, LikeIt string distance [LI97], and the Common Citation Inverse Document Frequency [Giles98]. Google Scholar, Google scholarly literature search engine [GS], does not provide publication similarity functions which are needed to answer example-based queries where the user provides an example publication and asks for similar publications.

By evaluating "multiple levels" of paper similarities based on bibliographic-coupling, co-citation and author-coupling, we make the following observations:

(a) Similarity value distribution curves are similar within the same group of similarity measures, i.e., bibliographic-coupling-based, co-citation-based, and author-coupling-based measures,

(b) Citation-based and author-coupling-based similarity measures are more separable than bibliographic-coupling based measures,

(c) Citation-based and author-coupling-based similarity measures are all highly correlated. This phenomena is due to the citation and coauthorship behavior in the literature [MN04].

(d) Text-based similarity measures show low overlapping with citation-based and with author-coupling-based measures. Therefore, providing two sets of similarity scores, one text-based and another based on citation and/or author-coupling may prove to be a useful practice.
Similarity Measures between Two Publications

2.3.2 Text-Based similarities

Similarity of documents by textual content has been studied extensively in the field of Information Retrieval (IR) [BR99], where the vector space model, also called the Term Frequency-Inverse Document Frequency (TF-IDF) model, with cosine similarity measure is developed.

One can use the vector space model of text documents [C98] to evaluate the title, abstract, index terms, and body similarities between two papers. Consider a vocabulary $T$ of atomic terms $t$ that appear in each document. A document is represented as a vector of real numbers $v \in \mathbb{R}^{|T|}$, where each element corresponds to a term. Let $v_t$ denote an element of $v$ that corresponds to the term $t$, $t \in T$. The value of $v_t$ is related to the importance of $t$ in the document represented by $v$. Using the Term Frequency-Inverse Document Frequency (TF-IDF) weighting scheme [GS89], $v_t$ is defined as

$$v_t = \log(TF_{v,t} + 1) * \log(IDF_t)$$

where $TF_{v,t}$ is the number of times that term $t$ occurs in the document represented by $v$, $IDF_t = N/n_t$, $N$ is the total number of documents in the database, and $n_t$ is the total number of documents in the database that contain the term $t$.

The cosine similarity between two documents with vectors $v$ and $w$ is computed as

$$\cosine(v, w) = \frac{\sum_{i=1}^{|T|} f(v_i) \cdot f(w_i)}{\sqrt{\sum_{i=1}^{|T|} f(v_i)^2} \cdot \sqrt{\sum_{i=1}^{|T|} f(w_i)^2}}$$
where \( f( ) \) is a damping function, which is either the square-root or the logarithm function. Other similarity functions often used in the text domain include the dice measure and the Jaccard measure \([K97]\) where both change the normalizing factor in the denominator to account for different characteristics of the data.

The TF-IDF model with the \textit{cosine} similarity measure can be used to compute similarities between the titles, abstracts, index terms, and bodies of two papers. As a preprocessing step, one needs to first remove the stopping words from the terms of a document, and then use the Porter’s algorithm \([Porter]\) to stem the terms.

### 2.3.3 Citation-Based similarities

The citation-based similarity between two papers \( P_Q \) and \( P_X \) can be computed using

- \textit{bibliographic coupling}: the common citations between the two papers \([JW98]\), and
- \textit{co-citation}: the common citers between the two papers \([Small73]\).

One can then define reference similarity between two papers \( P_Q \) and \( P_X \) as

\[
\text{Sim}_{\text{References}}(P_Q, P_X) = \text{BibWeight} \times \text{Sim}_{\text{bib}}(P_Q, P_X) + (1 - \text{BibWeight}) \times \text{Sim}_{\text{co-cit}}(P_Q, P_X)
\]

where \( 0 \leq \text{BibWeight} \leq 1 \). In this section, we discuss various ways of computing bibliographic coupling and co-citation.

#### 2.3.3.1 Basic Bibliographic Coupling

\( \text{Sim}_{\text{bib}}(P_Q, P_X) \) can be defined as

\[
\text{Sim}_{\text{bib}}(P_Q, P_X) = \frac{\text{No. of common citations between papers } P_Q \text{ and } P_X}{\text{MaxB}}
\]

where \( \text{MaxB} \), is the maximum number of common citations between any two papers in the database. One problem with this definition is that it assumes that each common citation
contributes to the reference similarity equally, and ignores the effects of papers that are cornerstone works leading to significant research in the field. Such a paper is cited by all the papers that discuss an issue related to the field, and its citation by two papers needs to carry a lesser significance. Hence it is quite possible for two papers about two unrelated topics citing the same paper. To remedy this problem, we can define a $Sim_{bib}()$ measure where each common citation contributes at a different level depending on the extent to which it is “influential”. A highly important paper is cited by a large set of papers, and therefore, cannot provide an informative measure. On the other hand, if two papers cite a paper with a relatively low importance score, this citation information provides more clues towards the similarity of the two papers. Therefore, for the common citations, one can assign weights to common citations, which are inversely proportional to their importance scores as follows. If $P_1, P_2, \ldots, P_k$ are papers cited by $P_Q$ and $P_X$ then we can define a second version of $Sim_{bib}()$ as

$$Sim_{bib2}(P_Q, P_X) = (w_1 + w_2 + \ldots + w_k) / MaxW$$

where $w_i = (1-Imp(P_i))$; and $MaxW$, used as a normalization factor, is the maximum $(w_1+\ldots+w_k)$ for any $k$ in the database.

**2.3.3.2 Bibliographic Coupling with Reachability Analysis**

One can extend the bibliographic coupling similarity by considering the citations iteratively, which we refer to as reachability analysis. The formulation in Section 3.2.3.1 above can be considered as the first level (level-0) evaluation of the given citation information. We can also make use of the second-level and third-level citation information. Due to the efficiency considerations, we consider only the most basic reachability analysis cases. Normally if a paper is cited only by one of the papers (either $P_Q$ or $P_X$, but not both) then this paper is not considered
in \( \text{Sim}_{bib2}() \). Nevertheless, by following the citation information one more level, we may obtain additional similarity information. For instance, assume that paper \( P_{i} \) is cited by \( P_{Q} \), but not cited by \( P_{X} \). It is possible that, at one level below, \( P_{i} \) may be cited by one of the papers, say \( P_{j} \), which is in turn cited by \( P_{X} \) as shown in Figure 2.10.

![Figure 2.10 Illustration of citation networks](image)

From Figure 2.10, there is an indirect similarity indicator between the references of \( P_{Q} \) and \( P_{X} \). Nonetheless, this similarity indicator is not as strong as the common citations of \( P_{Q} \) and \( P_{X} \) since it is “indirect”. Therefore we can assign different weights to level-0 and level-1 common citations, leading to another variation on bibliographic coupling similarity as

\[
\text{Sim}_{bib3}(P_{Q}, P_{X}) = WL_{0} \ast \text{Sim}_{bib2}(P_{Q}, P_{X}) + WL_{1} \ast \text{Sim}_{bib2-L2}(P_{Q}, P_{X})
\]

where \( WL_{k} \) is the assigned weight for the first (i.e., \( k=0 \)) and the second (\( k=1 \)) level common citations, and \( \text{Sim}_{bib2-L2}(P_{Q}, P_{X}) \) is computed as follows. For each \( P_{i} \) that is directly cited by \( P_{Q} \) (or \( P_{X} \)) and indirectly cited by \( P_{X} \) (\( P_{Q} \)) through same paper \( P_{j} \), \( P_{i} \) is considered as if it is cited by both papers \( P_{Q} \) and \( P_{X} \), and contributes to \( \text{Sim}_{bib2-L2}(P_{Q}, P_{X}) \) by the inverse of its importance score, normalized by the corresponding \( MaxW \).

Note that second-level common citations can also be used to strengthen first-level common citations of papers \( P_{Q} \) and \( P_{X} \). Assume that \( P_{i} \) is cited by both \( P_{Q} \) and \( P_{X} \). This common citation
may lead to more similarity clues such that \( P_i \) might cite a paper \( P_k \) which is cited by \( P_Q, P_X \) or both as shown in Figure 2.11.

![Figure 2.11 Two examples illustrating the use of second-level citation information.](image1)

Obviously, citation structures in Figure 2.11 suggest a stronger coupling between the references of papers \( P_Q \) and \( P_X \), hence, these citations should also be considered among the second level. Common citations; to save space, we do not consider such a strengthening in our implementation.

Finally, third level common citations can be considered as the common citations between the citations of papers \( P_Q \) and \( P_X \). This notion is illustrated in Figure 2.12.

![Figure 2.12 An illustration of third-level citations](image2)
By taking the above-discussed cases into account, we can capture the reference similarity between two papers in terms of level-2 citations. Now we can rewrite the formulation for $Sim_{bib}(P_Q, P_X)$ to include third-level common citations as follows:

$$Sim_{bib}(P_Q, P_X) = WL_0 \cdot Sim_{bib2}(P_Q, P_X) + WL_1 \cdot Sim_{bib2-L2}(P_Q, P_X) + WL_2 \cdot Sim_{bib-L2}(P_Q, P_X)$$

We do not consider any more indirect referencing in our implementation since, at each new level, papers get more diverse in terms of their content and their citations become less significant.

![Figure 2.13 Three-level co-citation similarity illustrations.](image)

### 2.3.3.3 Co-Citation Similarity with Reachability Analysis

We can apply the same zero, one, or two-level model of bibliographic coupling to co-citation similarity in a similar manner. Different cases are illustrated in Figure 2.13, and the corresponding co-citation definitions are given next.

$$Sim_{co-cit1}(P_Q, P_X) = \left| C_{P_Q} \cap C_{P_X} \right| / MaxN$$
where $C_{P_Q}$, $C_{P_X}$ are the set of paper which cites $P_Q$ and $P_X$ respectively. And MaxN is the maximum number of common citers between any pair of papers in the database. If a citing paper is a hub (like surveys), then it will refer to many papers.

$$\text{Sim}_{\text{co-cit2-L1}}(P_Q, P_X) = \frac{\sum_{P \in Q \cap P_X} (1-\text{Imp}(P))}{\text{MaxC}}$$

where $P_{Q,X}$ is the set of papers that co-cite $P_Q$ and $P_X$ and $\text{Imp}(P_i)$ is the importance of co-citer $P_i$, $\text{MaxC}$ is the maximum similarity value of any pair of papers in the database. To involve higher level of co-citation similarity between papers,

$$\text{Sim}_{\text{co-cit3}}(P_Q, P_X) = WL_0 * \text{Sim}_{\text{co-cit2-L1}}(P_Q, P_X) + WL_1 * \text{Sim}_{\text{co-cit2-L2}}(P_Q, P_X)$$

where $WL_0 + WL_1 = 1$, which involves two levels of similarity.

$$\text{Sim}_{\text{co-cit4}}(P_Q, P_X) = WL_0 * \text{Sim}_{\text{co-cit2-L1}}(P_Q, P_X) + WL_1 * \text{Sim}_{\text{co-cit2-L2}}(P_Q, P_X) + WL_2 * \text{Sim}_{\text{co-cit2-L3}}(P_Q, P_X)$$

where $WL_0 + WL_1 + WL_2 = 1$, which involves three levels of similarity.

Yet another possibility is to exploit the textual position information where the first references to papers $P_Q$ and $P_X$ are mentioned. Assume that $P_Q$ and $P_X$ are cited together in paper $P_i$. Then one can use the textual distance in terms of the number of characters or words between the first references made to papers $P_Q$ and $P_X$. Next, one can normalize this distance (by dividing it by the
minimum distance between the first references to two papers). The smaller is the distance between the first references, the more similar the papers \( P_Q \) and \( P_X \) are. Thus, we have

\[
\text{Sim}_{\text{co-cit5}}(P_Q, P_X) = \frac{1}{\max C \sum_{P_i \in P_{Q,X}} (1 - \text{Imp}(P_i)) / \left| \text{dist}(O_{P_Q,p_i}, O_{P_X,p_i}) \right|}
\]

where \( \text{dist}(O_{P_Q,p_i}, O_{P_X,p_i}) \) is the distance between the first time \( P_Q \) and \( P_X \) are cited in paper \( P_i \).

Finally, if papers \( P_Q \) and \( P_X \) are cited together by more than one paper, then we can weigh the contribution of each such paper by its “hub score” [Kl98] of hubs and authorities. We are using here hub score of the citing paper because this relationship represents an outgoing link from the citing paper to \( P_Q \) and \( P_X \). For outgoing links, in Kleinberg’s model [KL98], hub score of the entity determines the strength of this outgoing link. Therefore if the citing paper is a good hub with a relatively high hub score then it contributes more than other citing papers rather than each citing paper contributing equally. Thus, we have

\[
\text{Sim}_{\text{co-cit6}}(P_Q, P_X) = \frac{1}{\max C \sum_{P_i \in P_{Q,X}} (1 - \text{Imp}_{\text{hub}}(P_i))}
\]

Where \( \text{Imp}_{\text{hub}}(P_i) \) is the hub importance of paper \( P_i \), and \( \max C \) is the maximum co-citation similarity between any pair of papers in the database.

### 2.3.3.4 Recursive Iteration over Co-Citer Similarity Scores

Jeh and Widom propose [JW02] recursively computing a similarity score for each pair of objects by using similarity scores of the object that have links to these pair of objects. The motivating assumption in this paper is that two objects are similar if they are referenced by similar objects or if they are referring to similar objects. This idea is formulated as follows. Define incoming and outgoing similarity between objects \( a \) and \( b \) as
\[ s_1(a,b) = \frac{C_1}{|O(a)||O(b)|} \sum_{i=1}^{|O(a)|} \sum_{j=1}^{|O(b)|} s(O_i(a), O_j(b)) \]

\[ s_2(a,b) = \frac{C_2}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b)) \]

where \( I(a) \) and \( I(b) \) are the set of objects referring to objects \( a \) and \( b \), respectively, and \( O(a) \) and \( O(b) \) are the set of objects referred by \( a \) and \( b \). Then the similarity score \( s(a, b) \) between objects \( a \) and \( b \) can be computed by combining \( s_1 \) and \( s_2 \) with some weights as follows:

\[
\text{Sim}_{\text{Recursive_co-Citer}}(a, b) = W_{\text{out-neighbor}} \cdot s_1(a, b) + (1 - W_{\text{out-neighbor}}) \cdot s_2(a, b)
\]

In other words, through this formula, the similarity between \( a \) and \( b \) is defined to be the weighted sum of average similarity between in-neighbors of \( a \) and in-neighbors of \( b \), and average similarity between out-neighbors of \( a \) and out-neighbors of \( b \). Here \( C_i \) is a constant which defines how much the similarity score of the referring objects decays before they contribute to the similarity scores of the objects they are referring to.

Considering the scientific papers, in-neighbors are the ones citing that paper and out-neighbors are the citations of each paper. Therefore if two papers are cited by similar papers and also they cite similar papers then these two papers are assumed to be similar.

### 2.3.3.5 Citation Title Similarity

If the titles of cited papers by two papers \( P_Q \) and \( P_X \) are similar then, even when the citations are not exactly the same, this information gives us clues towards the reference similarity of \( P_Q \) and \( P_X \). In order to decide as to what extent those titles are similar, we can use the TF/IDF similarity measure. We can form a document for each of \( P_Q \) and \( P_X \) by just combining the titles of the papers that they refer to. Our term vocabulary then consists of the words contained in at least one of the citations’ title. Thus, we have
\[
\text{Sim}_{\text{Cit-Title}}(P_Q, P_X) = \left( \sum_{i=1}^{\text{|v|}} f(v_i) : f(w_i) \right) / \left( \sqrt{\sum_{i=1}^{\text{|v|}} f(v_i)^2} \cdot \sqrt{\sum_{i=1}^{\text{|w|}} f(w_i)^2} \right)
\]

where \(v\) and \(w\) are vectors representing \(P_Q\) and \(P_X\) respectively, and formed by combining titles of papers that \(P_Q\) and \(P_X\) refer to.

### 2.3.3.6 Comparing Citation Author Sets

We can also compare the author sets of the citations of two papers \(P_Q\) and \(P_X\) in terms of their overlap as follows. The citations included in a paper are viewed as being related to the topic of that paper, hence they are considered as being related to each other as well. Based on this observation, after eliminating the “outliers” from the citation list, we regard the citations of each paper as a document cluster, and construct citation author sets of each cluster for the two papers \(P_Q\) and \(P_X\). Then we compare the two citation author sets for similarity.

To locate “outlier” papers (and eliminate them from the clusters and their authors from the citation author sets), we may identify those papers which discuss a topic in a different domain than the one we are interested in. As an example, if papers \(P_Q\) and \(P_X\) are computer science papers then we can ignore references to biology papers. Outliers can be detected by checking their publication venue to see whether such a paper is published in a, say, computer science journal/conference. In this process, our basic assumption is that cited papers are topically similar; so, larger number of overlapping authors between two citation author sets would imply a higher similarity between the references of two papers.

To measure the similarity of two citation author sets, French et al. [French96] uses weighted Jaccard coefficient [GS89] to measure the topical similarity of two citation author sets. For each author \(a\) in citation author sets \(A_1\) and \(A_2\), French et al. defines the weight of \(a\), \(w(a)\), to be the number of papers the author \(a\) has written. However, just using the number of papers may be
misleading because this information does not tell us anything about the quality or the impact of the papers written by an author; and we can use the importance score of an author \( a \) as the weight \( w(a) \) of the author. Given two clusters \( C_1 \) and \( C_2 \), the weighted Jaccard coefficient is calculated by the following formula:

\[
J_w(C_1, C_2) = \frac{\sum_j w(a_j)}{\sum_i \left( \sum_j w(a_j) \right) + \sum_i \left( \sum_j w(a_j) \right) - \sum_i \sum_j w(a_j)w(a_j)}
\]

where \( w(a_j) \) denotes the weight of author \( j \) in cluster \( i \). \( J_w(C_1, C_2) \) gives us a score for the topical similarity of two clusters based on the ratio of overlapping authors [French96]. Therefore given two papers \( P_Q \) and \( P_X \), we can compute

\[
Sim_{Reference-AuthorList}(P_Q, P_X) = J_w(\text{Citations}(P_1), \text{Citations}(P_2)),
\]

### 2.3.4 Author-coupling-based similarities

We can compute the author similarity between two papers directly via the number of common authors between the two papers (referred to as the \textit{Level-0-author overlap}) or indirectly via co-authorship in other papers, e.g., two different authors, each of different papers \( P_Q \) and \( P_X \), are co-authors in a third paper \( P_W \) (referred to as the \textit{Level-1-author-overlap}). We then use the following formula to compute the author similarity between two papers \( P_Q \) and \( P_X \):

\[
Sim_{Author}(P_Q, P_X) = \text{L0Weight} \times Sim_{Level-0-Author}(P_Q, P_X) + (1 - \text{L0Weight}) \times Sim_{Level-1-Author}(P_Q, P_X),
\]

where \( 0 \leq \text{L0Weight} \leq 1 \).

