Rheinische Friedrich-Wilhelms-Universität Bonn
Institut für Informatik III

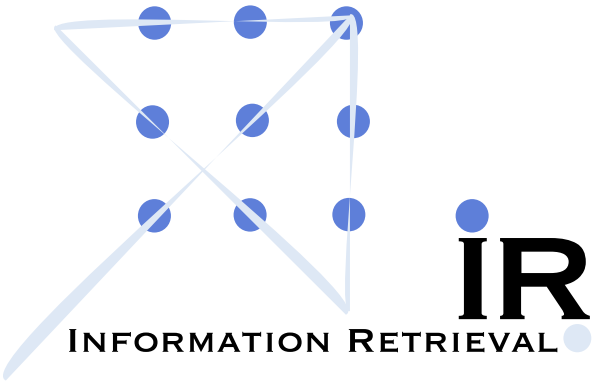
Diploma Thesis in Computer Science

Social Information Retrieval

Sebastian Marius Kirsch
kirschs@informatik.uni-bonn.de

Advisor: Prof. Dr. Armin B. Cremers

2nd November 2005



Diplomarbeit nach der Diplomprüfungsordnung für den Studiengang Informatik an der Rheinischen Friedrich-Wilhelms-Universität Bonn vom 14. März 2003.

Abstract

In this diploma thesis, we research whether the inclusion of information about an information user's social environment and his position in the social network of his peers leads to an improval in search effectiveness.

Traditional information retrieval methods fail to address the fact that information production and consumption are social activities. We ameliorate this problem by extending the domain model of information retrieval to include social networks.

We describe two different techniques for information retrieval in such an enviroment. We evaluate these techniques in comparison to vector space retrieval.

Acknowledgements

I thank my advisor Prof. Dr. Armin B. Cremers for giving me the opportunity to work on this interesting subject, and for his support during the preparation of this thesis.

Thanks to Dipl.-Inform. Melanie Gnasa for advising me on the details of my thesis and for invaluable discussions and advice. Further thanks to Andreas Behrend, Julia Kuck, Patrick Lay, Stefan Lüttringhaus-Kappel and Oliver Speidel of the Institut für Informatik III.

I thank my parents for their support, and my partner for his appreciation and encouragement.

I am greatly indebted to the authors of various open-source software packages that eased the implementation of a prototype system. Specifically, I would like to thank the authors of the Lucene library, the JUNG library and the Colt package.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The goal of **information retrieval** (IR) is facilitating a user's access to information that is relevant to his information needs. According to Baeza-Yates and Ribeiro-Neto (1999), an information retrieval system 'should provide the user with easy access to the information in which he is interested.' Earlier definitions took a narrower and more technical view on the purpose of a retrieval system, for example Lancaster (1968): 'An information retrieval system does not inform (i. e. change the knowledge of) the user on the subject of his inquiry. It merely informs on the existence (or non-existence) and whereabouts of documents relating to his request.', or Frakes and Baeza-Yates (1992): 'An IR system matches user *queries* – formal statements of information needs – to documents stored in a database.' Manber (1992) traces the history of information retrieval back to the first Sumerian literary catalogues, about four thousand years ago.

An information retrieval system must first determine the exact nature of the user's information needs, then select a subset of documents that help him satisfy his information need, and finally rank the selected documents according to which documents are most likely to provide a satisfactory answer.

1.1 Information Retrieval and the Social Realm

Wilson (1981) notes that both the user's information needs and his strategies for satisfying them are influenced by the socio-cultural environment, since they arise in social situations. Wenger (1996) introduced the idea of the 'community of practice': the notion that person can satisfy his information needs more efficiently if he is embedded in a community of practitioners with similar interests and problems. Indeed, before the advent of modern information retrieval systems, most information needs were satisfied by social means: by asking friends and acquaintances, by going to the library and asking the librarian for help, or by enquiring at specialized agencies.

Although the amount of information available in automated retrieval systems is far greater than can be acquired from other people, information that comes from immediate contacts is usually preferable to information obtained from anonymous sources: Since the provider is known, it is easier to assess the quality of the information. Here, quality has several different aspects; the first and foremost is factual accuracy. But there are

also secondary aspects, for example the provider's subjective evaluation, the ability to further discuss the topic with the provider, and obtain references to other relevant pieces of information. Only when one's immediate contacts are not able to satisfy the information need or more in-depth information about a topic is required, one turns to secondary sources – equipped with the information acquired by asking within the community.

Information retrieval meets the social realm at another, more subtle point: Information is also produced in social situations. Few authors work in a social vacuum. Participation in the community and active exchange with like-minded persons fosters information production and improves the quality of the work.

Granovetter (1973) notes that 'weak ties' – ties between acquaintances rather than between close friends or family – are particularly important for information dissemination and diffusion: Weak ties allow information to spread from one closely-knit community to another. Individuals with many weak ties – 'hubs' in the social network – are important for the adoption of new ideas, since their authority is accepted by a large number of immediate acquaintances.

We conclude that social networks are an important factor for finding and spreading information, and that an individual's position in the social network of his peers is indicative of his authority and influence. Accordingly, we define **social information retrieval** as the incorporation of information about social networks and relationships into the information retrieval process.

1.2 The Internet: A Social Medium?

With the increasing use of electronic communications media, viz. the Internet, social ties and the structure of the social network become tractable. This section outlines some examples of online networks where data about social ties between users is available, in addition to similarity data or references between documents and information about authorship. In such a setting, incorporating social information into the retrieval process is an obvious next step: Since both information usage and information production occur in social environments, both are influenced by the social network of the user and the author. Knowledge of these networks affects all parts of the information retrieval problem.

1.2.1 Scientific Community

Social network analysis in the scientific community has a long tradition. Through the use of bibliometric measures such as co-citation coupling and bibliographic coupling, the network structure of scientific publications and the publications they cite can be assessed.

A famous anecdotal application of network analysis in the natural sciences is a person's Erdős number[1]: The minimum length of a path in the co-authorship network between the Hungarian mathematician Paul Erdős and a given person.

Network analysis in the scientific community is usually conducted on the basis of publications in well-known journals or conference proceedings, as well as the cited publications. These documents usually do not capture the full extent of social relationships between authors, since much communication occurs via secondary channels, such as email. The observable content is of very high quality.

A number of databases of scientific publications exist, for example MathSciNet[2], PubMed[3] and CiteSeer[4]. Some databases, most notably CiteSeer, support download of records via the Open Archive Initiative Protocol for Metadata Harvesting[5], making social retrieval on scientific publications possible.

A corpus with data from 25 years of SIGIR proceedings, stemming from work on (Smeaton et al., 2002) and enhanced locally, is used for evaluation in subsequent chapters.

1.2.2 Wikis

Wikis are a form of collaborative authoring environment that is characterized by the fact that every user can add, edit, and delete content at will. The first wiki was WikiWikiWeb[6], launched by Ward Cunningham in 1995 as a supplement to the Portland Pattern Repository, a web site about software design patterns. A number of software packages and similar projects followed; the largest wiki is purported to be Wikipedia[7], an online encyclopedia that employs the wiki principles.

Wikis usually have a flat structure, with one designated entry page that links to other pages; some use fixed number of categories. Most wikis keep a revision history that allows changes to be linked to individual users. Direct interaction between users usually occurs on the user's home page.

The quality of published content varies wildly; some wikis contain nothing more than a few quickly written ideas, others, like Wikipedia, aim for publication-quality content.

1.2.3 Blogs

Weblogs or 'blogs' are an internet phenomenon originating in the late 1990s: Websites that continually publish new articles on their front page, written by one individual or

[1] http://www.oakland.edu/enp/, last visit on 2005/03/08.
[2] http://www.ams.org/mathscinet, last visit on 2005/03/08.
[3] http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=PubMed, last visit on 2005/03/08.
[4] http://citeseer.ist.psu.edu/, last visit on 2005/03/08.
[5] http://www.openarchives.org/OAI/openarchivesprotocol.html, last visit on 2005/05/02.
[6] http://c2.com/cgi/wiki?WikiWikiWeb, last visit on 2005/03/08.
[7] http://www.wikipedia.org/, last visit on 2005/08/03.

a group of people. Blog entries can be tied to their author; linking between entries is supported in the form of comments or so-called 'trackback links', in which the author of another blog refers in his entry to the original entry.

Blogs can take many forms: personal blogs usually form a sort of diary of the owner's thoughts and interests. Topical blogs are usually edited by several people and publish information about a specific topic. Corporate blogs may give the executives and other employees a platform for publishing news articles. A number of service provides exist on the internet that allow one to create a blog free of charge; examples are BlogSpot[8] (now owned by Google) and LiveJournal[9].

Another typical feature is the so-called blogroll: A list of other blogs the author reads regularly. This may be used to determine social links between authors, but it is not universally adopted.

1.2.4 Messenging Systems

There are a number of messenging systems that are sufficiently similar to each other to be grouped under one heading; examples are email mailing lists, Usenet, and web forums. These systems are among the oldest collaborative electronic mediums; however, as articles are often written 'off the cuff' and cannot be revised, they are often lacking in quality. Mailing list archives can be a valuable repository of knowledge, but separating the wheat from the chaff is notoriously difficult.

Messenging systems are usually characterized by a tree structure of links between individual documents. A further speciality is that the polarity of the link structure is unclear: A follow-up article is often not a sign of support, but a sign of disagreement or a sign that the original article is lacking information.

1.2.5 ISKODOR: Congenial Web Search

ISKODOR[10] is an experimental system developed at the University of Bonn. The stated goal of the project is the implementation of 'congenial web search' (Gnasa et al., 2004) – meaning a user-centred approach where search quality is constantly evaluated through explicit feedback.

The functional prototype of ISKODOR employs a peer-to-peer architecture in order to share search results with other users. Thus, a single point of failure or bottlenecks are avoided. The user's faith in the service is strenghened, as he himself controls which information is stored and disseminated about him.

ISKODOR implements personalized ranking matrices; collaborative information retrieval is implemented in the form of peer groups, which are used to limit the scope of a search (Gnasa et al., 2003).

[8] http://www.blogspot.com/, last visit on 2005/03/09.
[9] http://www.livejournal.com/, last visit on 2005/03/09.
[10] ISKODOR is an acronym for 'Is Sharing Knowledge Online a Dream Or Reality?'

An ISKODOR peer can keep track of the quality of the results provided by its peers and re-rank results according to the peer that supplied it. This 'peer relevance' judgement leads to a network of trusted peers that produce the most relevant results.

Social search techniques can be applied in this network of trusted peers, to provide better search results and find peers that are well versed in a specific topic. Thus, social information retrieval can be used to improve web search effectiveness.

1.2.6 Semantic Web

Semantic Web (Berners-Lee et al., 2001) is a loosely-defined term for exchanging information on the world wide web, characterized by the information being in a format with precise semantics. In the current incarnation, as developed by the Semantic Web working group[11] of the World Wide Web Consortium[12], it is built on XML as the underlying markup language and data exchange format, and RDF and OWL as knowledge representation languages. Standardized ontologies, expressed in the knowledge representation languages, allow the description of entities and their relations.