We already have from above a similarity measure between the authors of the citations and the citing papers. To compute Level-0 and Level-1 author-overlap, one intuitive way is, for Level-0-author-overlap, to count (and normalize) the number of overlapping authors, and, for Level-1-
author-overlap, to count (and normalize) the number of co-authorship occurrences over the other papers. Thus, we have

$$\text{Sim}_{\text{Level-0-Author}}(P_Q, P_X) = \frac{|A_{P_0} \cap A_{P_X}|}{\text{Max}A}$$

where $A_{P_0}$ and $A_{P_X}$ are the sets of authors of papers $P_Q$ and $P_X$ respectively, and $\text{Max}A$ is the maximum number of co-authors in any paper in the database.

$$\text{Sim}_{\text{Level-1-Author}}(P_Q, P_X) = \frac{1}{\text{Max}A^2} \sum_{A_i \in A_{P_0} \land A_j \in A_{P_X}} \left| [S_{A_i} - P_Q] \cap [S_{A_j} - P_X] \right|$$

where $A_{P_0}$ and $A_{P_X}$ are the sets of authors of papers $P_Q$ and $P_X$ respectively, $S_{A_i}$ is the set of papers of author $A_i \in A_{P_0}$, $S_{A_j}$ is the set of papers of author $A_j \in A_{P_X}$, and $\text{Max}A^2$ is the maximum number of papers common between all pairs of authors picked from $A_{P_0}$ and $A_{P_X}$ in any pair of papers in the database.

As another variant, one can also consider using a different mechanism so that, each shared author contributes to the similarity of the papers in different proportions, depending on his/her author importance scores—based on the assumption that the works of important authors share a common thread. By using the average of the importance scores assigned to papers of an author as the “weight” of that author, one can easily get a higher similarity score for two papers which share one author with a high importance score in comparison with two other papers which share three authors with relatively low rankings. On the other hand, in real life, with some exceptions, well-known authors are usually the ones who publish many high quality papers. Moreover, due to their prolificacy, it is not uncommon for these authors to publish on relatively different topics. Therefore we may use a weighing mechanism which leads to author weights that are inversely
proportional to their importance scores. In this way, the information that two papers share a less important author implies more towards the similarity of the papers in contrast to the case that these papers share an author with a higher importance score. Thus, we define the Level-0-author-overlap as follows:

$$\text{Sim}_{\text{Level-0-Author}}(P_Q, P_X) = \frac{\left( \sum_{i=1}^{N} (1 - \text{imp}(a_i)) \right)}{\text{Max}N}$$

where $N$ is the number of shared authors between $P_Q$ and $P_X$, $\text{imp}(a_i)$ is the importance score for a shared author, and $\text{Max}N$, used as a normalization factor, is the maximum $\sum(1-\text{imp}(a_i))$ for any $N$ in the database. Level-1-author-overlap can be formulated in the same way.

$$\text{Sim}_{\text{Level-1-Author}}(P_Q, P_X) = \frac{1}{\text{Max}A_2} \sum_{A_j \in A_Q \cap A_j \in A_X} (1 - \text{imp}(A_j))(1 - \text{imp}(A_j)) \left| S_{A_j} - P_Q \right| \left| S_{A_j} - P_X \right|$$

We can also introduce yet another level of author similarity which incorporates information about the collaborating-author set of a paper’s author. Assume that a given paper has three authors; then we can form a collaborating-author set for the authors of this paper by just taking the union of the list of co-authors that each of this paper’s authors has ever published within the past on any paper. Having formed this set for both papers $P_Q$ and $P_X$, we can compute the weighted Jaccard coefficient for these author sets as we did for the author sets of citations of the papers. This level of similarity captures the new type of co-authorship: as an example, assume that we are to assign an author similarity score for papers $P_Q$ and $P_X$. Let $P_Q$ be written by the author $A_Q$, and $P_X$ be written by the author $A_X$. Assume that there is no other third paper that these two authors collaborated together, but there are two other papers $P_S$ and $P_T$ such that $P_S$ is written by authors $A_Q$ and $A_Z$, and $P_T$ is written by authors $A_X$ and $A_Z$. There is an indirect co-authoring relationship between authors $A_Q$ and $A_X$ since they have collaborated with the same
author $A_Z$ in other papers. This information should also be evaluated as a clue towards the author similarity of two papers $P_Q$ and $P_X$ as a third-level author similarity. Hence, Level-2-author-overlap can be reformulated as

$$\text{Sim}_{\text{Level-2-Author}}(P_Q, P_X) = \frac{\sum_j w(a_{ij})w(a_{2j})}{\sum_j w(a_{ij})^2 + \sum_j w(a_{2j})^2 - \sum_j w(a_{ij})w(a_{2j})}$$

where $a_{ij}$, $a_{2j}$ are the elements of author set which is extended to include the authors who collaborated with the original authors of papers $P_Q$ and $P_X$.

### 2.3.5 Empirical evaluation of paper-similarity measures

We used AnthP in our experiments. More details of the dataset can be found under the experimental setup of the first part of this chapter.

We compare paper similarity measures in terms of their separability, independency and accuracy. For separability, we use similarity scores distribution plots. We check similarity measures independency using per wise Top-K overlapping ratio. To compute the overlapping ratio between the any two particular measures $m_1$ and $m_2$:

$$\text{Top-K overlapping ratio}(m_1, m_2) = \left( \sum_{p \in \text{AnthP}} \frac{|SS_1(p) \cap SS_2(p)|}{\min(|SS_1(p)|, |SS_2(p)|)} \right)$$

where $SS_1(p)$ and $SS_2(p)$ are the set of $K$ most similar papers to $p$ based on $m_1$ and $m_2$ respectively. For our experiments we used $K=50$. We do not consider the papers with zero similarity in the set of similar paper. That is why some similar sets contain less than 50 elements. This explains the fact that we divided by the minimum similar set size. The experiments are performed over AnthP.
We performed our comparison experiments on the citation-based and the author-coupling-based similarity measures. We choose the following similarity measures for comparison:

Bibliographic-coupling-based measures:

1. $\text{Sim}_{\text{bib1}}$
2. $\text{Sim}_{\text{bib2-L1}}$
3. $\text{Sim}_{\text{bib2-L2}}$
4. $\text{Sim}_{\text{bib2-L3}}$

co-citation-based measures

5. $\text{Sim}_{\text{co-cit1}}$
6. $\text{Sim}_{\text{co-cit1_L1}}$
7. $\text{Sim}_{\text{co-cit2-L2}}$
8. $\text{Sim}_{\text{co-cit2-L3}}$
9. $\text{Sim}_{\text{co-cit6}}$

Author-overlap-based measures

10. L0-author overlap/Counting method $\text{Sim}_{\text{AOC_L0}}$
11. L1-author overlap/Counting method $\text{Sim}_{\text{AOC_L2}}$
12. L0-author overlap/involving common author’s importance $\text{Sim}_{\text{AOW_L0}}$
13. L1-author overlap/involving common author’s importance $\text{Sim}_{\text{AOW_L1}}$

**Separability**

Figure 2.14 displays the similarity score distribution for the three groups of similarity measures; namely, bibliographic-coupling-based measures (a), co-citation-based measures (b) and author-coupling measures (c).
Figure 2.14 Similarity score distribution of citation-based and author-coupling based paper similarity measures

We have few things to notice in Figure 2.14. First is the symmetry between the score distribution curves of different levels of similarity measures within the same group. Co-citation based similarity measures showed the clearest symmetry among the three groups. The second thing to notice is that co-citation based and author coupling based similarity measures distributes better bibliographic-coupling based similarity measures over the [0, 1] interval.

**Observation:** Similarity score distribution curves are symmetric within the same group of similarity measures.

**Observation:** co-citation based and author-coupling based paper similarity measures distributes better than bibliographic-coupling.

**Independency - Overlapping between paper similarity measures**

Table 2.7 displays the Top 50 overlapping paper ratio based of the different similarity measures. The table shows that, for a given paper P, the top 50 similar papers to P based on any two different similarity measures overlap with a ratio that ranges between 0.76 and 1.0. Although they have different nature, bibliographical coupling-based similarities showed high overlapping ratio with
co-citation-based measures. Also the measures based on these two approaches highly overlap with author-based measures.

To organize our observations on Table 2.7, we divide it into regions. For example, region AA refers to the top 50 overlapping paper ratio between the similarity measures based on bibliographical coupling.

**Observation**: Top 50 overlapping paper ratio between two bibliographical coupling-based measures, or two co-citation-based measures, or two author-coupling-based range from 0.82 to 0.95, from 0.98 to 1.0, and from 0.94 to 0.97, respectively.

The high overlapping between different levels of co-citation-based or two bibliographical coupling-based measures indicates that, for a particular paper P, the set of similar papers based on one level of either of the two will be the top similar papers based on any other level of similarity based on the same measure.

Overlapping in the case of author-coupling-based measures is less because research works are usually grant-driven. That is, at the first level of author-coupling-based similarity measures, the similar papers to a particular paper P will return those papers published and supported by the same grant. Moving to higher levels, papers from different grants may appear more similar than some of those of lower level based similar papers. Consequently, they will replace, thus, this leads to lower overlapping ratio.

Overlapping in the case of bibliographical coupling-based measures is the least. The reason for that is that, although a particular paper P deals with very limited and well-defined problem, its references may cover much wider range of research areas. This diversity increases by moving to the references of these references. Based on that, similar papers to P at one level, different papers
may replace some of those papers by moving to different levels of the same measure. Notice that, by moving from lower to a higher level in bibliographical coupling-based measure we face more diversity, and in turn, less overlapping ratio.

**Observation:** Top 50 overlapping ratio between similarity measures based on bibliographical coupling and those based on co-citation ranges from 0.81 to 1.0.

The reason for this may be that, authors usually tend to cite their own previous papers. On the other hand, most of one author’s papers in general cover a small number of research interests, which makes most of their work cite similar works. This leads to the high top 50 overlapping papers ratio between the similarity measures based on bibliographical coupling and those based on co-citation.

**Observation:** Top 50 overlapping papers ratio between similarity measures based on author-coupling overlapping and those based on co-citation ranges from 0.89 to 0.97.

The explanation of observation 2 above applies here too.

**Observation:** Top 50 overlapping papers ratios between similarity measures based on author-coupling and those based on bibliographical coupling ranges from 0.76 to 0.97.

If two papers are similar based on an author-coupling measure, then it is expected that these papers will be similar based of bibliographical coupling because the common authors usually have the same or at least somehow related research interests. This makes the papers they publish commonly cite almost same works.
Table 2.7 Top 50 overlapping ratio between paper similarity measures

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBS</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TBS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.20 0.92</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.20 0.87 0.94</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.20 0.88 0.95</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.20 0.81 0.95 1.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.19 0.81 0.88 0.941.000.99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.19 0.81 0.88 0.941.000.991.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.19 0.81 0.88 0.941.000.980.990.99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.19 0.81 0.88 0.941.000.990.990.991.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.30 0.80 0.87 0.930.990.960.960.960.970.97</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.26 0.78 0.85 0.910.960.930.930.930.930.930.97</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.26 0.77 0.84 0.900.950.910.910.910.910.910.960.96</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0.25 0.76 0.83 0.890.940.890.890.900.900.940.910.95</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 2.15 shows that, by increasing the levels of bibliographical coupling more and more papers appears to be similar.

2.3.6 Conclusions

In this chapter, we have presented and evaluated three groups of paper similarity measures in terms of their (i) accuracy (ii) separability and (iii) independence. For evaluation, we have used the ACM SIGMOD Anthology, a digital library of about 15,000
Chapter 3. Improved Publication Scores via Research Pyramids

3.1 Introduction

Searching on-line Digital Libraries (ODLs) efficiently and effectively is becoming more and more important as the size and use of ODLs expand at a very high rate. As examples, (i) in Computer Science, ACM Digital Library [ACM] has close to 1 million full-text publications collected over 50 years, to search and download; (ii) in Electrical Engineering and Computer Science, IEEE Xplorer [IEX], another ODL, provides users with on-line access to more than 1,700 selected conferences proceedings; and, (iii) ScienceDirect [SD], the world’s leading scientific, technical and medical information resource celebrated its billionth article download in November’06 since launched in 1999.

Providing accurate publication scores for search results and ranking publications returned as search results accurately can help users in reducing the time spent in searching ODLs. And, better publication rankings are also useful for comparative assessments of publication venues and scientists as well. At the present time, ODLs lack effective and accurate publication rankings [RE07]. For instance, ACM Digital Library returns rankings of publication search results that are unexplained and not useful to users [ACM]. Moreover, search outputs of ODLs tend to suffer from the “topic diffusion” problem, where commonly, keyword-based searches produce a large number of publications over a large number of topics, thereby producing scores that are nonspecific to topics.
Using social networks or bibliometrics, a number of publication score functions has been defined [BP98, KL98, SIC05]. In an earlier work [SIC05], we compared and evaluated several publication score functions, including PageRank [BP98] and Authorities scores [KL98], both adopted from the www search domain, and citation-count scores from the bibliometrics domain [CS03]. We observed the *separability* problem with all of these functions which is that none of these scoring functions assigns scores that distribute well over a given scale, e.g., [0, 1]. Instead, distributions of existing publication score functions are highly skewed, and decay very fast [RS04], resulting in a much less useful comparative publication assessment capability for users. This lack of separability is caused by the “rich gets richer” phenomena [RS04, XG03], i.e., a very small number of publications with relatively high numbers of in-citations have even higher chances of receiving new citations. Yet, these scoring functions are still not very accurate, probably caused by topic diffusion in search outputs [TH02].

The research evolution model proposed in [AYA] suggests that citation relationships between research publications produce multiple, small *pyramid-like* structures, where each pyramid represents publications related to a highly specific research topic. A *research pyramid* is defined [AYA] as a set of publications that represent a highly specific research topic, and usually has a *pyramid-like* structure in terms of its citation graph [AYA]. Publications within an individual research pyramid are (i) *motivated by* earlier publications in the topic area (e.g., our paper [SEC07] is motivated in part by citations [RE07], and [AYA]), or (ii) *use techniques* proposed in publications from other research pyramids (e.g., our paper [SEC07] in part uses some of the techniques presented in citations [BP98] and [KL98]). Other “reasons” for citations may also be observed [AYA].
In this chapter, our goals are to (a) provide a solution to the ODL search output ranking problem due to the topic diffusion problem, by grouping search outputs at the most-specific (detailed) topic level and without identifying the topics themselves, (b) eliminate the low separability problem of score functions, and (c) improve the accuracy of three score functions, namely, PageRank, Authorities and Citation Count score functions. Our approach uses the research pyramid (RP-) model to improve the separability and accuracy of publication scores, and is based on normalizing publication scores within a limited scope, namely, within individual research pyramids. These improvements come from the fact that publications are now compared to their peers within their peer groups, namely, their own research pyramid publications that are on the same topic.

This chapter proposes and empirically evaluates two approaches to identify research pyramids. The first, called LB-IdentifyRP, uses Link-Based Research Pyramid identification, which captures research pyramids by identifying pyramid-like structures from the citation graph of the publication set. The second approach, called PB-IdentifyRP, uses Proximity-Based Research Pyramid identification, utilizes a graph-based proximity measure, namely SimRank [JW02], to compute similarities between publications, and then restructures the k-most-similar publications into a research pyramid.

This chapter’s contributions are:

- Validate the research pyramid model of research evolution.
- Propose and evaluate two algorithms to identify research pyramids.
- Improve publication scores in terms of accuracy and separability via publications’ research pyramids.
As a testbed, we have utilized AnthP, a publication set of 14,891 publications from the ACM SIGMOD Anthology. Our experimental results show that

- The complete publication citation graph (of AnthP) is highly clustered.
- Each cluster of the complete publication set has a pyramid-like structure in terms of the citation graph of the cluster.
- Each cluster represents a highly specific research topic.

Note that the above three findings validate the research pyramid model proposed in [AYA].

- Topic similarities decay over both the citation age and citation paths.

We used the two topic similarity decay curves to guide the RP construction.

- Within RP citation graphs, the average number of in-citations per paper varies, pointing to the importance of comparative publication scores within RPs.
- Publication scores within RPs are accurate, due to our approach where each publication is compared only to its peer (research pyramid paper) group.

The rest of the chapter is organized as follows. Section 3.2 presents publication score functions, and introduces the notion of normalizing publication scores within their research pyramids. Section 3.3 lists the properties of the research pyramid model. In section 3.4, we present two algorithms to identify research pyramids, namely, \textit{LB-IdentifyRP}, and \textit{PB-IdentifyRP}. Section 3.5 empirically validates the research pyramid model of research evolution, and evaluates the effectiveness of employing research pyramids for score separability and accuracy.

**3.2 Publication Scores**

Existing citation-based publication score functions are all based on the notion of prestige in social networks [WF94] and bibliometry [CS03]. In this chapter, as publication score functions we use:
* **PageRank** [BP98] algorithm: PageRank score $P_{PgRank}$ of a publication P is recursively computed as the normalized sum of PageRank scores of documents citing P.

* **Authority score** of the HITS (Hyperlink Induced Topic Search) algorithm [KL98]: Each document P gets two scores, namely *hub* and *authority* scores. Hub score of P is computed by summing up authority scores of the publications that P cites, and the Authority score of P, denoted by $P_{Auth}$, is computed by summing the hub scores of publication citing P.

* Normalized **citation count score**: For a particular paper P that receives $C_P$ citations, the normalized citation count $P_{CitCnt}$ is the ratio of $C_P$ to the number $C_{P_{max}}$ of in-citations of the most cited paper in the publication set.

Figure 3.1 shows that the three score functions, namely, $P_{PgRank}$, $P_{Auth}$, and, $P_{CitCnt}$ are highly skewed, and do not separate scores well. In [FP06], the author observed the skewness and inseparability of these functions independently in computer science and life sciences publications (70,000 documents in each) as well. And, it is shown [RS04, XG03] that distributions of citation-
based score functions are also highly skewed and decay very fast. We think that the cause is topic diffusion since scores are computed with respect to the full publication set. By using the research-pyramid model proposed in [AYA], we normalize scores of publications within their own research pyramids, which allows for a fair comparative assessment of publications as publications are compared to their peers in their own research pyramids.

3.3 Properties of Research Pyramid Model

We have observed three properties of research publications in three separate data sets, namely, ACM Anthology (AnthP; 15,000 publications) [AA03], and computer sciences and life sciences publication sets (each with 70,000 publications) [FP06]. In the next section, we utilize these properties in the identification of research pyramids.

**Property 1** (*Maximum Citation Age*). In online digital libraries (ODLs), most publications receive most of their in-citations within a fixed number of years after their publication dates. We refer to this value as the *Maximum Citation Age*, and denote it by $C_{AgeMax}$.

We have observed [SDE05, FP06] that, in AnthP and Computer Sciences and Life Sciences ODLs, most publications receive 90% of their in-citations in 10 years, i.e., $C_{AgeMax}=10$. Figure 3.2 presents the citation age distributions in AnthP. Below in Property 4, we give a tighter bound for citation age within which topical similarity within an RP is maintained between citing and cited publications.

In rare cases, publications may cite works older than $C_{AgeMax}$. It is found [TA04] that a great proportion of these citations are for historical reasons, which we interpret as: old cited works (a) have coarse similarity to citing papers, and (b) do not belong in the RP of the citing publication.
Property 2 (Topic Specificity Over Time). Scientific research publications quickly become very topic-specific over time, usually referable via a highly specific topic.
As illustrated in Figure 3.3, an old research pyramid that covers a certain research topic leads to instantiations of new research topics, and thus to creations of new RPs, that use techniques proposed in the publications of parent RP(s). Again, such old citations carry topical similarity between the citing and cited publication at a coarse granularity level. Possible citation exchanges between different RPs also occur and are of type “uses”, i.e., the citing paper uses techniques proposed by the cited paper.

**Example.** Codd’s paper “E. F. Codd, “A Relational Model of Data for Large Shared Data Banks”, Commun. ACM 13(6): 377-387(1970)” is about the topic relational model, and cited around 580 times. A new and more specific topic of 2000’s (i.e., citation to Codd’s work is 30+ years old), say, rank-aware join algorithms, is coarsely related to the more general topic relational model in that, a publication P in the RP of rank-aware join algorithms and citing Codd’s paper “uses” the techniques proposed in the RP of the relational model.

**Property 3** (Topic Similarity Decay Over Citation Path). After very small citation path distances, topical similarity between papers decays significantly.

From Figure 3.5, in AnthP, after a citation path of length 3, the topical similarity, as measured by SimRank, significantly decays. We refer to this value by $L_{\text{Max-TopicDecay}}$. This observation led us to build RPs of height at most 3 in the experimental results section.
Figure 3.4 SimRank score change with citation age

Figure 3.5 SimRank score change with citation distance
**Property 4** (Topic Similarity Decay over citation age). After a certain citation age, topical similarity between the citing and the cited papers significantly decays.

From Figure 3.4, in the AnthP set, after a citation age of about 5 years, the topic similarity between the citing and cited papers decays significantly. We refer to this value by $C_{\text{AgeMax-TopicDecay}}$. This observation led us to build RPs in the experimental results section such that the maximum citation age within an RP is 5 years.

Next we present the two characteristics that identify a research pyramid RP.

**RP-Property 1** (High Topic Specificity). An RP, usually organizable into a pyramid, is a set of publications that represent a highly specific research topic.

We maintain high topic specificity of RPs by applying properties 3 and 4, and keeping the height of research pyramids low (property 3). Note that we make no attempts to identify the topic associated with an RP, as our approach does not need the topics explicitly. But, in interactive environments, providing topics to users is useful [NEC].

**RP-Property 2** (Research Pyramid Construction). RPs are arranged into pyramid structures either directly by using citation graphs (i.e., the link-based approach) [AYA] or indirectly using the publication times and close proximity of papers (i.e., the proximity-based approach).

### 3.4 Research Pyramid Identification Procedures

Based on the properties of publications and characteristics of RPs, next we propose two offline research pyramid identification procedures, namely, the link-based (LB) and the proximity-based (PB) RP identification procedures.
Both procedures start by choosing a candidate root node for an RP, called the *cornerstone paper*. The paper that is located at the root of a research pyramid receives more citations than others as other publications within the research pyramid are “motivated” by it, and directly or indirectly cite it. Thus, our approach is to *identify papers with high in-citations as cornerstone papers* (i.e., the roots) of RPs to be constructed.

The *link-based* procedure locates research pyramids by identifying pyramid-like structures in the citation graph of the publication set. In summary, within an individual RP, publications are topically related [AYA], and motivated by each other (see figure 3.3) [AYA], and we use the four properties of section 4.3 to identify citations within RPs—as summarized next.

In *AnthP*, the average number of citations to a paper (“in-citations”), denoted by $C_i$, is 2.066. Note that, in our experiments, we consider only the *AnthP* citations that are completely within *AnthP*; any citation from a paper within *AnthP* to a paper that is not in *AnthP* is removed. Using Property 3 and RP-Property 1, we limit RP heights to 3. Thus, the expected number of papers within a research pyramid $RP_p$ with paper $P$ as the root and with height 3 is $|RP_p| = 1 + C_i + C_i^2 + C_i^3 \approx 15$. Of course, the actual identified RP sizes (the number of papers in $RP_p$) vary. Some RPs may deal with active research topics, and, in such cases, the number of in-citations of publications are noticeably higher than $C_i$, leading to noticeably higher RP sizes as well.

Figure 3.6-(a) presents the link-based $LB$-$IdentifyRP()$ procedure. The proximity-based $PB$-$IdentifyRP()$ is similar, except that the function call to $LB$-$FormRP()$ is replaced by the function call $PB$-$FormRP()$. The procedure $LB$-$IdentifyRP()$ (a) selects a cornerstone paper $P$ from the existing publication set (originally, say, *AnthP*) as an RP root, by simply picking the current most-cited publication (only citations that are $C_{AgeMax-TopicDecay}$ old according to property 4
above), (b) calls $LB$-$FormRP()$ to locate the RP set $RP_P$ of $P$, and (c) eliminates $RP_P$ from the current publication set $CurrAnthP$, and repeats (a)-(c) again, until no more publications are left in $CurrAnthP$.

Note that our approach in this chapter is to create distinct and nonoverlapping research pyramids. An alternative approach, not reported here due to space limitations, is to allow overlapping research pyramids as follows: Do not to eliminate any papers from the original publication set (i.e., remove step (c) above); instead, simply color each selected publication, and continue until all publications are colored, meaning that, when the algorithm ends, each paper belongs to at least one RP set, and possibly more.

The two main functions of the link-based $LB$-$IdentifyRP()$ procedure are $ChooseRoot()$ and $LB$-$FormRP()$. $ChooseRoot()$ (See Figure 3.6.b) chooses publications that are cornerstone papers, or roots of research pyramids. The function $LB$-$FormRP()$ (Figure 3.6.c) forms the $RP_P$ of a root publication $P$ by adding direct citers of $P$ (i.e., level-1 citers) into $RP_P$, and indirect citers of $P$ at a level up to the $L_{Max}$; in experiments, we choose $L_{Max}$ as 3, by following the property 3. The function $Citors(P, l, C_{AgeMax-Topic-Decay})$ returns the set of publications that cite $P$ at a level $l$ (which is at most $L_{Max}$) where the citation age of the citing paper with respect to $P$ is less than the maximum citation age $C_{AgeMax-Topic-Decays}$ (Properties 1 and 4). In more detail,

1. Paper-id $pid_P$ of root $P$ along with its level 0 is inserted into $RP_P$ and the queue $Q$, which holds paper-ids for future expansions and their distances to the root paper $P$.

2. Two-tuple $<P_i, t>$ in $Q$ is dequeued, and expanded by locating direct or indirect citers of $P_i$ so long as their levels with respect to $P$ is at most $L_{Max-Topic-Decay}$ (i.e., 3) and their citation age with
respect to $P$ (the root) is less than the maximum citation age $C_{AgeMax\text{-TopicDecay}}$ (i.e., 5). All expanded publications and their level info with respect to $P$ are inserted into the queue $Q$.

3. The above two steps are repeated until $Q$ is empty; then $RP_P$ is returned.

```
proc LB-IdentifyRP(AnthP, RP-Sets)
{
    RP-Sets := Ø;
    CurrAnthP := AnthP;
    while (CurrentAnthP = Ø)
    {
        Root:=ChooseRoot (CurrAnthP);
        RP_Root:=LB-FormRP(Root, $L_{Max\text{-TopicDecay}}$);
        RP-Sets:=RP-Sets U RP_Root;
        CurrAnthP:=CurrAnthP - RP_Root;
    }
}

(a) Procedure LB-IdentifyRP

funct ChooseRoot (CurrAnthP)
return TopCited_{TopicDecay} (CurrAnthP);

(b) Function ChooseRoot

funct LB-FormRP (P, $L_{Max}$)
{Set RP$_P$:= {P}; Queue Q;
    Q.Enqueue( {P}, 0);
    while (Q is not empty)
    {<P, i>:=Q.Dequeue;
        if (i<$L_{Max}$) then
        {CiterSet=Citers(P, i, $C_{AgeMax\text{-TopicDecay}}$);
           Q.Enqueue(CiterSet, (i+1));
           RP$_P$ = RP$_P$ +CiterSet;
        }"
    }
}
```
The function \textit{PB-FormRP()} (figure 3.6.d) of the proximity-based approach utilizes a graph-based proximity measure, namely \textit{SimRank} [JW02], to compute similarities between publications. It captures \(RPP\) of the root publication by locating publications that are most similar to \(P\) and yet (a) are linked to \(P\) with a citation path length of at most \(L_{\text{Max}}-\text{TopicDecay}\), and (b) have a citation time distance less than \(CA_{\text{AgeMax}}-\text{TopicDecay}\). \textit{SimRank} iteratively computes similarity scores between nodes in a graph \(G\) following the rule that “two nodes are similar if they are linked with similar nodes”. In other words, the \textit{SimRank} similarity between two nodes \(a\) and \(b\), \(S(a, b)\), is iteratively computed using the formula (until the similarity scores converge):

\[
S(a, b) = \left[ C / \| I(a) \| \| I(b) \| \right] * \sum_{i=1}^{I(a)} \sum_{j=1}^{I(b)} S(I_i(a), I_j(b))
\]
where $I(a)$ and $I(b)$ are sources of in-links of $a$ and $b$, respectively. $C$ is the decay factor between 0 and 1. We choose $C=0.8$ [JW02]. If $|I(a)|=0$ or $|I(b)|=0$ then $S(a, b)=0$ by definition, in the case where $a=b$, $S(a, b)=1$. The space complexity of the naive SimRank algorithm is $O(N^2)$ where $N$ is the graph size (the citation graph in publication domain). We prune as in [JW02] by considering node pairs that are near each other in the range of radius $r$. We choose $r=6$, which is twice the value of the expected research pyramid height as also explained in Section 3.5.

$PB\text{-FormRP}(\cdot)$ receives as input the root $P$, the maximum level $L_{\text{Max}}$ from root, and utilizes the maximum citation age $C_{\text{AgeMax-TopicDecay}}$ (as 5) and returns the RP set $RP_P$ of publication $P$ following the same main steps of $LB\text{-FormRP}(\cdot)$ with one main difference: the way the two-tuple $<P_b, \epsilon>$ dequeued from $Q$ is expanded, as follows:

- Top $|\text{Citers}(P_i, \epsilon, C_{\text{AgeMax-TopicDecay}})|$ similar papers, based on SimRank, to $P_i$ are identified.
  
  The number of citers of $P_i$ is used to capture the density of the RP being identified, and thus to expand RP at $P_i$ accordingly.

- The identified similar papers are added to $RP_P$ and also enqueued to $Q$ for further expansion, this time with the level increased by 1. Similar to $LB\text{-FormRP}(\cdot)$ a maximum level of $L_{\text{Max-TopicDecay}}$ (which is 3) is employed.

The advantage of $PB\text{-FormRP}(\cdot)$ over $LB\text{-FormRP}(\cdot)$ is that it successfully captures co-existing members of RP as well as those that are not reachable through any citation path from RP’s root (as illustrated in Figure 3.7 above). We give an example.

**Example.** Figure 3.7 shows two RPs; $RP_1$ and $RP_2$. $RP_1$ contains two co-existing roots $A$ and $B$. Such a case occurs when two researchers work on the same problem simultaneously. At some point of our RP identification process, $A$ will probably be recognized as a root of a new RP, say $RP_3$, as it has more in-citations than $B$. And, since $B$ is not reachable through any path from $A$,
LB-FormRP() will fail to identify B as a member of RP3. PB-FormRP() will succeed to place both A and B into RP3 in this case as B is very similar to A. A similar problem will be observed with paper C that is not reachable through any path from the root. Furthermore, LB-FormRP() may incorrectly identify F, that probably “uses” a technique proposed in A, as a member of RP3 when F is really a member of RP2 which co-exists with RP3. PB-FormRP() successfully repels F from RP3 as F is not similar to A or any of RP3’s members, based on SimRank.

![Diagram](image)

Figure 3.7 Examples where PB-FormRP() is more successful than LB-FormRP().

We observe here that PB-FormRP() may capture pyramid-like structures, but not exactly pyramid structures. SimRank computes similarity between two papers $P_1$ and $P_2$ by averaging the similarity of the citers of both. However, note that similar papers to a member of an RP will be the other members of the same RP since members of an RP are usually cited by each other (as they are motivated by each other).
3.5 Empirical Evaluations

*AnthP*, utilized as the ODL testbed here, is a publication set of 14,891 publications from the ACM SIGMOD Anthology. After eliminating citations to papers outside *AnthP*, the average in-citations per AnthP paper is 2.066.

The three citation-based publication score functions (*PageRank*, *Authorities*, and *Citation count*) have separability (high skew) and accuracy problems. We have observed that 99% of AnthP publications have scores below 0.1. This is because in-citations conform to the *Power Law* distribution, which describes the scale invariance found in many natural phenomena including publication citation graphs. As for low accuracy (probably due to “topic diffusion” problem [TH02]), different research topics differ in their citation graph densities. Thus, a paper P’s chances of receiving new citations depends on how dense the citation graph of the research topic of P is.

**Observation:** *AnthP* RPs (that represent specific research topics) have an almost normal distribution in the average in-citations received by members of an RP (figure 3.8).

For separability, first we verify the RP model on the *AnthP* set. We have experimentally observed that only 3.32% of SimRank scores are higher than 0.1, indicating that *AnthP* is highly clustered.

**Observation:** Average size of *AnthP* RP is 15 (as expected from section 3.4).

Figure 3.9 shows the distribution of the observed RP sizes within AnthP. Note that the PB approach identified larger RP sizes as it can identify co-existing RP roots and members that are not reachable through any citation path from the roots (section 3.4).

Figures 3.10 and 3.11 illustrate that $p_{PrL}$, $p_{PrPB}$, $p_{CtL}$ and $p_{CtPB}$ publication scores distribute much better over the interval [0, 1].
Figure 3.8 Variance of citation-graph densities in different topics

Figure 3.9 Score distributions of PageRank normalized within RPs.
Figure 3.10 Observed RP sizes by LP-IdentifyRP and PB-IdentifyRP

Figure 3.11 Score distributions of CitCnt normalized within RPs

Observation: For RP-based scores, the observed skew values (table 3.1) range between (-0.05) and (1.88) in the RP-based scores (zero skew indicates that the distribution is symmetric).
In comparison, the original scores showed highly skewed values that range between 8.12 and 13.04, which means that they are sharply left-skewed.

**Observation:** For RP-based scores, kurtosis values (that measure how sharply peaked a distribution is) range between (-0.26) to (2.65) (near zero Kurtosis values indicate normally peaked data).

In comparison, in the case of globally normalized scores, Kurtosis values range between (113.28) and (291.10). The enhancement of score distribution comes from the fact that publications are being compared to their peer groups, i.e., publications that belong to the same scope, and thus have the same chances of receiving new citations.

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>IQR</th>
<th>Skewness</th>
<th>Kurtosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>CitCnt</td>
<td>0.02527</td>
<td>0.01845</td>
<td>8.12</td>
<td>113.28</td>
</tr>
<tr>
<td>Auth</td>
<td>0.11352</td>
<td>0.01134</td>
<td>13.04</td>
<td>291.10</td>
</tr>
<tr>
<td>PageRank</td>
<td>0.12091</td>
<td>0.01733</td>
<td>8.84</td>
<td>134.65</td>
</tr>
<tr>
<td>LBCitCnt</td>
<td>0.55698</td>
<td>0.88462</td>
<td>-0.05</td>
<td>-1.81</td>
</tr>
<tr>
<td>LBAuth</td>
<td>0.81266</td>
<td>0.37723</td>
<td>-1.02</td>
<td>-0.26</td>
</tr>
<tr>
<td>LBPageRank</td>
<td>0.77649</td>
<td>0.46181</td>
<td>-0.80</td>
<td>-0.84</td>
</tr>
<tr>
<td>PBCitCnt</td>
<td>0.20802</td>
<td>0.21910</td>
<td>1.88</td>
<td>2.65</td>
</tr>
<tr>
<td>PBAuth</td>
<td>0.62386</td>
<td>0.32036</td>
<td>-0.07</td>
<td>-0.58</td>
</tr>
<tr>
<td>PBPageRank</td>
<td>0.55653</td>
<td>0.31615</td>
<td>0.30</td>
<td>-0.60</td>
</tr>
</tbody>
</table>

The above observations on *PageRank* ($p_{pgRank}$, $p_{pgRank-LB}$, $p_{pgRank-PB}$) also apply to *Authorities* scores ($p_{Auth}$, $p_{Auth-LB}$, $p_{Auth-PB}$). Here we report only PageRank-related results as we have observed that $p_{Auth}$ and $p_{pgRank}$ scores are highly correlated with a correlation coefficient of 0.98, and the correlation between $p_{pgRank}$ and $p_{CitCnt}$ is 0.74. [SIC05]
Figure 3.12 Distribution of no. of RPs annotated with each author

Figure 3.13 Quality values distribution of the search results.
**Observation:** Each author in *AnthP* is identified with (i.e., author papers in) 2.19 and 2.16 LB and PB research pyramids (figure 3.12).

This indicates that publications within an RP are highly related, and, thus, the identified RPs are accurate.