The Semantic Web allows for the inclusion of precise information about documents on the web, their authors, and relations between individuals. The Dublin Core Metadata Element Set[13] contains a set of attributes for documents, such as authorship, title or publication date. Similarly, the Friend of a Friend (FOAF) project[14] published a standard for machine-readable information about individuals, their relations to others, and their activities. Together, these standards allow for the automated extraction of authorship information and social information. OpenSearch[15] is a standard for describing search engines and their query formats, and for returning result lists in a machine-readable format.

1.3 Paving the Way: Personalized and Collaborative Information Retrieval

In conventional information retrieval systems, all of the user's information needs are embodied in a **query**, a short string of key words or a question. Further indicators of the user's general information needs are not taken into account, such as his previous searches or his web sites of interest. Indeed, a query with one or two keywords is much too short to contain a complete picture of a user's needs. A search engine is therefore susceptible to a form of tyranny of the majority: It can only display those sites that

[11] http://www.w3.org/2001/sw/, last visit on 2005/08/30.
[12] http://www.w3.org, last visit on 2005/08/30.
[13] http://dublincore.org/, last visit on 2005/08/30.
[14] http://www.foaf-project.org/, last visit on 2005/08/30.
[15] http://opensearch.a9.com/, last visit on 2005/08/30.

will be relevant to the majority of its users, but not to the actual user who submitted a query.

Personalization seeks to solve this problem by keeping a record of the user's previous activity and using it to attune the results to his profile. Implementations of personalized search exist, but are not yet in widespread use; examples are Amazon's a9.com[16] and Eurekster[17], which are implemented as a central service, or SearchPad (Bharat, 2000), a client application.

A collaborative element can be added by comparing and combining the profiles of different users. This approach is popular in information filtering systems such as the GroupLens system (Konstan et al., 1997) for filtering Usenet posts. It has also been used in information retrieval systems, for example in the aforementioned Eurekster system, or the experimental I-Spy[18] search engine (Freyne and Smyth, 2004).

Personalization strategies and collaborative retrieval attack the problem of determining a user's information needs from different angles. Personalization aims to infer a more detailed view of the information needs based on past usage, whereas collaborative ranking acknowledges that the information seeker is part of a community of like-minded individuals.

1.4 Social Retrieval and the World Wide Web

Much if not most of the current research in information retrieval is focused on searching the World Wide Web, a topic that at the same time presents inherent obstacles (due to its size and its lack of structure) and great promises (due to the amount of information that is publicly available.) Extracting the most relevant pages from 8 billion web pages[19] is a daunting task, especially if all information about the desired results is condensed to one or two keywords. (Silverstein et al. (1999) give an average length of 2.35 keywords for their analysis of AltaVista query logs.) During the evolution of internet search engines, it quickly became apparent that this problem cannot be solved by relying only on automatic evaluation of a web page's content, but needs some sort of human assessment of a page's relevance.

Early attempts to build a manual index of web pages, selected by human editors, (so-called 'web catalogues') were largely unsuccessful – because of the sheer size of the web and the limited manpower of the companies. Most major web portals still provide some kind of directory, for example the Google Directory[20] or the Yahoo! Directory[21], or use data from the Open Directory Project[22]. However, the focus for navigating the

[16] http://www.a9.com, last visit on 2005/04/11.
[17] http://eurekster.com, last visit on 2005/04/11.
[18] http://ispy.ucd.ie/, last visit on 2005/04/15.
[19] According to their front page, the Google search engine indexes 8,058,044,651 as of 2005/04/14.
[20] http://www.google.com/dirhp, last visit on 2005/05/08.
[21] http://dir.yahoo.com/, last visit on 2005/05/08.
[22] http://www.dmoz.org/, last visit on 2005/05/08.

web has been on automated information retrieval, not manual indexes, for several years.

Recent efforts in collaborative projects have shown that it is possible to garner a large, active user community, in the tens of thousands or even millions of users, within a few months. Projects such as Wikipedia[23] show that large undertakings purely on the basis of volunteer labour are possible. In this sense, PageRank (Page et al., 1999) is also a collaborative effort in information retrieval and ranking, since it uses link information published on millions of web pages.

These examples motivate a vision for the future of web search that is not dominated by centralistic efforts of single companies, providing us with results derived from a global view of the web. One may envision a service that provides each user with results that are tailored to his individual information needs, and that derives its results by collaborating with other users, sharing information and relevance assessments. Such a tool would be an ideal application for social information retrieval, since it combines the social network with the wealth of information available on the World Wide Web.

1.5 Research Contribution

This thesis defines the social information retrieval task and describes its domain. A formalization on the basis of associative networks is provided, as well as search procedures for these networks. An evaluation compares the described methods to conventional information retrieval methods.

1.6 Outline of the Thesis

The remaining part of this diploma thesis is structured as follows:

Chapter 2 introduces typographical conventions and key terminology.

Chapter 3 describes the state of the art in information retrieval and related fields.

Chapter 4 lists related work.

Chapter 5 defines social information retrieval in terms of a domain model and requirements for a system implementing this model.

Chapter 6 describes two algorithms implemented on the domain model which realize social IR.

Chapter 7 evaluates the described algorithms.

Chapter 8 contains notes on the implementation of the prototype system used for evaluation of the algorithms.

[23] http://www.wikipedia.org/, last visit on 2005/04/08.

Chapter 9 concludes the thesis by discussing its impact and limitations of the described methods, and listing future work.

Chapter 2

Notation and Terminology

This chapter introduces notation and typographical conventions used in later chapters. It defines key terminology for describing graphs.

2.1 Notation

Vectors are denoted by bold lowercase letters: $\mathbf{v} \in \mathbb{R}^n$ is a vector in the vector space $\mathbb{R}^n$. The components of a vector are denoted by a subscript: $\mathbf{v} = (v_1, \ldots, v_n)$. Matrices are denoted by uppercase letters: $M \in M_{m\times n}(\mathbb{R})$ is a matrix with m rows and n columns, where the components are real numbers. $(M)_{ij}$ denoted the component in row i and column j of matrix M. $\mathbf{1}$ denotes a matrix of appropriate dimensions where every component is equal to 1: $(\mathbf{1})_{ij} \equiv 1$.