<table>
<thead>
<tr>
<th>Quality</th>
<th>Publication Title</th>
<th>Relevancy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Measuring The Complexity Of Join Enumeration In Query Optimization</td>
<td>9</td>
</tr>
<tr>
<td>0.487889</td>
<td>On The Complexity Of Testing Implications Of Functional And Join Dependencies</td>
<td>4</td>
</tr>
<tr>
<td>0.449827</td>
<td>Distributive Join  A New Algorithm For Joining Relations</td>
<td>8.5</td>
</tr>
<tr>
<td>0.449827</td>
<td>The Value Of Merge Join And Hash Join In Sql Server</td>
<td>2</td>
</tr>
<tr>
<td>0.449827</td>
<td>Multi Table Joins Through Bitmapped Join Indices</td>
<td>4</td>
</tr>
<tr>
<td>0.351713</td>
<td>Diag Join  An Opportunistic Join Algorithm For 1 N Relationships</td>
<td>8</td>
</tr>
<tr>
<td>0.339844</td>
<td>Utilizing Page Level Join Index For Optimization In Parallel Join Execution</td>
<td>4.5</td>
</tr>
<tr>
<td>0.315144</td>
<td>Evaluation Of Main Memory Join Algorithms For Joins With Set Comparison Join Predicates</td>
<td>8</td>
</tr>
<tr>
<td>0.287197</td>
<td>Join Algorithm Costs Revisited</td>
<td>10</td>
</tr>
<tr>
<td>0.287197</td>
<td>Heuristic And Randomized Optimization For The Join Ordering Problem</td>
<td>9.5</td>
</tr>
<tr>
<td>0.287197</td>
<td>Seeking The Truth About Ad Hoc Join Costs</td>
<td>10</td>
</tr>
</tbody>
</table>

**Sample 1**

<table>
<thead>
<tr>
<th>Quality</th>
<th>Publication Title</th>
<th>Relevancy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.148119</td>
<td>Measuring The Complexity Of Join Enumeration In Query Optimization</td>
<td>9</td>
</tr>
<tr>
<td>0.074381</td>
<td>Multiprocessor Hash Based Join Algorithms</td>
<td>5.5</td>
</tr>
<tr>
<td>0.067604</td>
<td>Efficient Processing Of Spatial Joins Using R Trees</td>
<td>7</td>
</tr>
<tr>
<td>0.062389</td>
<td>Join Processing In Database Systems With Large Main Memories</td>
<td>7.5</td>
</tr>
<tr>
<td>0.061929</td>
<td>On The Complexity Of Testing Implications Of Functional And Join Dependencies</td>
<td>4</td>
</tr>
<tr>
<td>0.060843</td>
<td>Join And Semi join Algorithms For A Multiprocessor Database Machine</td>
<td>6.5</td>
</tr>
<tr>
<td>0.060467</td>
<td>Evaluation Of Main Memory Join Algorithms For Joins With Set Comparison Join Predicates</td>
<td>8</td>
</tr>
<tr>
<td>0.059288</td>
<td>Multi Table Joins Through Bitmapped Join Indices</td>
<td>4</td>
</tr>
<tr>
<td>0.055105</td>
<td>Partition Based Spatial Merge Join</td>
<td>2</td>
</tr>
<tr>
<td>0.053342</td>
<td>Multi Step Processing Of Spatial Joins</td>
<td>2</td>
</tr>
<tr>
<td>0.05314</td>
<td>Tradeoffs In Processing Complex Join Queries Via Hashing In Multiprocessor Database Machines</td>
<td>8</td>
</tr>
</tbody>
</table>

**Sample 2**

Figure 3.14 Sample results of the “complexity of join” query. Quality is computed using RP-based (sample 1) and the globally-normalized PageRank (sample 2) along with the average relevancy scores as assigned by experts.
We used expert knowledge in the data management field to manually evaluate the accuracy of searching via RPs. For this purpose, we built a prototype keyword-based search system that

- Sends search keywords to Microsoft’s Fulltext Search engine (MsFTS), that indexes the titles of AnthP publications. In turn, MsFTS generates a list of relevant publications (result set) along with rank values (which measures text-based relevancy between the publications and the search keywords).

- For each publication \( p \) in the result set, aggregates \( p \)'s rank value returned by MsFTS with its scores, measured in two ways, namely globally-normalized PageRank and LBPageRank. We refer to this final score as the quality of paper \( p \) or \( Q(p) \). The quality scores are then used to sort the search output list so that high quality results appear at the top. The idea behind this aggregation is to push down publications that have high PageRank/LBPageRank scores and yet also have low rank values \( \text{Rank}(p) \), i.e., low relevancy to the search keywords. \( Q(p) \) is computed according to the following formula

\[
Q(p) = \text{Rank}(p) \times [\text{LB}]\text{PageRank}(p)
\]

We performed multiple searches and manually evaluated the accuracy of our system’s outputs. We observed that LBPageRank-based quality scores resulted in 16% - 25% more accurate search outputs than the PageRank-based quality scores. Accuracy was measured for the top-k publications in the result sets, where k is 10. In figures 3.13 and 3.14, we report our observations on one search experiment for the keywords “complexity of join”.

**Observation:** Quality scores of search results distribute better when computed based on RP-based publication score functions (figure 3.13).
Each publication in the Figure 3.14 is evaluated by several domain experts who assigned a score between 0 and 10 (0: non-relevant and 10: completely relevant).

**Observation**: The average expert relevancy scores assigned to publications of Samples 1 and 2 are 7.07 and 5.77 (figure 3.14).

The above observation indicates that searching via RP-based publication scores is more accurate than globally normalized publication scores.

### 3.6 Conclusions

In this chapter, we used the Research-Pyramid model proposed in [AYA] to solve the separability and accuracy problems of publication score functions. We showed that (i) normalizing publication scores within their research pyramids provides more accurate and less skewed scores, moreover (ii) ranking search results by these scores promises to give higher accuracy compared to ranking by globally normalized publication scores due to reduction of topic diffusion effect.
4.1 Introduction

Equipping keyword-based search user interfaces with efficient and user-friendly search-keyword suggesters has proven to be useful [CSE, HI06, GSG]. Studies show that users spend considerable amounts of time in search sessions to properly select keywords [DSQ], and to modify their search keywords in order to successfully locate documents that they are searching for. A search-keyword suggester may help users choose keywords properly, and thus, users are less likely to face unsuccessful search attempts. In the case of literature digital libraries (LDLs) searching for “query processing using query graphs” using CiteSeer [CS] (a digital library from the computer science domain), a list of 500 documents are identified (see figure 4.1). Furthermore, the top-5 relevant documents to the query are of low relevancy to search terms. Thus, guiding the user selection of search keywords prior to actual query execution is an important problem.

In the case of web queries, frequently, users are not sure as to how to characterize the search using keywords [DSQ], and gradually build more focused search keywords [IP06]. One scenario where users find difficulty formulating their queries is when a search term has synonyms that the user does not remember. As an example, the “Big O notation”, which is a mathematical notation
used to describe the asymptotic behavior of functions, is also referred to as “Landau notation” or “asymptotic notation” [WK]. Another scenario is when the same keyword has different meanings in different contexts, i.e., polysemy [KR97]. This may force the user to add more keywords to prune out irrelevant contexts. A possible approach to solve these problems is to provide users with immediate feedback on the digital library contents as well as on how focused their search keywords are, at an early stage, i.e., as they enter search keywords. In this chapter, we propose and evaluate such a system which we call *Search Keyword (SK-) Suggester*.

500 documents found. Only retrieving 250 documents (System busy - maximum reduced). Order: relevance to query.

Ontologies for Enterprise Integration - Fox, Grüninger (1994) (Correct) (21 citations)
The enterprise model must also support deductive query processing. In this paper, we will first present model must also support deductive query processing. In this paper, we will first present the

www.ie.utoronto.ca/EIL/public/onto_eil.ps

of the computational tasks undertaken in the processing and solution of constraint satisfaction

www.dcs.rhbnc.ac.uk/research/compint/publications/constraints/pubs-ps/con_and_universal.ps

A CMOS Chopper Opamp with Integrated Low-Pass Filter - Bakker, Huijsing (1997) (Correct) (1 citation)
ProRISC Workshop on Circuits, Systems and Signal Processing 1997 the transfer function is zero, as shown in

www.stw.nl/prorisc/workshop/proc/psz/bakker.ps.gz

RC Semantics using Rewriting Rules - Boussinot (1992) (Correct) (1 citation)
ftp-sop.inria.fr/meije/rc/rapport18-92.ps

Cpack Client-Server Routines And Utilities - Cern (Correct)
:37 7.1.7 Send character array to remote server process :38 7.1.8 Get Apollo, Cray, Decstation 3100, Ibm Rs6000, Silicon Graphics, Mips And Sun. This

wwinfo.cern.ch/asdoc/./psdir/cspack.ps.gz

Figure 4.1 Searching CiteSeer for keywords “query processing using query graphs”.

In contrast with our approach, Google [GSG] provides an SK-Suggester through Google Suggest (Figure 4.2) which employs users’ search-history repository. Google's SK-Suggester utilizes the search history of all users as keyword suggestions, and recommends keywords from (i) popular searches, (ii) searches from the current user’s search history, and (iii) current user’s bookmarks.
Studies show that this approach has multiple shortcomings that make it inadequate for the literature digital library domain [SB08].

"Content-Driven" Search-Keyword (CDSK) Suggesters, as opposed to Google’s Search-History-Driven SK-Suggester, recently received more attention [CSE, HI07, HI06]. In general, a CBSK-Suggester anticipates users’ search keywords by (i) parsing the document collection to be searched, (ii) preparing offline refinements to search keywords, and (iii) dynamically suggesting keywords as the user types his/her keywords.

In this chapter, we present the framework of a CDSK-Suggester that boosts the performance of the auto-completion tool proposed in [HI06] and implemented in the CompleteSearch engine [CSE]. We experimentally verify that the proposed enhancements result in a more scalable, higher quality, and a more user friendly CDSK-Suggester than the CompleteSearch engine autocompletion tool, and also overcomes the shortcomings of Google Suggest.
Our proposed CDSK-suggester is based on an a priori analysis of the publication collection of the digital library at hand. We (i) parse the document collection using the Link Grammar parser, a syntactic parser of English, (ii) group publications based on their “most-specific” research topics (using the notion of research pyramid [SEC07]), (iii) use the parser output to build a hierarchical structure of simple and compound tokens to be used to suggest search terms, (iv) use TextRank, a text summarization tool, to assign topic-sensitive scores to keywords, and (v) use the identified research-topics to help user aggregate focused search keywords prior to actual search query execution.

To properly establish the basis to compare our proposed SK-Suggester to the CompleteSearch engine autocompletion tool [HI06], we start with an overview of our proposed framework through an example from the literature digital library domain. In the linguistic pre-processing step, we start by tokenizing documents of the publication repository, which transforms documents into a categorized block of text called tokens. At this stage, we ignore the stopwords; later, when forming complex tokens, i.e., combining more than one token into one complex token, we consider the stopwords to guarantee syntactically and semantically correct suggestions. For performance issues, one may choose to parse properly selected parts of each document. Experimentally, we have found that it is advantageous to parse two parts: (a) publication titles since (i) the number of tokens in a title are an order of magnitude less in count than the tokens of the full document, and (ii) publication titles are significantly less likely to have ambiguous tokens (like impersonal pronouns) than the full document even though, in rare occasions, authors choose for their articles humorous, but irrelevant, names, for instance, “On

---

1 Our proposed approach can be used on any document collection given that documents are assigned to clusters where members of the same cluster are highly topically related.
saying enough already in SQL” by Michael J. Carey and Donald Kossmann. Having said this, such titles are humorous and thus easy to be remembered by users, and they have great value in navigational queries in which the user has a particular target that s/he is searching for [UZ05]. On the other hand, these titles negatively affect the performance of informational queries, in which the user is looking for sources that provide background knowledge about the search topic [UZ05]. To remedy this approach, we also suggest preprocessing (b) abstracts of publications in addition to titles. We give an example.

Example: Tokenizing the title "The Linear Complexity of a Graph” generates the following simple tokens (i) “the”, "of" and “a” which are stopwords, and (ii) "linear", “complexity”, and "graph" which are non-stopwords that are expected to appear in user search queries. Stopwords are useful in forming compound tokens through combining two or more simple tokens at a time. For instance “linear complexity” and "graph" can be linked using “of” to form the full title. Simple and compound tokens then serve as building blocks for expected user search keywords. □

Next, we organize the collection of simple and compound tokens into a token hierarchy. During a search query session, the proposed SK-Suggester recommends search keywords by traversing the token hierarchy as follows: at the beginning of the search session, the Single Token Anticipator, STA, is called to make suggestions based on the first few letters entered by the user. The STA is called each time the user enters a new search term during the session; however, the suggestion scope is continually reduced based on the previously fed terms within the same session. The suggestion scope is defined as the set of most-specific research “topics” where suggestions are extracted. Starting from simple tokens (i.e., the “most general” suggestions) towards higher levels of compound tokens (i.e., “more focused” or “more specific” suggestions), the SK-suggester guides the user towards building successful search keywords. During this
process, the user has the choice to stop further focusing his/her search terms when the following items are acceptable: (a) the **expected query result size** (i.e., the number of publications), (b) **topic-sensitive significance**, computed using TextRank [RP04], and (c) **scope**, i.e., the number of relevant research topics [SEC07].

Research topics are represented by **research pyramids**, where a research pyramid [SEC07] is a set of publications that are related to a highly specific research topic. The term “pyramid” refers to the fact that the relevant citation graph has a pyramid-like structure [SEC07]. For more details on research pyramids, see [SEC07].

Next, to compare our approach, we briefly present how the CompleteSearch engine works [CSE, HI06]. First, an index, named HYB, is prepared by preprocessing the document collection to pre-compute inverted lists of **unions of words**. Unions of words are identified using proximity measures between words separated by \( w \), that is, the pre-determined window size. To maintain a good level of locality of search, **similar** words are placed in the same block within the index in the form of document-word pairs. As the user enters his search words, relevant blocks, i.e., blocks where search terms are observed, are identified and, thus, (searching) scope, or context, narrows down to only relevant documents.

In summary, the main contribution of this chapter is to design and evaluate a content-driven SK-Suggester that (i) eliminates the drawbacks of Google’s search history-based SK-Suggester, and (ii) boosts the performance of the techniques used in the CompleteSearch. Our SK-suggester characteristics are summarized in the following table; more details will be presented in subsection 4.5.1.
Comparing to Google Suggest compared to CompleteSearch:

- Free from search history-based noisy keywords.
- Does not use user's search history, and, thus, is (i) not biased to user's interests, and (ii) not affected by user's incorrect-search characterization.
- Has one-time-only and significantly less maintenance costs than the search-history-based SK-Suggesters.
- Has no startup requirements. Our approach uses the digital library content to suggest search keywords; it does not need users' search-history to be built first.

Comparing to CompletionSearch:

- Enhanced notion of context; our view of context goes beyond the set of documents where the keywords cover all possibly relevant research topics.
- Produces significantly less numbers of pre-computed union of words, and, thus leads to smaller index sizes.
- More scalable; given similar hardware configurations for servers, our technique promises faster query execution due to small index sizes.
- Produces higher quality and more user-friendly suggestions.
- Requires less processing effort from the user side, as suggestions are isolated from the surrounding text.

Since our proposal extends CompleteSearch suggester, our approach maintains all the advantages of CompleteSearch. For instance, our approach has an excellent locality of access. Moreover, the completion of subwords and phrases is automatically supported since phrases are linguistically pre-computed.

Finally, our proposed tool does not only help user refine his search keywords, but also helps make search keywords more specific, and thus reduces the number of irrelevant documents appearing in the result set.

The rest of the chapter is constructed as follows. In section 4.2 we present the search-keyword suggestion problem and highlight the design principles of the proposed search-keyword suggester. The linguistic pre-processing step that generates the token hierarchy is described
section 4.3. In Section 4.4, we present our approach of computing topic-sensitive significance (or popularity) scores of filtered tokens, which are then used to order the computed refinements as well. In section 4.5, we describe how query refinements are made. We also briefly describe the techniques used in building the search interface. In Section 4.6 we present the experimental results. Section 4.7 concludes.

4.2 Problem Statement and Design Principles of the CDSK-Suggester

The SK-Suggester problem involves the anticipation of the search keywords that the user is attempting to specify. We define our SK-suggestion problem as follows:

**Def’n**: An SK-Suggestion query is a 5-tuple $Q(W, I, R, \beta_s, \beta_p)$, where $W$ is all possible completions of the last word that the user started typing, and $R$ and $I$ are the sets of relevant topics (research pyramids) and Islands (unions of words) from the preceding query. $\beta_s$ and $\beta_p$ are thresholds for the maximum scope and the minimum popularity required to control the number of suggestions made available to the user. Processing query $Q$ involves the following steps: (i) compute the subsets $W'$ of $W$, and a word in $W'$ that occurs in at least one island in $I$, (ii) compute $R'$ and $I'$ that form the set dominant research topics and islands respectively, where $I' \subseteq I$ and $R' \subseteq R$. Alternatively, the user may choose to be shown a fixed number of suggestions as in Google Suggest and the CompleteSearch engine, in which case the query becomes a 4-tuple of the form $Q(W, I, R, \beta_k)$.

The main design goals of the proposed SK-Suggester are:

**Principle 1**: The SK-Suggester should provide instant feedback to users prior to query execution.

Studies show that search sessions usually have multiple queries [YA06], and, that 82% of users who face unsuccessful searches modify their search keywords to better target what they are
searching for [IP06]. Further studies show that unsuccessful searches are followed by probably multiple clickthroughs before keyword refinement [YA06]. This is probably because users become more knowledgeable of what is available, and thus refine their search terms accordingly. The primary goal of the proposed SK-Suggester is to help focus user's search term to what is already available prior to performing the search, and thus reduce the time spent on search failure. To meet this goal, the content-based SK-Suggester provides instant feedback as to how focused the search keywords are to the user, prior to query execution.

**Principle 2:** The SK-Suggester should suggest linguistically valid search keywords. Having two words that frequently co-occur (or are similar to each other via a syntactic proximity measure) does not necessarily imply that we can put them together and provide the combination of the two as a meaningful suggestion. To meet this goal, we utilize an English language parser to tokenize and parse the digital library collection contents, and to build linguistically valid search keywords. An alternative approach, used in the CompleteSearch engine [CSE, HI06], is to show the user snippets of text; however, this approach needs preprocessing and more effort by the user to interpret them.

**Principle 3:** The SK-Suggester should provide guidance to the user, as (s)he builds up the search keywords. We achieve this by providing statistics on the search output prior to search execution. The proposed SK-Suggester provides the user with (i) the *scope* of each of the search keywords (i.e. the set of papers to be returned), which also warns the user against keywords that are very common and may lead to large search outputs. Notice that the number of documents where search terms are observed is not a good indicator of how focused search keywords are. Given that the user is interested in a particular research topic, some research topics (represented as research pyramids) are large because many researchers are working on that topic [SEC07], and
thus large numbers of documents may be found relevant to a user query. Consequently, the fact
that a search keyword is observed in large numbers of documents does not necessarily indicate
that the keywords are not focused enough. A better indicator of how focused search keywords
are, is the number of relevant research topics, which is the number of research pyramids.

**Principle 4**: The SK-Suggester should work online efficiently, and suggest refinements to
keywords on the fly. For efficiency, our approach involves parsing properly selected parts of
each document in the collection, and recognizes nouns, adjectives and verbs a priori. Further, all
of the time consuming tasks are performed offline. We use (i) the link grammar-based parser,
developed at Carnegie Mellon University [LGP], and (ii) TextRank text summarizing algorithm
[RP04] to identify the most significant island (or compound tokens) that will be used to suggest
refinements to user search keywords.

### 4.3 Constructing Token Hierarchy

In this section we summarize how the Token Hierarchy is built. Our discussions refer to a
repository of around 15,000 publications from ACM SIGMOD Anthology, a digital library from
the field of data management.

We first present a number of natural language (English Language (EL) properties and definitions
that will be used throughout this chapter.

**EL property 1**: *Simple Sentence types* include (1) declarative, (2) interrogative, (3) imperative,
and (4) *conditional types*. Compound sentences have the format “<simple sentence>
<conjunction> <simple sentence>”. Declarative sentences consist of a *subject* and a *predicate.*
Subject may be *simple* (i.e., consists of a noun phrase or nominative personal pronoun) or
*compound* (i.e., consists of multiple subjects combined with conjunctions). □
Figure 4.3 shows the parser output for the title "Outlier detection for high dimensional data" with multiple linkages identified within the title. Each linkage represent a linguistic relationship between two tokens (see EL Property 3 next). A full list of linkages that the parser can identify is available in [LGP]; however, only a few of them are common and observable in publication titles.

To suggest linguistically valid keywords, we utilize the linkages identified by the parser to form compound tokens out of simple tokens. The following definitions and observations form the basis of our discussion on how the token hierarchy is built.

**Def'n:** A *simple token* is a categorized block of text consisting of indivisible characters. A *compound token* is a linguistically valid combination of one or more *simple tokens*. □

As an example, "sort", "merge" and "join" are simple tokens. "sort-merge" and "sort-merge-join" are linguistically valid compound tokens; but, "join sort-merge" is linguistically invalid as the adjective should precede the noun in English.

Note that, not all linguistically valid compound tokens are “observed” in a digital library. For instance, "merge-sort join" is linguistically valid, but there is no such join algorithm in the data management field. We will refer to linguistically valid, but not necessarily observed, compound tokens as *unrealistic* compound tokens.

**EL property 2:** *Part-of-speech token types* include (1) *articles*, (2) *nouns* (*subjects* or *objects*), (3) *adjectives*, (4) *adverbs*, (5) *pronouns*, (6) *conjunctions*, (7) *verbs*, and (8) *prepositions*. □

To form realistic compound tokens, we identify *part-of-speech tokens* that are linguistically adjacent. The goal is to make keyword suggestions that make sense to the user. We use the link-
grammar-based parser proposed in [LGP] to identify linguistically adjacent tokens and build the token hierarchy of the publication set.

**EL property 3**: Possible *linguistically adjacent or related token type* cases include (1) *(subject, verb)*, (2) *(verb, object)*, (3) *(adjective, noun)*, (4) *Compound subjects*, (5) *Compound objects* (6) *(noun-possessive, noun)*, (7) *(article, noun)*, (8) *(adverb, verb)*. □

**Example**: the title "Adaptive Rank-Aware Query Optimization in Relational Databases" has the following compound tokens. (i) *(Adjective, noun)*: "relational databases", (ii) *(compound adjective)*: "rank-aware", (iii) *(adjective, noun)* "adaptive query", (4) "query optimization". Note that more complicated combinations of tokens are also possible, e.g., (i) *(compound subject, verb)*, (ii) *(verb, compound object)*, or (iii) *(simple subject, verb, object)*. □

The parser is used as a tool to parse the titles. Table 4.1 presents a list of the linkage types observed in titles. A full list of all linkage-types can be found in [LGP].

<table>
<thead>
<tr>
<th>Linkage type</th>
<th>% across ACM SIGMOD Anthology</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)</td>
<td>24.35</td>
<td>Connects adjective to noun</td>
</tr>
<tr>
<td>(AN)</td>
<td>23.43</td>
<td>Connects noun-modifier to noun</td>
</tr>
<tr>
<td>(J)</td>
<td>18.28</td>
<td>Connects preposition to its objects</td>
</tr>
<tr>
<td>(D)</td>
<td>8.42</td>
<td>Connects determinator to noun</td>
</tr>
<tr>
<td>(M)</td>
<td>8.69</td>
<td>Connects noun to post-noun modifiers</td>
</tr>
<tr>
<td>(MV)</td>
<td>4.46</td>
<td>Connects verbs to adjectives</td>
</tr>
<tr>
<td>(O)</td>
<td>6.15</td>
<td>Connects transitive verbs to objects</td>
</tr>
</tbody>
</table>
Building blocks of the token hierarchy are nouns, adjectives and verbal nouns (which are sometimes identified as verbs or gerands by the parser), altogether forming around 75% of the identified tokens in titles. The rest are stopwords, which are not totally ignored while constructing the hierarchy; we keep them in order to build meaningful compound tokens.

We construct the token hierarchy by collapsing the observed linguistically adjacent tokens into compound tokens. Any two tokens that are linked via a linkage are considered to be linguistically adjacent even if they are separated by other tokens or stopwords. A super-linkage, that is, a linkage that encompasses one or more linkages, is used to construct further compound tokens. We give an example.
**Example:** Figure 4.4 shows the parser results for the title "a model for querying annotated documents". Two levels of linkages are identified: (i) the 'A' linkage is at level 1 and used to form the compound token "annotated documents", (ii) the super-linkage 'OP' at level 2 (which encompasses the 'A' linkage) is used to form the compound token "querying annotated documents".

```
L2
|------------Op------------|
|-----------+------|
| Ds        | Mgp   |
L1
|-----A------|
| a model.n for.p querying.v annotated.v documents.n
```

Figure 4.4 levels of linkages and super-linkages.

Figure 4.5 shows the different layers of the token hierarchy. To illustrate we give an example.

**Example:** *(Island)* In the parser output shown in figure 4.3, the simple tokens *(outlier, detection, high, dimensional and data)* are located in the lowest level of the token hierarchy. Using the identified linkages of this title, we construct the compound tokens “outlier detection” and “high dimensional data”. Each one of the two compound tokens forms *an island* because there is no linkage identified between the compound tokens (only linkages between filtered tokens are considered). □

Publication titles, in turn, belong to papers that are clustered into *research pyramids*. Each research pyramid includes publications that deal with highly specific research topics [SEC07]. Consequently, the full hierarchy that is utilized by the proposed SK-Suggester consists of four layers as illustrated in figure 4.5: (i) the research pyramids layer, (ii) the title layer, (iii) the island layer, (iv) the simple/compound token layer.
4.4 Topic-Sensitive Token Weight

Each user search session can be viewed as aiming at finding information about a specific topic. This implies that the user's suggestions of search keywords should be chosen as close to the topic being targeted as possible. However, the topic being targeted is unknown to us. Thus, we use the already entered search keywords to prune out topics where these keywords are not or rarely observed. We refer to this phenomenon as the locality of search principle.

**Observation:** (The locality of search principle): Within a single search session, the user targets documents within a specific topic.

This principle allows us to narrow down the suggestion scope as the user enters more search keywords. The token-hierarchy relation is then accessed once in between keystrokes each time the user modifies the search keywords by typing one more character. Given our hypothesis that the document(s) that the user is looking for belongs to a specific research topic or few related
topics, we can reduce dramatically the diversity of the collection set by suggesting keywords from the most relevant research topic(s), which we refer to as the suggestion scope.

One issue is that a term may be used in more than one research topic. To solve this problem, we weigh tokens within each research topic. The goal is to identify the significance of tokens in each research topic, and thus prevent search-keyword refinement from topics where keywords are of lesser significance.

Figure 4.6 TextRank Input Graph [RP04], vertices are tokens and links represent linguistic adjacency.

To weigh tokens in each research-pyramid, we use the TextRank algorithm [RP04]. Briefly, TextRank algorithm constructs a graph (see figure 4.6) between a properly selected set of tokens of a document (nouns and adjectives), where an edge between two tokens exists only when they co-appear together in a window of some size. Then we apply the PageRank algorithm on the
formed graph to identify the most important tokens. PageRank is a an algorithm applied on graphs to measure relative importances of vertices [BP98]. Finally, phrases are manually constructed out of the top-scored tokens; these phrases represent keyphrases of the document.

We use TextRank at research pyramid level to compute topic-sensitive significance score of terms. We apply TextRank on each research pyramid \( r \) as follows. (i) The titles of all publications that belong to \( r \) are tokenized and annotated with part-of-speech tags using the link-grammar parser [LGP]. (ii) The tokens are filtered through a syntactic filter which selects only lexical units of certain parts of speech, namely; nouns (as well as verbal nouns and gerands) and adjectives, that give the best results [RP04]. (iii) A graph \( G_r(V,E) \) is formed using the tokens returned by the filter. \( V \), or the set of vertices, is the set of tokens. E, i.e. the edge list, is constructed such that an edge is created between any two tokens that appear in the same title. (iv) PageRank [BP98] is used to measure relative importance of all tokens. Tokens that have high PageRank scores are expected to be more significant and better representatives of the research pyramid \( r \).

4.5 Suggesting Search Keywords

In this section we present how to suggest refinements to users’ search keywords. This task is performed online; thus, real-time performance is critical.

The SK-Suggester is triggered online “in-between keystrokes”. After each keystroke, the search terms already entered are sent through an AJAX-enabled interface form to CDSK-Suggester STA (Single-Token-Anticipation) and QR (Query-Refinement) Modules at the server side (see figure 4.8). An AJAX-enabled search interface is needed in this application in order to provide an immediate, flexible, and responsive interaction [JP06, HI07].
Online steps of our approach are:

(i) **Single token anticipation.** The STA (Single-Token-Anticipation) Module (figure 4.8) is triggered each time the user starts entering a new search keyword. This module suggests completions to the incomplete term entered by the user from the current suggestion scope (by using R and I in definition 1). At the beginning, the suggestion scope is all the research pyramids and all the islands, which is the most time-consuming step [HI06].

(ii) **Search keyword refinement suggestion.** The QR (Query-Refinement) Module in figure 4.8 suggests the top-scored islands I to the user as possible refinements to the user’s search terms.

(iii) **Focusing suggestion scope** (the feedback module in figure 4.6): In this step, the subsets R’ and I’ are computed and saved in the search session status structure to be used in query refinements after the next keystroke.

(iv) **Post-processing suggestions** (the Presentation module in figure 4.8).

Next, we list and discuss the advantages of our proposed framework as compared to [HI06]:

(i) Tokens, simple and compound, observed in the same research pyramid, or multiple strongly related research pyramids, are stored within the same block as in [HI06]. This gives better locality of search and reduces I/O operations especially after suggestion scope reduces to few relevant research pyramids (see subsection 4.7.3 on conversion of suggestion scope).

(ii) A term may be used in more than one research topic. To solve this problem, we weigh tokens within each research topic. The goal is to identify the significance of tokens in each research topic, and thus prevent search-keyword refinement from topics where keywords are of lesser significance.
(iii) In Bast and Weber [HI06], suggestions are presented to the user as snippets of text from the documents in the LDL. This puts an extra burden on the user to isolate useful information from the presented text. Users usually type fast and may not have enough time for post-processing the presented suggestions. In our case, the repository is linguistically preprocessed to identify compound tokens, or islands (see figure 4.8), that will be presented to the user isolated from the surrounding text.

(iv) Scalability: in order to suggest phrases instead of single words, Bast and Weber [HI06] use text-based adjacency (within a predetermined window size) as an indicator of token-to-token proximity. We observed that this proximity measure generates long lists of possible phrases which (a) significantly increases the index size [HI06]; this problem is solved by viewing the auto-completion problem as a multi-dimensional range searching problem [HI06] and (b) may result in meaningless phrases. Our proposal uses linguistic adjacency (see EL property 3) which produces meaningful and much smaller lists. Consequently, our approach is more scalable.

In order to match completions with the being entered query word, Bast and Weber [HI06] store the positions of terms within each document in an array separate from the index. We refer to this technique by the text-based adjacency (see section 4.6.2 in the experimental results). Online processing of this extra array takes time. In our case, we use linguistic linkage-based proximity of tokens to build compound tokens (see section 4.6.2 in the experimental results). This gives more realistic and better results as tokens (nouns and adjectives, for instance) may be separated by intermediate words but still linguistically related. Thus, depending on the assigned proximity window, some close terms may be missed in the case of small widow sizes, or false positives may appear in the case of big window sizes. Our approach, in some sense, uses proximity windows with variable sizes.
4.5.1 SK-Suggester Query Execution

As stated at the beginning of section 4.2, an SK-Suggestion query is a 5-tuple \( Q(W, I, R, \beta_s, \beta_p) \) where \( W \) is all possible completions of the last word that the user started typing, i.e. the STA output, and \( I \) (Islands) is a list of the most promising refinements of the already entered search keywords. The parameter \( R \) is the set of dominantly relevant topics (or research pyramids). \( \beta_s, \beta_p \) are thresholds of the maximum scope and minimum popularity required to control the number of suggestions.

Processing the query \( Q \) involves the following steps: (i) compute the subsets \( W' \) of \( W \), and a word in \( W' \) that occurs in at least one island in \( I \), (ii) compute \( R', R' \subseteq R \) and \( I', I' \subseteq I \), that form the set dominant research topics and Islands respectively. Alternatively, the user may choose to be shown a fixed number of suggestions as in Google Suggest and the CompleteSearch engine, in which case the query becomes a 4-tuple of the form \( Q(I, R, W, \beta_k) \).

Islands are used as refinements to user queries, and vary in their sizes. To avoid proposing a long suggestion, compared to user search terms, we propose a gradual expansion of the user query as follows. Given user’s search terms \( W \), refinements of length up to \( ef^* |W| \) are presented to the user, where \( ef \) is the expansion factor, and \( |W| \) is the number of tokens in \( W \).

We empirically observed that initially choosing the expansion factor to be 1.5 gives good results allow for a gradual expansion during user’s search term construction. However, when the user chooses terms that are separated by a relatively large distance, i.e., separated by long series of words which is the case in large islands, a particular choice of an expansion factor may fail to retrieve refinements. To remedy this problem, we propose dynamically choosing \( ef \) through probing as follows. First, we choose \( ef=1.5 \). If no suggestions can be retrieved, the value of \( ef \) is
increased up to $e_{f_{max}}$ which is chosen as the length of largest island observed. The amount by which $e_f$ is increased is left to the LDL server to estimate, based on how many online users are available and whether real-time performance is achieved or not.

---

**Procedure SK-SuggesterInterface()**

**Input**  
User Input $w$: current search-terms  
Server Input: $R$, and $I$ (stored in session status)

{  
(1) For $w$
  
  (1.1) LISK <- the uncompleted search keyword in $w$
  
  (1.2) CSK <- the completed search keywords in $w$
  
  (1.3) If (CSK="" && LISK="")
          
          **STA_Module**(LISK);
  
  (1.4) Elseif (CSK="" && LISK="")
          
          **QR_Module**(CSK, LISK);
  
  (1.5) Elseif (CSK="" && LISK="")
          
          SK-List1 <- **STA_Module**(LISK);
          SK-List2 <- **QR_Module**(CSK, LISK);
          
          **Join**(SK-List1, SK-List2)
  
(2) **Presentation_Module**(W')

---

*Figure 4.7 The SK-Suggester interface procedure.*

Figure 4.7 sketches the SK-Suggester search interface procedure. The procedure receives user’s search-terms from the client, calls either the STA or the QR modules depending on $w$ as follows: ($LISK$ is the last uncompleted search keyword, and $CSK$ is the set of completed search keywords in $w$). If LISK is not empty, the STA Module is triggered. If CSK is not empty, the QR Module
is called. If both, QR and STA modules are called, and suggestions from both modules are joined such that suggestions from similar research topics are coupled, and suggestions from dominant research topics are propagated to the presentation module.

Figure 4.8 Search-Keyword Suggester Query Execution Modules

4.5.2 Guiding Statistics

Next we present a list of statistics that are used to guide the user selection of search keywords.

**Suggestion Scope:** Research Pyramid based suggestion scope considers the number of research topics (or research pyramids) where the search keywords \( w \) are observed, that is, the scope is

\[
\text{Scope}(w) = (\# \text{ of RPs where } w \text{ appears})
\]

**Topic-Sensitive Popularity of Search Keywords:** For a set of words \( W' \), the topic-sensitive popularity of \( W' \) with respect to the topic represented by some research pyramid \( r \), i.e., TSP\( (W', r) \),
r), is computed as the sum of TextRank scores of all words in W′. TextRank scores are topic-sensitive and computed within each research pyramid r. The suggestions are retrieved from dominant research pyramids computed by the feedback module in figure 4.8.

Query refinements (W′) are presented to the user in the order of their matching scores which we define as follows:

\[
M_{Score(W′,W)} = \text{Similarity}(W′, W) \times \max_{r \in \text{ERP}(W′)}[\text{TSP}(W′, r)]
\]

Where the \text{Similarity}(W′, W) is the text-based similarity between the suggested refinement W′ and the search terms entered already entered by the user. And \max_{r \in \text{ERP}(W′)}[\text{TSP}(W′, r)] is the maximum TSP value observed for the refinement W′ in all research pyramids where W′ is observed.

One more statistic used is the Specificity of Individual terms. Specificity of token t is measured as

\[
\text{Specificity}(t) = -\log\left(\frac{\text{# of Docs where t appears}}{\text{total # of Docs}}\right)
\]

We use this number to color user’s already entered terms to indicate how general his/her individual search terms are. This helps when user’s search terms consist of stopwords or terms used in a wide range of research topics.

4.6 Experimental Results

To demonstrate the effectiveness of the proposed content-based suggester, we built a prototype search tool [CBSKS], which utilizes the research pyramid model. The document collection used
includes 14,891 publications from the ACM SIGMOD Anthology, a digital library from the field of data management.

Experimental results section is organized as follows. In section 4.6.1, we list our observations on the accuracy of the linguistic pre-processing step and the quality of suggestions. Our observations and on the scalability of our approach and the convergence of suggestion scope are presented in sections 4.6.2 and 4.6.3, respectively.

4.6.1 Accuracy of Linguistic Pre-processing and Quality of Suggestions

The untagged tokens are distributed as follows.

<table>
<thead>
<tr>
<th>Token</th>
<th>% of untagged tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>of</td>
<td>17.39</td>
</tr>
<tr>
<td>a</td>
<td>14.76</td>
</tr>
<tr>
<td>in</td>
<td>14.30</td>
</tr>
<tr>
<td>and</td>
<td>14.12</td>
</tr>
<tr>
<td>the</td>
<td>11.92</td>
</tr>
<tr>
<td>to</td>
<td>4.72</td>
</tr>
<tr>
<td>an</td>
<td>4.51</td>
</tr>
<tr>
<td>on</td>
<td>3.43</td>
</tr>
<tr>
<td>with</td>
<td>3.34</td>
</tr>
<tr>
<td>from</td>
<td>1.62</td>
</tr>
</tbody>
</table>

Table 4.3 The distribution of un-tagged tokens.