For variables which change over time, the time is denoted by a superscript: x^t is the value of x at time t.

For a set of values $x_1, \ldots, x_l$, the average of the values is $\overline{x}$.

2.2 Terminology

A **graph** $G = (V, E)$ consists of a finite set of **nodes** (or **vertices**) V and a set of **edges** E connecting the nodes. An edge is either **directed**, in which case it is a tuple $(v, v') \in V \times V$, or **undirected**, in which case it is a set $\{v, v'\} \in 2^V$. A graph which has only directed edges is called a **directed graph**, a graph with only undirected edges is an **undirected graph**.

The **underlying undirected graph** of a directed graph is a graph $G' = (V, E')$ with

$$E' = \left\{\{v, v'\} \mid (v, v') \in E \vee (v', v) \in E\right\}$$

The **degree** δ of a node in an undirected graph is the number of edges containing the node:

$$\delta(v) = |\{e \in E \mid v \in e\}|$$

For directed graphs, we distinguish between the indegree δ^-, which is the number of edges terminating in a node, and the outdegree δ^+, the number of edges emanating from

a node:

$$\delta^-(v) = \left|\left\{(v', v'') \in E \mid v'' = v\right\}\right|$$
$$\delta^+(v) = \left|\left\{(v', v'') \in E \mid v' = v\right\}\right|$$

A **path** between two nodes v and v' is a sequence of nodes $v_0, \ldots, v_k$ with $v_0 = v$ and $v_k = v'$, such that $(v_i, v_{i+1}) \in E$ (in the directed case) respectively $\{v_i, v_{i+1}\} \in E$ (in the undirected case) for $0 \leq i < k$. The **length** of the path is k; there is a trivial path from v to v with length 0 for every node. The **distance** of two nodes v and v' is the minimal length of a path connecting them.

A graph $G' = (V', E')$ is a **subgraph** of $G = (V, E)$, if $V' \subseteq V$ and $E' \subseteq E$. The **induced subgraph** $G[V']$ is the graph

$$G[V'] = (V', E \cap (V' \times V'))$$

respectively

$$G[V'] = (V', E \cap 2^{V'})$$

An undirected graph G is **connected** if there exists a path in G from v to v' for every pair of nodes in V. A **connected component** of an undirected graph is a maximal subgraph $G' = (V', E')$ such that G' is connected.

A directed graph G is strongly connected if there exists a path in G from v to v' and a path from v' to v for every pair of nodes in V. A **strongly connected component** of a directed graph is a maximal subgraph $G' = (V', E')$ such that G' is strongly connected. A subgraph $G' = (V', E')$ is a **weak component** of a directed graph if the underlying undirected graph of G' is a connected component of the underlying undirected graph of G.

A graph is **weighted** if there is a weight $c_e \in \mathbb{R}$ associated every edge $e \in E$.

For a graph with a set of nodes $V = \{v_1, \ldots, v_n\}$, we often write e_{ij} for the edge from v_i to v_j; likewise, c_{ij} is the weight associated with e_{ij}. The **adjacency matrix** is the matrix $A \in M_{|V|\times|V|}(\mathbb{R})$ with

$$(A)_{ij} = \begin{cases} 1 & \text{if } e_{ij} \in E \\ 0 & \text{otherwise} \end{cases}$$

For a weighted graph, the adjacency matrix is

$$(A)_{ij} = \begin{cases} c_{ij} & \text{if } e_{ij} \in E \\ 0 & \text{otherwise} \end{cases}$$

For an undirected graph, the adjacency matrix is symmetric.

Chapter 3

State of the Art

This thesis draws its techniques and inspiration from a number of different sources, and tries to acknowledge current and emerging trends in information retrieval and related fields.

This chapter contains an introduction to fundamental information retrieval techniques. It reviews other approaches to personalized and collaborative IR. Techniques from social network analysis for characterizing large graphs are described, which are used for comparing social networks with other networks in later chapters. Associative networks as a means of knowledge representation are discussed, as well as search techniques for such networks. Evaluation metrics describe the performance of an IR method and are used for comparison with other methods.

3.1 Information Retrieval Models

The domain of an information retrieval system is a set of **index items** D, typically a set of documents. Each index item $d \in D$ is represented by a set of **indexing features** $\{t_i, \ldots, t_j\} \subset T$; T is the set of all indexing features. Indexing features are typically index terms or keywords extracted from text documents. A weight function $\text{weight} : D \times T \to \mathbb{R}$ determines the weight of a feature T as regards an item d.

The user information needs are represented by a query q from a set of possible queries Q. For a query q, an information retrieval system produces a set of relevant documents $D_q \subseteq D$. A ranking function $\text{rank} : D_q \to \{1, \ldots, |D_q|\}$ defines an ordering among the relevant documents for a specific query.

There are several equivalent representations of the relation between terms and documents, as shown in figure 3.1. A document-term matrix is a matrix $M \in M_{|D|\times|T|}(\mathbb{R})$ with $|D|$ rows and $|T|$ columns, with $(M)_{ij} = 1$ if the document d_i contains the term t_j, and 0 otherwise. A term list enumerates for each document the terms contained in it. An inverted index lists for each term the documents containing it. The associative network view represents the document-term space as a bipartite graph $G = (D \uplus T, E)$, with nodes representing terms and documents. An edge e_{dt} exists between a document d and a term t if the document contains the term.

	t_1	t_2	t_3	t_4	t_5
d_1	0	0	1	1	0
d_2	0	1	0	1	0
d_3	1	1	0	0	1

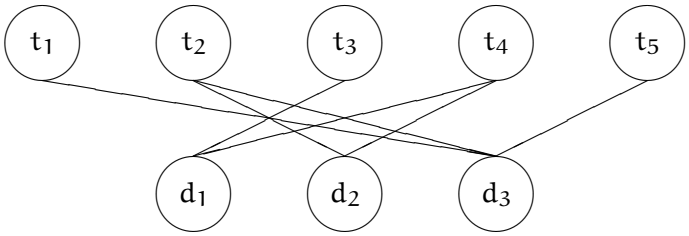
(a) document-term matrix

doc	terms
d_1	t_3, t_4
d_2	t_2, t_4
d_3	t_1, t_2, t_5

(b) term lists

term	doc.
t_1	d_3
t_2	d_2, d_3
t_3	d_1
t_4	d_1, d_2
t_5	d_3

(c) inverted index



(d) associative network

Figure 3.1: Different views of the document-term space (examples reproduced from Preece, 1981, page 11)

3.1.1 The Vector Space Model

In the **vector space model**, documents and queries are represented by vectors in the **term vector space** $\mathrm{TVS} = \mathbb{R}^{|T|}$. A document d is assigned a document vector $\mathbf{d} = (\mathrm{weight}(d, t_1), \ldots, \mathrm{weight}(d, t_{|T|})) \in \mathrm{TVS}$; query vectors are assigned to queries likewise. The weight function weight is defined to be non-negative.

Since documents and queries share the same representation, the rank of a document d as regards a query q is determined by the similarity between document and query $\mathrm{sim}(d, q)$. A popular similarity function is the cosine of the angle between the query vector and the document vector:

$$\mathrm{sim}(d, q) = \cos \angle(\mathbf{d}, \mathbf{q}) = \frac{\mathbf{d} \cdot \mathbf{q}}{\|\mathbf{d}\| \cdot \|\mathbf{q}\|}$$

Different weighting functions have been proposed for weighting of features in queries and documents, see for example (Salton and Buckley, 1988b). A popular choice is the $\mathrm{tf} \cdot \mathrm{idf}$ weighting scheme

$$\mathrm{weight}_{\mathrm{tf} \cdot \mathrm{idf}}(d, t) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t)$$

where the **term frequency** $\mathrm{tf}(t, d)$ is the number of times the indexing feature t occurs in the document d. $\mathrm{idf}(t)$ is the **inverse document frequency**

$$\mathrm{idf}(t) = \frac{1}{\log(\mathrm{df}(t)) + 1},$$

where $\mathrm{df}(t)$ is the number of documents the feature t occurs in. The $\mathrm{tf} \cdot \mathrm{idf}$ scheme expresses that idea that an index feature is more important for characterizing a document if it occurs often in the document, but seldom in the document collection.

The vector space model with $\mathrm{tf} \cdot \mathrm{idf}$ weighting or one of its variations is currently the most popular model in commercial and other information retrieval applications. It provides implicit ranking through the similarity measure, it is reasonably fast to implement, and it provides support for partial matches. Despite its simple design, it exhibits a consistently high performance.

Since documents and queries are represented as vectors in the same vector space, this model lends itself easily to techniques for relevance feedback and query expansion.

A recent development in vector space retrieval is **latent semantic indexing**, or LSI (Deerwester et al., 1990). Latent semantic indexing aims to compress the term vector space into a lower-dimensional space by means of singular value decomposition of the document-term matrix. By projecting the term vector space onto a lower-dimensional space, associations between terms become apparent. LSI is designed to handle the synonymy problem: Authors use different words for the same concept, but searchers usually use just one term in the query formulation. LSI retrieves documents relevant to the query concept even if the query keywords are not present in the document, thus improving the number of relevant documents found.

3.1.2 Associative Retrieval

Associative retrieval treats documents and terms as nodes in an associative network. The network can contain document-term, term-term and document-document associations. This model is called 'neural network model' by Baeza-Yates and Ribeiro-Neto (1999). The associative network is usually searched by the means of techniques from semantic networks, namely spreading activation search. (Search in associative networks is described in detail in section 3.6.)

Term-term associations can be determined by statistical measures, for example term co-occurrence; document-document associations can also be computed in terms of the overlap of their vocabulary. Salton (1963) suggested using bibliographic coupling (the number of citations shared by two documents) as an association measure for documents. Links in hypertext environments can also be used as associations in an associative network (Crestani and Lee, 2000).

Preece (1981) conducted an extensive study and concluded that several other information retrieval models (for example the vector space model, the boolean model, and relevance feedback mechanisms) can be simulated with associative retrieval techniques.

Salton and Buckley (1988a) evaluated a simple associative retrieval model in an experimental setting and concluded that its performance was similar to vector space methods.

Associative Retrieval is an attractive model since it allows one to model associations between nodes in a natural way. Integration of dissimilar node types and several types

of associations between nodes is easily achieved; for an example see (Pirolli et al., 1996).

3.1.3 Hypertext Retrieval

Information retrieval in a hyperlinked environment and especially in the world wide web (WWW) presents challenges not met by conventional information retrieval methods. The WWW (Berners-Lee et al., 1994) is an extremely large, highly distributed collection of semi-structured hypertext. One of the biggest challenges in web retrieval is not finding pages that meet the user's information needs – for many queries, there will be millions of pages that contain all or some of the query words. The challenge is finding pages of high quality and ranking them accordingly. Kleinberg (1999) calls this the 'abundance problem' of the WWW.

Hyperlinked environments are usually described as a graph $G = (D, E)$. An edge exists from document d to document d' if d contains a hyperlink pointing to d'. In the case of the world wide web, we call this graph the **web graph**.