In the document collection, untagged tokens are all stopwords. We do not totally ignore stopwords, but rather we use them to construct islands and connect linguistically adjacent islands. Stopword are also useful to guide the user towards his search goal as explained in the following example.
Example (Identifying Reasons for Citation): Figure 4.9 shows a list of hits for the user's query “query graph”. What could be the user’s goals behind this search? One possibility is that the user is interested in publications that use “query graph”-related algorithms as tools, such as hit (1) in figure 4.9. Another possibility is that the user is interested in publications on implementing and manipulating query graphs as in hit (2). This variance in the interests of users comes from the fact that citations in the literature have multiple reasons [SEC07, AYA, D00]. Our search-keyword suggester serves better focus search keywords to user’s goals after finding relevant publications, that is his/her reason(s) for searching for citations. This has been the goal of many recent works especially in search engine domains [UZ05, YA06, DSQ, IP06].

In the following example, we show how the proposed SK suggester also serves in early construction of successful search keywords for k-word proximity search, which is a very useful technique in narrowing down the results to more relevant ones, and at the same time allowing users to better express what they are looking for [G08, KH99].

Example (k-word Proximity Search): In figure 4.9, notice that the search keywords “query graph” are already identified as one island. Suggesting query refinements based on islands may help towards a successful proximity search. Notice that item (3) is probably irrelevant to the
query at search time since this publication most probably belongs to different research pyramid from the first and second hits; this false positive is pruned or pushed down in ranking query results. Informing users of the linguistic proximity of search terms prior to query execution can thus be useful. Furthermore, informing the user of the order in which terms appear may help eliminate false hits like hit (4) in figure 4.9, which is called k-word ordered proximity search [G08].

### 4.6.2 Scalability and Index Sizes

Our approach uses *linkages* to construct unions of words, or islands. This technique generates significantly smaller numbers of constructs than the text-based adjacency used in Bast and Weber [HI06]. For instance, by parsing the titles of around 9 thousand publications from the repository, 5,652 tokens were retrieved (6,896 tokens including stopwords). And, around 5 thousand islands are constructed. Considering text-based adjacency using the same window sizes generated 220,000 of links between tokens. These links are to be processed further to identify the most significant unions of words.

The value of using linguistic pre-processing to identify islands comes from the quality of pre-computed unions of words that can be constructed. Along with the post-processing required by the text-based adjacency approach, both factors balance the time needed to perform the linguistic pre-processing step (which is done offline).

<table>
<thead>
<tr>
<th></th>
<th>Text-based adjacency</th>
<th>Linkage-based proximity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>220145</td>
<td>4872</td>
</tr>
</tbody>
</table>
4.6.3 Convergence of Suggestion Scope

One more factor that is critical in producing search-keyword suggestions in realtime is the locality principle of search and the convergence speed of the suggestion scope.

Figure 4.10 Distribution of RP coverage

Figure 4.11 Distribution of token scope (after filtering)

Figure 4.11 shows the distribution of scope of the observed filtered tokens, i.e., excluding the stopwords.
**Observation** (figure 4.11): Filtered tokens have limited scope.

The above observation is important as it significantly effects the QR module performance. Considering this observation, we may prune the scope of the suggestions, which is the set of dominant research pyramids from where suggestions are retrieved.

Notice that some research topics may have wide range of origins. This makes the diversity of terms used in such research topics wide as well. For example, the publication "TextRank: Bringing order into text" has origins in linguistics (tokenizing and parsing), graph theory and graph-based ranking.

To measure how wide and diverse the origin of the research topic (or a research pyramid) $r$ is, we use the notion of $RP$ coverage computed as follows:

$$\text{Coverage}(r) = -\log\left(\frac{\text{# of tokens used in } r}{\text{total # of tokens}}\right)$$

This means that the higher the coverage factor of $r$ is, the less diverse its tokens become. Zero coverage of $r$ indicates that all tokens ever observed in the collection are used in $r$.

Figure 4.10 shows the distribution of coverage values of all research pyramids. Coverage values range between 3 and 8, which means that (i) the tokens within each research pyramid are of low diversity, and (ii) this signifies the importance of ranking tokens within research topics. This serves in pushing refinements extracted from dominant research topic(s) up in the suggestion list. We achieve this goal by using the topic-sensitive popularity (TSP) of search terms to order the list of computed refinements.
One critical factor that effects the STA module performance is the speed of convergence of the suggestion scope at the beginning of each search session.

We have experimentally observed that, usually within 3 characters entered by the user, the suggestion scope significantly decreases. We have also observed that the suggestion scope of STA reaches a saturation region within 4 characters entered by the user.

Figure 4.12 shows the distribution of specificity values of all tokens extracted from the document collection. High specificity value of a token \( t \) indicates that \( t \) is observed in many documents (see subsection 4.5.2 on guiding statistics). Figure 4.12 shows that high percentage of observed tokens have high specificity values, and thus, may lead to large search output lists. This further signifies the importance of warning users against such popular tokens and encourage him/her properly choose tokens of lesser specificity values.

![Histogram of Specificity(All), Specificity(All-SW)](image)

**Figure 4.12 Distribution of specificity values of (a) all tokens (left) and (b) all tokens except stopwords**

Figure 4.13 shows TextRank score distribution of all simple tokens that pass through the syntactic filter. High TextRank scores indicate popular and more significant tokens. Thus, tokens
that score high (>0.8) are content-bearing and better represent the research topic of the corresponding research pyramid. At the other extreme, low-scored tokens are usually widely used tokens.

We use topic sensitive popularity scores of tokens to order computed refinements such that the most relevant refinement from dominant research pyramids appear close to the top of the suggestion-list.

![Histogram of TextRank Score of Tokens](image)

**Figure 4.13 Distribution of TextRank scores for simple tokens.**

Since the post-processing module processes the final selected list of single token completions and the computed refinements, it is scalable and takes constant time to finalize the suggester output in the proper HTML format. Further, this module can be run at the client side using client-side scripting language.

One possible extension of our approach is to build *Hybrid search-keyword suggester* (i.e. content-driven and search-history based) as follows. Initially, when the LDL is put to service,
users’ search-history is not available, thus we suggest keywords extracted from document contents. By time, we track users’ search terms and save search terms/phrase that appear frequently in user’s queries, i.e. search-history. Hybrid search keyword suggesters, thus, combine the advantages of both approaches. That is; (i) search-history keywords are accurate, because users are less likely issue search commands with incomplete or misspelled search, (ii) search-history keywords are based on well-characterized queries. For a given search, a user may have difficulty formulating his/her queries as keywords and may repetitively modify the search keywords [SB08], (iii) provide users with personalized search-keyword suggestions. Which is an advantage of Google’s search-history based suggester.

4.7 Conclusions

We have proposed and experimentally validated a content-driven search-keyword suggester. We have experimentally shown that the proposed framework, which is optimized to work on literature digital libraries, promises a more scalable, high quality, and user-friendly search-keyword suggester when compared to its competitors. We have also shown that, as it (i) pre-computes topic-sensitive scores and (ii) directs user’s choice of search terms toward most-specific research topics; our approach has an excellent locality of access.
Chapter 5. On Popularity Quality: Growth and Decay

Phases of Publication Popularities

Within publication digital collections, citation analysis and publication score assignment are commonly used (i) to evaluate the impact of publications (and scientific collections, e.g., journals and conferences), and (ii) to order digital collection search outputs, e.g., Google Scholar. The popular citation-based web page (and, thus, publication) score measure PageRank is criticized for (a) computing only the current (and, thus, time-independent) publication scores, and (b) not taking into account the fact that citation graphs continuously evolve. Thus, the use of PageRank as is results in penalizing recent publications that have not yet developed enough popularity to receive citations. In order to overcome this inherent bias of PageRank and other citation-based popularity measures, Cho et. al. defined Page Quality for a webpage as its popularity after large numbers of web users become aware of it. Page Quality is based on the assumption that popularity evolves over time.

In this paper, we (i) experimentally validate that PageRank scores of publications, as they change over time, follow the logistic growth model that often arises in the context of population growth, (ii) model one aspect of researchers’ citation behavior in technology-driven fields (such as computer science) where authors tend not to cite old publications, (iii) argue and empirically verify that publication popularity, unlike web page popularity, has two distinct phases, namely, the popularity growth phase and the popularity decay phase, and (iv) extend the popularity
growth model developed by Cho et. al. to capture the popularity decay phase. All of our claims are empirically verified using the ACM SIGMOD Anthology digital collection.

5.1 Introduction

In the field of literature digital libraries, citation analysis is employed to evaluate the impact of publications and scientific collections (e.g., journals and conferences). It is also employed to order digital library search outputs (e.g., Google Scholar). Examples of citation-based measures are citation-count [SEC07] and PageRank [BP98]. However, as noticed by Cho et. al. [Cho05], citation-based measures compute popularity of publications based on the “current” state of a citation graph that continuously changes and evolves. Next we present two scenarios where usage of such popularity scores becomes problematic.

Example 1 (Scores for recent publications; Google Scholar [GS]). Figure 5.1 shows a sample search output from Google Scholar, a digital library search tool by Google [GS], for keywords “top-k query processing for semistructured data”. On the left-side of figure 5.1, relevant documents are ordered based on text-based relevancy to query terms and the citation-based popularity of the document. On the right-side, documents are ordered based on their publication date. The most relevant document to our query, the one entitled by “TopX: efficient and versatile top-k query processing for semistructured data”, is published in 2008, and appears at the top of the right-side search output list (where popularity didn’t affect the order of the output list). In comparison, this document is pushed down and appeared on page 5 of the left-side search output list of Google Scholar. Given that users usually check only a few pages of a returned list of documents [SEC07], this publication may not even have a chance to develop popularity unless, in time, awareness of readers increases, i.e., it becomes known to users.
Figure 5.1 Searching Google Scholar for “top-k query processing for semistructured data”

Example 2 (Scores for old publications; CiteSeer [CS]): The two plots in Figure 5.2 show citation counts of two relatively highly cited publications from CiteSeer [CS] (the observations made in this example do apply to most of the top-cited papers; check the full list posted by CiteSeer [CSMSA]). Notice that the popularities of the two publications have dropped significantly after 2004. We observe that the probability that a publication receives new citations drops as it gets older. And, we also observe that PageRank scores of old publications reach a certain value, and do not change after that, even when they are not cited anymore. The reason is that citations do not age or disappear, and as we shall shortly explain, citation graphs around old publications minimally change. We thus conclude that PageRank scores of old publications represent their peak popularity (that they achieved in the past), but not their current popularity.
This means that, even though old publications may in time be of lower interest to present users, their PageRank scores do not change.

Figure 5.2 Citation count per year for two publications that appeared in 1992 and 1994 (from CiteSeer) and cited around 300 times each.

Figure 5.3 Popularity drop of webpages, as opposed to observed in-citation life cycle of publications.
Based on the above two examples, we argue that, although PageRank is effective in capturing the peak popularity of publications, PageRank may assign inaccurate popularity scores for both old and recent publications.

In Cho et. al. [Cho05], a web-user model is introduced and a new popularity growth model of webpages is presented. Using the growth model, Cho et. al. derived a quality estimator to compute webpage quality as opposed to its peak-time popularity.

As the first contribution of this paper, we experimentally validate the popularity growth phase model of publications proposed by Cho et. al. [Cho05]. Moreover, we observe the following differences between publication citation and web-link graphs that the popularity growth model does not take in consideration: (i) publication citations do not ever disappear like web links, (ii) unlike web links, once two papers are published, no new citations between them are added, (iii) also unlike web links, new citations to old papers are very unlikely to occur, and (iv) indirect citations to a publication are of lesser effect on its PageRank score [DPSK05]. We observe that these differences result in popularity decay for old publications overtime, which we refer to as the publication popularity decay phase. In this paper, these differences guide us in extending the popularity growth model to accurately capture popularity decay of publications in technology-driven fields of study where authors tend not to cite publications that get older, and publication quality becomes less relevant. We demonstrate that our proposal successfully assigns accurate publication scores that are in turn useful for two tasks:

(i) **Ranking search results of user queries in literature digital libraries.** Accurate publication scores may help users retrieve new and yet promising publications; and new publications may contain undiscovered ideas at the frontiers of the topic of interest for users. Our extended quality
estimator identifies high quality papers, presents them to the user, and thus gives new papers a better chance to accumulate awareness more quickly.

(ii) **Modeling popularity life cycle of publications.** Coupled with the probabilistic model of researchers’ citation behavior, which we discuss in section 5.4, *popularity life cycle* of publications in different publication venues can be modeled. Cho et. al. analytically verified that the quality estimator they propose can successfully be used for pages with changing quality (growth and decay) (see figure 5.3). However, they did not investigate the popularity decay of pages [Cho05], probably because of the difficulties in capturing such web data and the complexity of web-link graph dynamics. We empirically observe in this paper that, for literature digital libraries, the popularity decay phase can be successfully modeled and integrated with the popularity growth phase.

Our two-phase publication popularity model, i.e., the popularity growth and decay model, is in heavily different than the webpage popularity model. To illustrate the differences, figure 5.3 shows two popularity growth and decay curves, one for a webpage (figure 5.3.a from Cho et. al. [Cho05]) and another for a publication (figure 5.3.b from CiteSeer [CS]). Notice that the popularity of a webpage keeps increasing as the webpage becomes known and those who “like“ it place links to it in other pages [Cho05]. After the webpage reaches the peak, its popularity decays until it reaches a steady-state popularity value [Cho05]. In comparison, the decay of publication popularity has a much different curve. Studies show that researchers rarely cite old works, especially in fast-moving fields like computer and life sciences. Consequently, we show that, by properly modeling users’ citation behavior along with accurate publication quality estimators, we obtain realistic publication popularity growth and decay curves similar to the
dashed curves of figure 5.3.b. Empirically, we observe that the majority of publication “citation count per year” curves conform to this growth and decay model.

We summarize our contributions as follows:

(1) For publication citation graphs, we experimentally validate the popularity growth phase model proposed by Cho et. al. [Cho05].

(2) We propose a probabilistic model for domain-specific publication citation behavior.

(3) We propose new definitions for popularity growth and decay for publications by coupling Cho et. al.’s model of popularity growth with our probabilistic publication citation behavior model, which we refer to as the publication quality with aging factor.

(4) We propose a mathematical model of publication popularity growth and decay, which is an extension of the popularity growth and decay model proposed by Cho et. al. [Cho05].

Next we list our major findings.

(a) Empirically, the probability of citing any publication conforms to the Weibull distribution [MW] over the age of that publication. However, the shape and scale parameters of the distribution changes with the quality of publication venues.

(b) We show that the derivative of the popularity growth function accurately represents (i.e., directly proportional to) the temporal publication popularity at any time.

(c) We observe that our definition of publication quality with aging factor matches the derivative of the popularity growth curve. This provides an analytical foundation for our growth and decay model of publication popularity.
Note that the proposed quality estimator is computationally inexpensive, and accurately captures publications’ intrinsic quality.

The rest of this chapter is organized as follows. In section 5.2, we present the popularity growth model [Cho05]. In section 5.3, we present evidences from publication citation graphs which prove that the popularity growth model is more applicable to publications than to web pages. We propose our user citation behavior model in section 5.4, and then, in section 5.5, we use this model to propose our notion of publication quality with aging factor. In section 5.6 we present our experimental results and observations. Section 5.7 concludes.

### 5.2. Page Quality and Webpage Popularity Evolution Model

Cho et. al. [Cho05], via a simple user-web model, developed a formula for the popularity growth of webpages, and then used the formula to estimate page quality [Cho05].

Publication quality, based on the web-user model, is defined as the popularity of the publication given that all possibly interested authors are aware of it and those who like it have cited it.

After getting published, a paper goes through two main phases:

(i) a **popularity growth phase** where its popularity increases as more authors become aware of it and cite it. After some time, the publication’s popularity reaches to a certain value. During the growth phase of the publication, (i) researchers develop awareness of the publication, i.e., more authors get to know it, and (ii) research problems inspired by the paper get studied by authors. This means that the longer the growth phase of a paper, the better the quality of the paper; and (iii) the authors who *like* the paper cite it in their works.
(ii) a **saturation phase**: after the transient growth phase, the publication’s PageRank score settles at a certain value, and minimally changes.

**Def’n:**

1. The *growth region* of a publication is the time interval during which the publication popularity grows.

2. The *saturation region* of a publication is the time interval that starts at the saturation point; and, afterwards, the publication usually does not receive new citations.

3. The *popularity function* $P(p, t)$ of publication $p$, is a function that computes the popularity of $p$ at time $t$.

4. *Publication quality* $Q(p)$ is the intrinsic and (saturation-time popularity) quality of a publication [Cho05].

We empirically calculate an estimation $\tilde{Q}(p)$ for the publication quality $Q(p)$ of publication $p$ as the PageRank score at the saturation region. Or, $\tilde{Q}(p) = PR(p, t_{sat})$ where $PR(p, t_{sat})$ is PageRank score of $p$ at the saturation time point $t_{sat}$.

The popularity *growth* function $P(p, t)$, proposed in Cho et. al. [Cho05], is derived as:

$$P(p, t) = Q(p)/(1 + C_1 \cdot e^{-\beta t})$$

Note that the function $P(p, t)$ is monotonically increasing with time $t$. The constant $Q(p)$ is the intrinsic quality of the publication $p$ (that is estimated as $p$’s PageRank score in the saturation region), constant $C_1$ is the rate of PageRank score growth in Cho’s PageRank score growth model. For new publications, $P(p, 0) \equiv 0$. In time, the exponent component, $e^{-\beta t}$, approaches
zero as $t$ increases, and, consequently, $P(p, t)$ converges to $Q(p)$, the intrinsic quality score of the publication, over time.

**Remark**: The popularity of a publication $p$ at time $t$ is estimated as the $p$’s PageRank score based on the citation graph at time $t$. Also, the quality of publication $p$ is estimated as the PageRank score at saturation phase [Cho05].

The above remark forms a bridge between the PageRank score change curve and Cho et. al.’s popularity growth model (and our model of publication popularity growth and decay model).

Cho et. al. [Cho05] base their model on the fact that the quality of a page is time-invariant and does not change overtime. Thus; $Q(p)$ is assumed to be a constant estimated at any time as the sum of (a) the current popularity or PageRank score of $p$, and (b) the relative popularity (PageRank) rate of change [Cho05], i.e.,

$$
\tilde{Q}(p) = PR(p, t) + \frac{1}{c} \cdot \frac{dP(p, t)}{dt} \cdot \frac{1}{PR(p, t)}
$$

(2)

where $0 < c \leq 1$ is a constant which we choose to be 0.1 as in Cho et. al. [Cho05].