Most algorithms for web retrieval analyse the links between individual web pages. The simplest form of link analysis measures the popularity of a page by the number of links pointing to it. More advanced algorithms rely on spectral properties of the adjacency matrix of the web graph or derived matrices. The idea of using linear algebra methods for measuring the importance of a document based on its references goes back to Pinski and Narin (1976).

One of the earliest algorithms for link analysis in the web is the HITS (*hypertext induced topic search*) algorithm (Kleinberg, 1999). HITS operates on a subgraph of the web that is focused on a particular topic; such a subgraph is usually produced by querying an existing web search engine on a specific topic and following the outbound links from the top 200 result pages. The algorithm then produces 'hub' and 'authority' scores for the focused subgraph: It assigns scores to the pages according to whether they are an authoritative source for the given topic (and linked to by many hub pages), or whether they are an information hub (and links to many authoritative pages.) The hub and authority scores can be seen as a rank one approximation of the web pages' bibliographic coupling matrix $AA^\top$ and co-citation coupling matrix $A^\top A$ (Flake et al., 2004).

A survey of algorithms for measuring the importance of a node in a network is found in (White and Smyth, 2003); the most popular algorithm, the PageRank algorithm, is described in detail in the next section.

3.2 Link Analysis with PageRank

PageRank (Page et al., 1999) is one of the most well-known algorithms for link analysis; it was popularized by its inclusion into the successful web search engine Google[1].

[1] http://www.google.com/corporate/tech.html, last visit on 2005/08/19.

The PageRank algorithm is usually formulated based on a random surfer model: A user starts on a random web page and follows one outlink of this page at random and repeats this process on every page he reaches. Assuming that the link graph consists of a single strongly connected component (ie. there is a path from every page to every other page), the random surfer will eventually visit every page in the web graph. One may consider this sequence of pages as a Markov chain and compute the stationary probability of the random surfer being on a given page at any time.

The stationary probability can be computed using an iterative process. For a directed graph $G = (V, E)$ with nodes $V = \{v_1, \ldots, v_n\}$, one assigns an initial probability at time $t = 0$ of $r_i^0 = \frac{1}{|V|}$ to every node. In every iteration, r_i is updated according to

$$r_i^{t+1} = \sum_{(v_j, v_i) \in E} \frac{c_{ji} r_j^t}{\sum_{(v_j, v_k) \in E} c_{jk}}$$

where c_{ij} is the weight of the edge from v_i to v_j, or 1 if the graph is unweighted. The iterative process stops when the probability vector $\mathbf{r} = (r_1, \ldots, r_n)$ converges.

For graphs that do not consist of a single strongly connected component, this calculation may lead to undesirable results, and may not converge. If the graph contains a **sink**, ie. a page with outdegree $\delta^+(v) = 0$, the stationary probability of the random surfer being on that page converges to 1, with the probability of being on any other page converging to 0. To ameliorate these effects, a dampening factor on the transitions of the underlying markov chain is introduced, in the form of a 'teleportation step': On every visited page, the random surfer 'teleports' to a random page with a probability of $0 \leq \epsilon \leq 1$, or chooses one of the outlinks with a probability $(1 - \epsilon)$. This step ensures that the random surfer has a finite probability of visiting every page, and that he does not get 'stuck' on a sink page. The teleportation step is carried out while updating the probability scores:

$$r_i^{t+1} = \frac{\epsilon}{|V|} + (1 - \epsilon) \sum_{(v_j, v_i) \in E} \frac{c_{ji} r_j^t}{\sum_{(v_j, v_k) \in E} c_{jk}} \tag{3.1}$$

ϵ is usually set to a value between 0.1 and 0.3.

The stationary probability may also be computed using linear algebra methods: Let A be the adjacency matrix of the web graph G. Let M be a row-normalized version of A, that is $(M)_{ij} = \frac{(A)_{ij}}{\sum_k (A)_{ik}}$. Then the PageRank vector $\mathbf{r}$ is the maximal eigenvector of

$$\left(\frac{\epsilon}{|V|} \mathbf{1} + (1 - \epsilon) M \right)^\top,$$

provided that G is ergodic (Flake et al., 2004). Reformulating equation (3.1) as a vector equation shows the kinship between PageRank computation and the power method for computing the dominant eigenvector of a matrix:

$$\mathbf{r}^{t+1} = \frac{\epsilon}{|V|} \mathbf{1} + (1 - \epsilon) M^\top \mathbf{r}^t$$

If G is not ergodic, $\mathbf{r}$ needs to be normalized after each iteration.

The **PageRank score** r_i is used for ranking web pages according to their overall popularity. This score may be used to boost popular pages in cases where there are many relevant documents for a query.

Evidence for the importance of PageRank in web retrieval is still scarce: According to Craswell and Hawking (2004), only 11 of 74 submitted runs at the TREC-2004 'Web' track used PageRank, and only one of the top systems used it. How to combine PageRank and query-specific relevance measures is also an unsolved problem. Zaragoza et al. (2004) reported the following method for their top-ranking system at TREC-2004: They normalized the PageRank scores, transformed them by

$$f(r_i) = \frac{w}{1 + e^{-\log(r_i)+b}}$$

and added this factor to the query-specific relevance score. w and b were determined empirically using queries from the TREC-2003 'Web' track; no indication is given in their report regarding the magnitude of these parameters.

3.3 Personalized and Collaborative Retrieval

Personalized and collaborative retrieval are two approaches for improving the performance and, indirectly, the satisfaction of the user. The central element of both strategies is a **user model** that keeps a log of past interactions with the system, and which is used to tailor the results of future interactions to the user.

One personalization strategy is the capture of search engine queries and the pages from the result list that were selected in response to the query. A search in this 'search history' provides access to pages that were previously determined to be of high quality, and as relevant to the query. This approach is implemented by several systems, for example ISKODOR (Ruhl, 2003), SearchPad (Bharat, 2000), Amazon's a9.com[2], Eurekster[3], and Google personalized search[4] (currently in beta stadium.)

More sophisticated attempts at personalization build profile of the user's interests. This profile is the used to either augment future queries, or to filter out unwanted results from the result set. This approach is followed by the OutRide system (Pitkow et al., 2002).

Inspired by collaborative filtering systems (Resnick et al., 1994), **collaborative ranking** uses implicit relevance data from previous queries. A system implementing the collaborative ranking approach is the I-SPY seach engine (Freyne and Smyth, 2004). I-SPY is implemented as a meta-search engine: it does not maintain its own index of web pages, but instead queries several underlying search engines for results, re-ranks the

[2] http://www.a9.com, last visit on 2005/04/11.
[3] http://eurekster.com, last visit on 2005/04/11.
[4] http://www.google.com/searchhistory/, last visit on 2005/07/02.

result lists and presents them to the user. I-SPY logs queries to the search engine, as well as which pages from the result list users select for further inspection. The number of hits on a page $p \in P$ for a query $q \in Q$ is stored in a hit matrix $H \in M_{|Q| \times |P|}(\mathbb{R})$. For previously-selected pages and queries, the relevance of a page p_k for a query q_l is determined by

$$\text{relevance}(p_k, q_l) = \frac{H_{lk}}{\sum_{i=1}^{|P|} H_{li}}$$

The output is a stratified result list, the first part containing previously-encountered pages, sorted according to their relevance score, and the second part containing results from the meta-search engine.

3.4 Statistical Network Analysis

Research in the statistical properties of naturally occuring networks, including social networks, indicates that many of them share several key characteristics. Well-researched examples for natural networks include collaboration networks between movie actors, internet autonomous systems, the web graph, the power grid of the United States of America, or the neural network of the roundworm Caenorhabditis elegans. The increased availability of data about natural networks has led to a number of publications studying their properties in the last decade.

The degree distribution of many natural networks seems to follow a **power-law distribution**: The probability of a vertex having a degree of k is $\Pr(\delta = k) \sim k^{-\gamma}$. Many naturally occurring networks follow such a degree distribution for varying values of γ. Commonly cited examples include the web graph (with $\gamma \approx 2.1$), the power grid of the western United States (with $\gamma \approx 4$), and the social network of movie actors (with $\gamma \approx 2.3$). The **fat tail** of the power-law distribution entails that a small number of nodes with a very high degree provide connectivity for the bulk of the network.

Barabási and Albert (1999) conjecture that this degree distribution is a result of the network growing over time, and the fact that new nodes in the network connect preferentially to nodes with a high degree. The latter phenomenon is called **preferential attachment** in the literature. In difference, the degree distributions of networks where edges are added randomly between nodes follow a Poisson distribution.

The **average shortest path length** is the average length of the shortest path between two nodes in the network. For networks that are not connected (or strongly connected in the directed case), it makes sense to only examine the largest connected (or strongly connected) component. For a random graph with n nodes and k edges per node, the expected average path length is $\bar{l} \approx \frac{\ln n}{\ln k}$ (Watts and Strogatz, 1998).

The **clustering coefficient** is defined by Newman and Park (2003) as

$$C = \frac{3 \times \text{number of triangles on the graph}}{\text{number of connected triples of vertices}}$$

(Watts and Strogatz (1998) use a slightly different formuation.) Here, a 'connected triple' is a node that is connected directly to two other nodes. The clustering coefficient is the probability, averaged over the network, that two neighbours of a node will also be neighbours of each other. For a random network, the expected clustering coefficient is

$$C = \frac{(\overline{k^2} - \overline{k})^2}{n\overline{k}^3}$$

First evidence that social networks have a very low average shortest path length was presented by Milgram (1967). In this experiment, participants were asked to send a letter to a specific recipient, but only by passing them on in person to an immediate acquaintance. Similar experiments were conducted later, for example by Dodds et al. (2003). The resulting chains of acquaintance were surprisingly low, with typical chain lengths between five and seven. The fact that two friends of a person are more likely to be friends of each other – leading to a high clustering coefficient – was predicted by Granovetter (1973); large-scale investigations of social networks (for example by Newman, 2001) confirmed this claim.

Watts and Strogatz (1998) coined the term 'small-world network' for networks that exhibit a high clustering coefficient while at the same time retaining a small average shortest path length between two nodes. They showed that a small amount of randomness introduced into a regular network (with a high average path length and high clustering coefficient) will suffice to drastically lower the average path length, while affecting the clustering coefficient hardly at all. Newman and Park (2003) conjecture that while a high degree of clustering is a natural state for small networks, large social networks exhibit a far higher degree of clustering than can be explained by the random model.

The **degree correlation** is the correlation between the degree of neighbouring nodes in a social network. Let $Pr(\delta)$ be the degree distribution of a network, that is, $Pr(\delta = k)$ is the probability of a node v having degree k. For an edge e connecting nodes v_i and v_j the **excess degree** δ^e of the nodes connected by e is one less than their degree $\delta(v_i)$ resp. $\delta(v_j)$. The normalized distribution of the excess degree is

$$Pr(\delta^e = k) = \frac{(k+1)Pr(\delta = k+1)}{\sum_k kPr(\delta = k)}$$

The joint distribution $Pr(\delta^e = j, \delta^e = k)$ is the probability that a randomly chosen edge connectes two nodes with excess degree j and k. If the excess degrees of neighbouring nodes are uncorrelated, then $Pr(\delta^e = j, \delta^e = k) = Pr(\delta^e = j)Pr(\delta^e = k)$; this is the null model.