A high quality publication is one with a scientific value, and one can intuitively estimate the quality of a publication based on its impact on other authors. Quantitatively, the quality can be measured as the conditional probability that an author will like the publication ($L_p$) given that s/he has became aware of it ($A_p$). Mathematically, $Q(p) = P(L_p|A_p)$, as defined by Cho et. al. [Cho05].

We argue that we need to distinguish between two measures of quality for a publication.
(i) The first measure represents the scientific value of that publication (i.e., how well-written it is, the authors follow a suitable technique to solve the research problem, …, etc). This value is time-invariant and is represented by \( Q(p) \) [Cho05].

Figure 5.4 In-citation per research pyramid (inter-research pyramid citations, i.e. citations from publications outside an RP to ones inside it).

Figure 5.5 Inter-pyramid Citation count (x-axis) vs the (average difference in publication dates of publications in a research pyramid).
(ii) The second measure represents the value of the paper to the user at the time s/he is searching the digital library. This value, in contrast to $Q(p)$, is time-dependent, especially in fast-moving fields of study. We refer to this quality measure as the *Publication Quality with Aging Factor*.

Next in section 5.3, we show that publications go through the popularity growth phase during which publications gain awareness and thus popularity. And in section 5.6, we empirically show the popularity growth curves conform to the “sigmoidal” evolution pattern derived by Cho et. al. [Cho05]. Finally, in section 5.4, we study one aspect of researcher citation behavior, and use it in section 5.5 to propose our notion of *Publication Quality with Aging Factor*.

5.3 Properties of Publication Citation Graphs and Research Pyramids

In this section we validate Cho et. al.’s popularity growth phase by (i) using PageRank as a popularity indicator, and (ii) utilizing the research-pyramid model of research evolution [SEC07, AYA], to show that popularity scores of publications converge to a steady-state value that can be estimated by equation (2) above.

We first note one difference between a publication citation graph from a web citation graph: Publication citation graph evolution behavior is to some extent more controlled than web graphs and can be anticipated. A webpage that has been on the web for a relatively long time may still receive new links (citations); old publications, however, are rarely cited [TA04, SEC07, D00]. Consequently, publication citation graphs are highly unlikely to face structural changes around relatively old publications. This special characteristic of publication citation graphs allows for developing accurate mathematical models for changes to publication’s PageRank scores, and thus better estimation of publication quality. In contrast, a web graph may face abrupt structural
changes at any time in any part of the graph. Studies show that, every week, around 8% web pages are replaced and that about 25% new links are created [NCO04].

Next we describe the research-pyramid (RP-) model [SEC07, AYA] of publications that also suggests time-dependent growth patterns in publication citation graphs. The RP-Model is based on the observation that citations between research publications produce multiple, small pyramid-like structures, where each pyramid represents publications related to a highly specific research topic [AYA]. A research pyramid is defined as a set of publications that represent a highly specific research topic, and usually has a pyramid-like structure in terms of its citation graph [AYA, SEC07].

The RP-Model suggests that publication citation graphs evolve in a time-controlled manner through the stimulation of most-specific research topics from one another as follows. A publication that deals with a new specific research problem appears, and proposes the first solution for it. More publications appear after that publication, addressing the same problem and proposing enhanced or refined solutions to that problem. In time, the research problem (i) is either solved, (ii) settles down with “good-enough” solutions, or (iii) subdivided into more specific research problems (i.e., new research pyramids) [SEC07].

Publications within an individual research pyramid are (i) motivated by earlier publications in the topic area, or (ii) use techniques proposed in publications from other research pyramids. We have observed that citations between different research pyramids conform to a highly left-skewed distribution, (figure 5.4), which indicates that as research pyramids of a particular research topic is formed and new research pyramids are instantiated, the RPs already formed receive few external citations from other research pyramids.
Figure 5.5 highlights the fact that citations between research-pyramids (inter-RP citations) mostly occur between RPs that appear close in time. The x-axis shows the number of citations that an RP receives, and the y-axis shows the average time difference, in years, between the source and the destination of the citation.

**Observation** (figure 5.5): Majority of inter-RP citations occur between RPs that appear within a few years, i.e., between an RP and those RPs stimulated by it.

Consequently, publication citation graphs are highly unlikely to face structural changes within an already constructed research pyramid because (i) citations do not disappear like web links, (ii) once two papers are published, no new links between them are added, (iii) new citations to old paper are less likely to occur, and (iv) indirect citations to a publication are of lesser effect on its PageRank score [DPSK05]. Structural changes affect only the developing (i.e., recent) research pyramids. Thus, popularity (or PageRank scores) of publications are expected to converge over
time to a steady-state value, which is the essence of the popularity growth model of Cho et. al. [Cho05].

5.4. The User Citation Behavior Model

![Weibull CDF vs Citation CDF with citation age](image)

Figure 5.7 (a) Age vs frequency of citations of publications. (b) Weibull distribution CDF

paper significantly decays over time. The best probabilistic distribution that fits the citation-age PDFs of figure 5.6 is the Weibull distribution [MW]. Figure 5.7 contains the cumulative distribution function (CDF) of the Weibull distribution, and the empirical CDF of the citation-age distribution for the data-management dataset. The two CDF curves show a high match. Using Minitab software [Minitab], we have observed that the citation age curve (figure 5.7) conforms to the Weibull distribution with the estimated parameters shape ($\gamma$)=1.548 and Scale ($\alpha$)=6.735. Thus, the probability $P(u \rightarrow v)$ of the citation from $u$ to $v$ to occur, is computed as

$$P(u \rightarrow v) = f_{weibull}(|age(u,v)|; \gamma, \alpha)$$  \hspace{1cm} (3)

where $|age(u,v)|$ is the absolute time difference (in years) between the publication years of $u$ and $v$. The probability density function of Weibull distribution is given by
assuming that \( f_{\text{weibull}}(x; \gamma, \alpha) = 0 \) for \( x < 0 \) (which is true in our case as a publication will not receive any citation if it is not published). In section 5.5, we use this formula in estimating the publication quality considering the aging factor.

### 5.5 Publication Quality with Aging Factor

Assume that a user issues a search query at time \( t \). Viewing the user as a potential author of an upcoming publication, the user will probably follow the Weibull distribution in his/her citations. i.e., the user cites a relevant publication \( v \) with probability equal to \( f_{\text{weibull}}(t - \text{t}_{\text{Year}}(v)) \) where \( \text{t}_{\text{Year}}(v) \) is the publication year of \( v \).

Thus, we argue that considering both the publication quality and the aging factor together leads to a better search output ranking. One possible way to order user search query results is to consider three factors: (i) text-based relevancy of \( v \) and the query terms, (ii) the publication quality, (iii) the probability that the user will cite the publication given the ages of relevant publications. Thus, for a given search query term \( w \), and output (publication) \( v \), one possible form of combining the three factors is as follows

\[
\text{final}_\text{score}(v) = \text{Sim}(w, v) \times \hat{P}(p, t)
\]  \hspace{1cm} (5)

where \( \text{Sim}(w, v) \) is the text-based similarity between \( w \) and \( v \), and \( \hat{P}(p, t) \) is the temporal popularity of the publication at time \( t \) which is computed as

\[
\hat{P}(p, t) = f_{\text{weibull}}(t - \text{t}_{\text{Year}}(v); \gamma, \alpha) \times P(p, t)
\]
**Def'n:** The *temporal popularity* of a publication $p$ at time $t$, $\hat{P}(p, t)$ represents users’ expected interest in $p$ at $t$.

### 5.6 Experimental Results and Observations

For the experiments, we use a publication collection of 14,891 publications from the ACM SIGMOD Anthology, a digital publication collection from the field of data management. The publications cover the time period between 1973 and 2003, and are parsed and the underlying citation graph is identified.

To verify that the developed models can accurately estimate publication’s PageRank (popularity) score, we use a leave-one-out approach as follows. The citation graph is captured at different snapshots between years 1985 and 2000. As an example, estimated future PageRank scores of papers published in 1987, say, are compared to their actual scores computed in year 2000, i.e., considering the complete publication citation graph.

Due to its probabilistic foundation, PageRank scores of individual nodes change significantly as the citation graph changes [BBWV07]. This effect makes PageRank scores of a node not directly comparable across different snapshots of an evolving graph, e.g., publication citation graphs [BBWV07, KSMC06]. To remedy this effect, we use normalized PageRank scores as proposed in [BBWV07, KSMC06].

#### 5.6.1 Popularity Growth Curves and Estimating Publication Quality

Cho et. al. briefly addressed temporal popularity changes [Cho05]. They assume that the popularity may decay, and give one possible scenario of this decay (figure 5.3.a), and showed that their quality estimator also works under that scenario. However, web dynamics are more complicated as discussed by Ntoulus et. al. [NCO04], and thus, the quality estimator may fail to
capture page quality when pages go through temporal ups and downs in their PageRank scores. This may explain the high relative error in the estimated quality value of pages that reached 67% in Cho et. al.’s experiments [Cho05]. The relative error is computed as

\[
Error(p, t) = \frac{|PR(p, t_{SatScore}) - PR(p, t)|}{PR(p, t_{SatScore})},
\]

Figure 5.8 shows the average relative error that we have observed when we applied the quality estimator for publications.

**Observation 1** (Figure 5.8). The lowest maximum average relative-error is observed when we choose the constant \(c\) in equation 2 as 0.1.

This observation matches to what Cho et. al.’s have found [Cho05]. The value for parameter \(c\) in equation 2 that generates minimal relative error in estimating quality is 0.1.

**Observation 2** (Figure 5.8). The maximum average relative error observed is 0.25 when we choose \(c\) as 0.1.
Cho et. al., however, have observed a maximum average error of 0.45 in the best case for the same choice of parameter $c$ in equation 2 (i.e, $c = 0.1$). This shows that Cho et. al.’s popularity growth model but not the decay) fits better to publication citation graphs than to web-link graphs (figure 5.8).

![Figure 5.9 Popularity growth curves for sample publications](image)

**Observation 3** (figure 5.9). Popularity scores of a publication monotonically increases over the first few years after its publication until it reaches the saturation phase.

This observation emphasizes Cho et. al.’s popularity growth model which states that the publication popularity increase conforms to the “sigmoidal” evolution pattern [Cho05].

**Observation 4** (figure 5.9). Some publications have relatively long transient phases.

Some outstanding publications may keep receiving citations much longer than regular publications, i.e., they have a long growth phase. We observe that less than 1% of the publications in our collection have relatively long popularity growth phases. Such publications are not very common in the literature as we explained in section 5.4. Nevertheless, the quality
estimator can successfully compute the future expected popularity better than PageRank, which computes popularity based on the current citation graph.

![Graphs showing popularity growth curve, citation age curve, derivative of popularity growth curve, and popularity multiplied by citation age distribution.](image)

**Figure 5.10** (a) $P(t)$ Popularity growth curve for a typical publication, (b) Citation age curve, the Weibull distribution, (c) the derivative of popularity growth curve $dP(t)/dt$ (d) Popularity multiplied by citation age distribution $P(t)$.}

### 5.6.2 Publication Quality with Aging Factor

**Observation 5.** During the saturation phase, the derivative of the popularity curve (figure 5.10.c) and the temporal popularity $\hat{P}(t)$ (figure 5.10.d) are both zero.

Within the saturation phase, the publication witnesses no increase of its PageRank (popularity) value, indicating that it receives no new citations, i.e., its temporal popularity (where temporal
popularity of a publication at any moment of time represents users’ interest in it at that time) is zero. The derivative of the popularity curve in this region is zero.

**Observation 6.** During the growth phase, the derivative of the popularity curve (figure 5.10.c) conforms to the shape of the temporal popularity $\hat{P}(t)$ curve (figure 5.10.d).

During the growth phase, the popularity of the publication increases as it gains more and more popularity, and thus more authors become aware of it. The rate of change of publication’s popularity at any point of time within the growth phase (its temporal popularity) reaches a peak value after which the rate of increase decreases, until it reaches to zero in the saturation region.

**Observation 7** (figure 5.10). The derivative of PageRank or popularity growth curve at any time $t$ is an accurate representative of its temporal popularity at that time.

This observation, a corollary of observations 5 and 6 above, provides an analytical foundation for our growth and decay model of publication popularity.

**5.7. Conclusions and Future work**

In this chapter, we have (i) experimentally validated the popularity growth phase of publications that is proposed by Cho et. al. [Cho05], (ii) proposed a probabilistic model for domain-specific publication citation behavior, and (iii) extended the *popularity growth phase* to capture publication popularity decay phase.
In this chapter, we present a tutorial for our prototype research-pyramid based search tool. The search tool is reachable via the url [http://www.elgiza.net](http://www.elgiza.net).

### 6.1. Introduction

elGiza is a Prototype Search Tool for searching Online Literature Digital Libraries (OLDLs). The name comes from the fact that this search tool is based on the Research Pyramid Model of research evolution [AYA].

The research evolution model suggests that citation relationships between research publications produce multiple, small pyramid-like structures, where each pyramid represents publications related to a highly specific research topic. We argue that computing publication score within its research pyramids results in accurate scores.

elGiza consists of three tools:

- Research-Pyramid based publication ranking tool.
- Keyword and example based search tools.
- Content-Based Search Keyword Suggester.

### 6.2. Overview of elGiza

elGiza is a research-pyramid based search tool for literature digital libraries. elGiza is equipped with the following tools:
Research-Pyramid-Based Ranking Tool for publications. This ranking tool is used to order search outputs of two search tools implemented in elGiza, namely, Keyword and example based search tools.

Content-Based Search-Keyword Suggester which can be useful for users to develop search keywords that possibly lead to a successful search. Compared to the Google Suggest search keyword suggester which is search-history based, elGiza’s search-keyword suggester is content-based; that is, it suggests keywords from the ‘most promising’ phrases observed in the digital library repository.

elGiza's repository includes (a) 106 conferences and journals, (b) 14,891 papers, and (c) 13,208 authors, all retrieved from ACM SIGMOD Anthology, a digital library of about 15,000 publications from data-management.

6.3. elGiza Ranking Tool

The elGiza scoring utility tool primarily uses the following well-known score functions for publications:

* **PageRank** algorithm: PageRank score of a publication \( P \) is recursively computed as the normalized sum of PageRank scores of documents citing \( P \).

* **Authority score of the HITS** (Hyperlink Induced Topic Search) algorithm: Each document \( P \) gets two scores, namely hub and authority scores. Hub score of \( P \) is computed by summing up authority scores of the publications that \( P \) cites, and the Authority score of \( P \) is computed by summing the hub scores of publication citing \( P \).
*Normalized citation count score:* For a particular paper $P$ that receives $C_P$ citations, the normalized citation count is the ratio of $C_P$ to the number $C_{P_{Max}}$ of in-citations of the most cited paper in the repository.

elGiza uses the identified research-pyramid structures to normalized scores of publications within their own research pyramids, which allowed for an accurate comparative assessment of publications as publications are compared to their peers in their own research pyramids.

**6.4. Keyword-Based Search**

elGiza performs keyword-based search as follows (see figure 6.1):

- Search terms are sent through an AJAX-enabled interface form to the Digital Library search engine at the server side.

![Figure 6.1 elGiza keyword-based search pipeline](image-url)
Search results are returned to the client and placed in the search results pane at the client’s side.

AJAX, *Asynchronous JavaScript And XML*, is based on JavaScript and HTTP requests. We used an AJAX-enabled interface since it makes user interfaces of web applications more responsive and interactive. The main functionality of AJAX is as follows (see figure 6.2): (i) the user generates an event which results in a JavaScript call, (ii) an XMLHttpRequest object is created and configured with a request parameters, (iii) the XMLHttpRequest object makes an asynchronous request to the web server. An object (the listener) receives the request and processes it, (iv) the object that processed the request returns an XML document to the client. (v) The XMLHttpRequest object receives the XML data, processes it, and updates the HTML DOM (Document Object Model) representing the page with the new data.

![Figure 6.2 General Sequence of Ajax Request](image)

**6.4.1 Query Execution Pipeline**

At the server side, the Literature Digital Library search engine processes search queries as follows (see figure 6.1):
(i) Search keywords are passed to Microsoft’s Fulltext Search engine (MsFTS) that indexes the titles of elGiza’s repository publications. Any commercial or open source fulltext search engine can be used for that purpose.

(ii) In turn, MsFTS generates a list of relevant publications (result set) along with *rank values* (which measures text-based relevancy between the relevant document and the search keywords).

(iii) For each publication \( p \) in the result set, the rank value of \( p \) (returned by MsFTS) is aggregated with its prestige scores \( \text{PubScore}(p) \). The aggregation function \( g() \) as well as the publication scoring function used vary based on user’s search settings. Publication scoring functions used to order search results are:

(a) Globally-normalized PageRank, HITS and Citation-Count scores, and

(b) Research-Pyramid-Based PageRank, HITS or Citation-Count scores.

The aggregated function output for a paper \( p \) in the search result set is called the *quality score* of \( p \), or \( Q(p) \). The quality scores of all papers in the search result set are used to sort the search output list so that high quality results appear at the top. The idea behind this aggregation is to push down scores of publications that have high scores (PageRank…) and yet also have low rank values \( \text{Rank}(p) \), i.e., low relevancy to search keywords. One way to compute \( Q(p) \) is according to the following formula

![Figure 6.3 Sample search output format](image)
Q(p) = Rank(p) * PubScore(p)

Identifying research pyramid structures as well as computing research-pyramid-based scores of publications are performed offline.

At this stage of query processing, search results are passed to the Presentation Module in XML format as shown in figure 6.3. The presentation module presents search result the way the user wants, i.e. grouped or planner. Cascaded Style Sheets CSS and XHTML technology are used to render search output at the client side web browser. This data and presentation layout separation facilitates changing the appearance and layout of search outputs.

Figure 6.4 elGiza Keyword-based Search Page
6.4.2 elGiza Search-Keyword Suggester

elGiza utilizes a "content-based" search keyword suggester, as opposed to Google’s Search-History-based search keyword suggester, for digital libraries. Repository content is linguistically pre-analyzed (offline) to extract important tokens and language constructs (e.g. adjective-noun, or adverb-verb constructs). The extracted constructs and tokens are used to anticipate users’ search keywords. elGiza’s search keyword suggester also provides the user with statistics on how focused his/her current as well as suggested keywords are.

![Figure 6.5 Sample Output for the Search Keyword Suggester](image)

Figure 6.5 shows the set of suggestions for the search terms “distributed al”. The suggestions are presented to the user with the following fields

**Token**: is the output of the single token anticipator (the STA module).
**Score**: the maximum topic-sensitive score of the token (the STA output).

**Scope**: Research Pyramid based suggestion scope considers the number of research topics (or research pyramids) where the search keywords \( w \) are observed, that is, the scope is

\[
\text{Scope}(w) = (\text{# of RPs where } w \text{ appears})
\]

**Refinements**: Refinements for the user keywords.

**The M-Score (Topic-Sensitive Popularity of Search Keywords)**: For a set of words (\( W' \)), the topic-sensitive popularity of \( W' \) with respect to the topic represented by some research pyramid \( r \), i.e., TSP(\( W' \), \( r \)), is computed as the sum of TextRank scores of all words in \( W' \). TextRank scores are topic-sensitive and computed within each research pyramid \( r \). The suggestions are retrieved from dominant research pyramids.

Query refinements (\( W' \)) are presented to the user in the order of their *matching scores* which we define as follows:

\[
M_{\text{Score}}(W', W) = \text{Similarity}(W', W) \times \text{Max}_{r\in\text{ERP}(W)}[\text{TSP}(W', r)]
\]

Where the \( \text{Similarity}(W', W) \) is the text-based similarity between the suggested refinement \( W' \) and the search terms entered already entered by the user. And \( \text{Max}_{r\in\text{ERP}(W)}[\text{TSP}(W', r)] \) is the maximum TSP value observed for the refinement \( W' \) in all research pyramids where \( W' \) is observed.

**6.4.3 Parameters of Keyword-Based Search**

Figure 6.6 shows the form through which the user chooses his/her keyword based search settings. Through this form, the user decides upon the following search parameters.
6.4.3.1 The "Order By" Parameter

This parameter sets how search results are sorted. Currently, results can be ordered based on the following features:

- 'GN PageRank': the Globally Normalized PageRank Score of publications
- 'GN Citation Count': the Globally Normalized Citation Count Score of publications
- 'GN Auth': the Globally Normalized Authority Score of publications
- 'GN Hub': the Globally Normalized Hub Score of publications
- 'LB PageRank': PageRank Score Normalized within research pyramids, LBRP-Based
- 'LB Citation Count': Citation count Normalized within research pyramids, LBRP-Based
- 'LB Auth': Authority Score Normalized within research pyramids, LBRP-Based
- 'LB Hub': Hub Score Normalized within research pyramids, LBRP-Based
- 'PB PageRank': PageRank Score Normalized within research pyramids, LBRP-Based
- 'PB Citation Count': Citation count Normalized within research pyramids, LBRP-Based
- 'PB Auth': Authority Score Normalized within research pyramids, LBRP-Based
- 'PB Hub': Hub Score Normalized within research pyramids, LBRP-Based
- 'Quality Score': Quality Score of publications. Quality score is computed in various ways as we will illustrate next.
- 'Publication Year': Publication year of publications.

6.4.3.2 Quality Score Computation

Quality of search results (or publications) is computed as multiplication of two parameters:

The Quality score multiplicand (M): currently M can be selected from two values (1.0) or the text-based similarity score.
**Score of publication (N)**: the user has the following alternatives to choose from:

- 'GN PageRank': the Globally Normalized PageRank Score of publications
- 'GN Citation Count': the Globally Normalized Citation Count Score of publications
- 'GN Auth': the Globally Normalized Authority Score of publications
- 'GN Hub': the Globally Normalized Hub Score of publications
- 'LB PageRank': PageRank Score Normalized within research pyramids, LBRP-Based
- 'LB Citation Count': Citation count Normalized within research pyramids, LBRP-Based
- 'LB Auth': Authority Score Normalized within research pyramids, LBRP-Based
- 'LB Hub': Hub Score Normalized within research pyramids, LBRP-Based
- 'PB PageRank': PageRank Score Normalized within research pyramids, LBRP-Based
- 'PB Citation Count': Citation count Normalized within research pyramids, LBRP-Based
- 'PB Auth': Authority Score Normalized within research pyramids, LBRP-Based
- 'PB Hub': Hub Score Normalized within research pyramids, LBRP-Based

![elGiza Keyword-Based Search](image)

**Figure 6.6 Search Settings Form - Keyword-based Search Tool**
6.4.3.3 The Group By Parameter

Users have the choice of grouping search results or

- 'none' : Results are presented to user as one group.
- 'LBRP' : Results are presented to user as groups, where groups are the research Pyramids identified following the link-based approach.
- 'BRP' : Results are presented to user as groups, where groups are the research Pyramids identified following the proximity-based approach.

6.4.3.4 The Output Format Parameter

The Presentation Module of elGiza can present search results to user in one of three different formats:

- 'XML' : XML files can then be used by user to view search results in any way depending on the Cascaded Style Sheet he/she may provide.
- 'ACM DL-Like' : Search results are presented to user in ACM DL-like fashion.
- 'Tabular' : Search results are presented to user as table.

6.4.4 Search Output Interpretation

Figure 6.7 shows sample output of elGiza’s keyword-based search for the keywords “query graphs”.

6.4.4.1 Relevant Publications Pane

In this pane, a list of relevant publications to user’s query terms. The following information is presented for each search output item:
**Publication ID**: an integer that uniquely identify each publication in elGiza’s repository.

**Text-Based Relevancy**: the text-based relevancy score between the publication title and the query terms entered by the user.

**Quality**: the quality of the publication.

**Title**: The publication title.

**Author(s)**: List of authors names.

**Appeared in**: the bane of the publication venue (e.g. VLDB) and its type (e.g. Journal, Conference).

**Year**: publication year.

**Importance Scores**: Citation-Based scores of the publication (globally normalized and locally normalized with research pyramids).

**Abstract**: the abstract of the publication.

**Links**: a link to similar publications (query by example).

**Research pyramid ID of the publication**: the research-pyramid ID to which the publication belongs.

**6.4.4.2 Search Summary Pane**

This small textbox shows search parameters chosen by the user and two search statistics, namely; the search time in seconds and the number of relevant documents retrieved.
6.5 Example-Based Search

Figure 6.8 shows a snapshot of sample example-based search output. Information about each publication in output set is similar to keyword-based search tool. For each search result, a link to “similar” publications is available.
Example-based search output

Publication ID: 14702
Similarity Score: 0.9290 Quality: 0.0290
Title: Seeking The Truth About Ad Hoc Join Costs
Authors: Laura M. Haas, Myron Limy, Amit Shukla, Michael J. Carey
Appeared in: VLDB (Journal) Year 1999.

<table>
<thead>
<tr>
<th>Scores</th>
<th>Citation Count</th>
<th>PageRank</th>
<th>Auth Hub</th>
</tr>
</thead>
<tbody>
<tr>
<td>Globally Normalized</td>
<td>0.9115</td>
<td>0.1151</td>
<td>1.0356</td>
</tr>
<tr>
<td>LB-RP Based</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>PB-RP Based</td>
<td>0.9606</td>
<td>0.9632</td>
<td>1.0632</td>
</tr>
</tbody>
</table>

Abstract: Not available
Links: Similar papers

Identified as member of the following Research Pyramid(s):
- Link-Based: RPID: 510 - Membership score: 1
- Link-Based: RPID: 215 - Membership score: 4

Publication ID: 12678
Similarity Score: 0.9290 Quality: 0.0290
Title: Sort Versus Hash Revisited
Authors: Goetz Graefe, Leonard D. Shapiro, Ami Guttman
Appeared in: VLDB (Journal) Year 1998.

<table>
<thead>
<tr>
<th>Scores</th>
<th>Citation Count</th>
<th>PageRank</th>
<th>Auth Hub</th>
</tr>
</thead>
<tbody>
<tr>
<td>Globally Normalized</td>
<td>0.9125</td>
<td>0.1151</td>
<td>1.0356</td>
</tr>
<tr>
<td>LB-RP Based</td>
<td>0.9606</td>
<td>0.9632</td>
<td>1.0632</td>
</tr>
<tr>
<td>PB-RP Based</td>
<td>0.9152</td>
<td>0.9152</td>
<td>0.9152</td>
</tr>
</tbody>
</table>

Abstract: Not available
Links: Similar papers

Identified as member of the following Research Pyramid(s):
- Link-Based: RPID: 338 - Membership score: 1
- Link-Based: RPID: 22 - Membership score: 3

Figure 6.8 Sample (ACM-Like) Search Output for Example-based Search Tool
Chapter 7. Conclusions and Future Work

7.1 Conclusion

In this thesis, we have introduced a number of novel concepts and search tools for online digital libraries, which are summarized below.

Evaluating citation-based score measures of publications. In chapter 2 of this thesis, we compare and evaluate several publication score functions, including PageRank [BP98], Authorities [KL98] and citation-count scores [CS03]. We observe the separability problem with all of these functions, which is defined as the scoring functions producing scores that do not distribute well over a given scale, e.g., [0, 1]. Instead, distributions of the existing publication score functions are highly skewed, and decay very fast [RS04], resulting in a much less useful comparative publication assessment capability for users. This lack of separability is caused by the “rich gets richer” phenomena [RS04, XG03], i.e., a very small number of publications with relatively high numbers of in-citations have even higher chances of receiving new citations. Yet, these scoring functions are still not very accurate, probably due to topic diffusion in search outputs [TH02]. We also demonstrate that separability and accuracy of PageRank-based paper scores can be enhanced by (a) weighing citations, (b) weighing the "Future Citation Probabilities" represented by the E parameter of PageRank, and (c) postprocessing PageRank raw scores by (i) nonlinear normalization, (ii) linear normalization by a properly selected percentile score, or (iii) combining PageRank-based paper scores and publication venue scores. However, enhancing separability using these techniques does not enhance the accuracy of PageRank.
**Improved publication scores via research-pyramids.** In chapter 3 of this thesis, we observe that (a) the complete publication citation graph (of AnthP) is highly clustered, (b) each cluster of the complete publication set has a pyramid-like structure in terms of the citation graph of the cluster, and (c) each cluster represents a highly specific research topic. These three observations validate the research pyramid model proposed by Aya et. al. [AYA].

We also find that topic similarities decay over both citation ages and citation paths. We use two topic similarity decay curves to guide the research-pyramid construction, and propose and validate two algorithms to identify research pyramid structures in citation graphs.

Within research-pyramid citation graphs, we notice that the average number of in-citations per paper varies, pointing to the importance of comparative publication scores within research pyramids. We then observe that normalizing publication scores within research-pyramids produces accurate and nearly normally distributed scores of publications.

**Evaluating publication similarity measures** (second part of chapter 2). By evaluating “multiple levels” of paper similarities based on bibliographic-coupling, co-citation and author-coupling, we make the following observations: (a) Similarity value distribution curves are similar within the same group of similarity measures, i.e., bibliographic-coupling-based, co-citation-based, and author-coupling-based measures, (b) Citation-based and author-coupling-based similarity measures are more separable than bibliographic-coupling based measures, (c) Citation-based and author-coupling-based similarity measures are all highly correlated. This phenomena is due to the citation and coauthorship behavior in the literature [MN04], (d) Text-based similarity measures show low overlapping with citation-based and with author-coupling-based measures.
Therefore, providing two sets of similarity scores, one text-based and another based on citation and/or author-coupling, may prove to be a useful practice.

**Content-driven search keyword suggester for OLDLs.** We propose and evaluate a “content-driven search keyword suggester” for keyword-based search in literature digital libraries. Our search keyword suggestion approach is based on an a priori analysis of the publication collection in the digital library at hand. We find that our proposed search keyword suggester (i) has an excellent locality of access because it is based on an enhanced notion of context (our view of context goes beyond the set of documents where the keywords cover all possibly relevant research topics), (ii) is more scalable and produces significantly less numbers of pre-computed union of words (query refinements, and, thus leads to smaller index sizes), and (iii) produces higher quality and more user-friendly suggestions.

**Popularity Growth and Decay of Publications.** In chapter 5 of this thesis, we propose new definitions for popularity growth and decay for publications by coupling Cho et. al.’s model of popularity growth with our probabilistic publication citation behavior model, which we refer to as the publication quality with aging factor. In detail, we (i) experimentally validate the popularity of publications change over time and follow the logistic growth equation as proposed by Cho et. al. [Cho05], (ii) propose an empirical model for one aspect of researchers’ citation behavior in technology-driven fields of study such as computer science (this model captures researchers’ tendency not to cite old publications), and (iii) extend the popularity growth model developed by Cho et. al. [Cho05] to capture publication popularity decay. Our major findings are as follows: (a) empirically, the probability of citing any publication conforms to the Weibull distribution [MW] over the age of that publication. However, the shape and scale parameters of the distribution changes with the quality of publication venues, (b) we show that the derivative of
the popularity growth function accurately represents (i.e., directly proportional to) the temporal publication popularity at any time, (c) we observe that our definition of *publication quality with aging factor* matches the derivative of the popularity growth curve. This provides an analytical foundation for our growth and decay model of publication popularity.

**7.2 Future Work**

**7.2.1 Advanced Search Interface via Research Pyramids**

As future work, we plan to work on the problem of automatically annotating research pyramids with keywords representing fine-grained research topics. Also, by using the identified research pyramids, we may work on visualization, namely, building a hierarchical structure that places research pyramids into a hierarchical structure. Using RP annotations and the hierarchical structure of RPs, building an advanced query interface that involves pruned searches becomes possible.

**7.2.2 Accurate Identification of Research Pyramids**

The two RP-identification algorithms proposed in chapter 3 are very basic, and form the first attempts. As future work, one may find more accurate techniques to identify cornerstone publications within research pyramids. Also, more accurate techniques to identify members of each RP need to be developed.

**7.2.3 Enhanced Content Driven Search-keyword suggester**

As future work, we will explore more refinements through the "cites" relationship. This is expected to be of help towards an exhaustive coverage of related, yet not co-occurring, search keywords. This way, we enhance the *linguistic analysis* with *citation analysis* to suggest search
keyword refinements, that is, by considering the “cites” relationship between distinct tokens (i.e., those that are not co-occurring in common titles). Such a relationship occurs when two tokens appear in the titles of two different publications that cite one another. This relationship may be weighed by the length of the citation path observed between two tokens. An individual citation relationship between publications may also be refined by citation age, and may also be “classified” as in [AYA, SEC07].

7.2.4 Hybrid Search-Keyword suggester

One possible extension to our search-keyword suggester is to build Hybrid search-keyword suggester (i.e. content-driven and search-history based) as follows. Initially, when the LDL is put to service, users’ search-history is not available, and thus, we suggest keywords extracted from document contents. In time, we track users’ search terms and save search terms/phrase that appear frequently in user’s queries, i.e., search-history. Hybrid search keyword suggesters, thus, combine the advantages of both approaches. That is; (i) search-history keywords are accurate, because users are less likely to issue search commands with incomplete or misspelled search, (ii) search-history keywords are based on well-characterized queries. For a given search, a user may have difficulty formulating his/her queries as keywords and may repetitively modify the search keywords [SB08], (iii) provide users with personalized search-keyword suggestions, which is an advantage of Google’s search-history based suggester.

7.2.5 Publication-venue Specific User Citation Behavior

As future work, we are also working on identifying the correlation between the impact of the publication venue on user’s citation behavior and publications that appear in prestigious conferences. More specifically, we are attempting to model users’ citation behavior for
prestigious publication venues. Our hypothesis is that, by understanding users’ citation behavior, one can provide users of online digital libraries with higher quality of services.
<table>
<thead>
<tr>
<th>Key</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA03</td>
<td>Al-Hamdani, A., <em>Querying web resources with metadata in a database</em>,</td>
</tr>
<tr>
<td>ACM</td>
<td>ACM Digital Library, <a href="http://portal.acm.org/dl.cfm">http://portal.acm.org/dl.cfm</a></td>
</tr>
<tr>
<td>ANTH</td>
<td>ACM SIGMOD Anthology, <a href="http://www.acm.org/sigmod/dblp/db/anthology.html">http://www.acm.org/sigmod/dblp/db/anthology.html</a></td>
</tr>
<tr>
<td>AT07</td>
<td>Adam Troy, Vertical Digital Libraries: A New Breed of Domain Specific</td>
</tr>
<tr>
<td>AU04</td>
<td>Aussenac-Gilles, N., Mothe, J.: <em>Ontologies as Background Knowledge to</em></td>
</tr>
<tr>
<td>AYA</td>
<td>S. Aya, C. Lagoze, and T. Joachims, <em>Citation Classification and its</em></td>
</tr>
<tr>
<td></td>
<td>Applications*, International Conference on Knowledge Management, 2005.</td>
</tr>
<tr>
<td></td>
<td><em>Comparing apples and oranges: normalized pagerank for evolving</em></td>
</tr>
<tr>
<td></td>
<td><em>graphs</em>. WWW ’07.</td>
</tr>
<tr>
<td>BP98</td>
<td>Brin, S., Page, L., <em>The anatomy of a large-scale hypertextual web search</em></td>
</tr>
<tr>
<td></td>
<td><em>engine, Computer Networks and ISDN Systems</em>, 1998.</td>
</tr>
</tbody>
</table>
Wesley, Reading, Massachusetts, 1999.


[CBSKS] **elGiza Content-Based Search-keyword Suggester**, http://sulieman.case.edu/giza/cbsks_xkb_search/


[CS] **CiteSeer**, www.citeseer.com


[CSMCA] **CiteSeer List of Most cited articles in Computer Science**, http://citeseer.ist.psu.edu/articles.html


[FP06] Fei Pan, **Comparative Evaluation of Publication Characteristics in Computer Science and Life Sciences**, MS Thesis, EECS, Case Western
Reserve University, 2006.


[HI06] Holger Bast, Ingmar Weber: Type less, find more: fast autocompletion search with a succinct index. SIGIR 2006: 364-371


iProspect Inc., iProspect Search Engine User Behavior study, iProspect 2006.


G. Jeh and J. Widom, SimRank: A measure of structural-context similarity, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002.


Kunihiko Sadakane, Hiroshi Imai, Text Retrieval by Using k-word
**Proximity Search.** dante, p. 183, 1999 International Symposium on Database Applications in Non-Traditional Environments (DANTE'99), 1999


[KSMG06] Klaus Berberich, Srikanta Bedathur, Michalis Vazirgiannis, Gerhard Weikum, *BuzzRank … and the trend is your friend*, WWW 2006


[RE07] Ratprasartporn, N., Po, J., Cakmak, A., Bani-Ahmad, S., Ozsoyoglu, Evaluating utility of different ranking functions in context-based
environment, DBRank Workshop, Istanbul, Turkey, April 2007.


[RS04] Redner, S., Citation statistics from more than a century of physical review, physics 0407137, 2004


[SD] ScienceDirect, www.sciencedirect.info


[STP07] Sulieman Bani Ahmad, Gultekin Ozsoyoglu, Content-based Search-