The degree correlation in comparison to the null model is

$$r = \frac{1}{\sigma^2} \sum_{j,k} jk(Pr(\delta^e = j, \delta^e = k) - Pr(\delta^e = j)Pr(\delta^e = k)),$$

where $\sigma^2 = \sum_k k^2 Pr(\delta^e = k) - (\sum_k k Pr(\delta^e = k))^2$ is the variance of $Pr(\delta^e)$.

Newman and Park (2003) note that a number of examined social networks have been found to have a positive degree correlation, ie. nodes with a high degree tend to be connected to other nodes with a high degree. In difference to this, non-social small-world networks usually exhibit a negative degree correlation: nodes with a high degree are usually connected to nodes with low degree. Examples for this phenomenon are neural networks, food webs, peer-to-peer networks, or the internet. In the case of the internet, a comparatively small number of primary 'hubs' distribute traffic to other autonomous systems. One may conjecture that in communication networks, negative degree correlation is a matter of economy: A highly-connected node requires a large investment and high maintenance costs, but can provide connectivity to a disproportionately large number of poorly connected nodes. In social networks, the number of social relations maintained by a person is a matter of personality: A introverted person has fewer social relations than a highly social person. Since individuals tend to associate with people similar to themselves, social persons associate with other social persons, whereas solitary persons associate with other solitary persons – leading to a positive degree correlation.

3.5 Semantic and Associative Networks

Semantic networks are a knowledge representation mechanism introduced by Quillian (1968). A **semantic network** in its original definition consists of **type nodes** (or concepts) and **links** between them (or their relations to each other), modeled as a directed graph with labelled edges. A semantic network encodes the objective meaning of a concept, as expressed by its relation to other concepts.

Quillian (1968) uses five different relations:

hyponymy or subclass-to-superclass relationships, usually called 'is a':

$$\text{plant} \xrightarrow{\text{is a}} \text{living structure}$$

modification pointers modify one concept by means of another:

$$\text{structure} \xrightarrow{\text{mod}} \text{living}$$

disjunction groups several concepts into a disjunctive set:

$$\text{air} \xrightarrow{\text{or}} \text{water} \xrightarrow{\text{or}} \text{earth}$$

conjunction denotes a relation between concepts that form a conjunctive set:

$$\text{living} \xrightarrow{\text{and}} \text{not animal} \xrightarrow{\text{and}} \text{with leaves}$$

open-ended category for all other relations, for example the relation 'from', where 'food' is the subject and 'water' is the object of the relation:

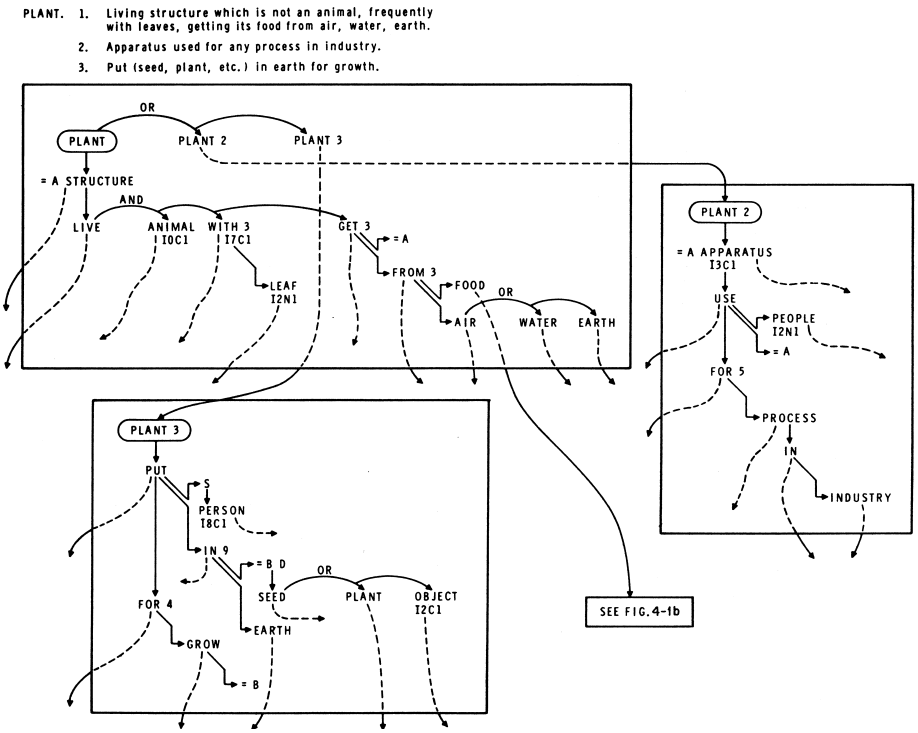
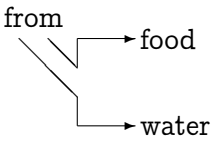


Figure 3.2: An example semantic network, illustrating three meanings for the word 'plant' (reproduced from Quillian, 1968, page 225).



Semantic networks were a first attempt to represent knowledge using a network structure; they do not support formal semantics or an inference mechanisms (Woods, 1975). Successor to semantic networks were description logics (Baader et al., 2003), which specified formal semantics while avoiding the problems of undecidability and computational complexity of full first-order logics.

The 'semantic web' initiative (Berners-Lee et al., 2001) recently rekindled interest in ontologies and knowledge representation. One of the cornerstones of the semantic web, the Web Ontology Language OWL[5], uses description logics as part of its specification.

Associative networks are a simplified precursor of semantic networks: An associative network contains only associations between concepts, but does not distinguish between different types of relations. The strength of the association between two concepts is expressed by the weight of the link connecting them, but no data is available

[5] http://www.w3.org/2004/OWL/, last visit on 2005/07/08.

as to why the two concepts are related. Associative networks are the foundation of associative retrieval (section 3.1.2).

3.6 Spreading Activation Search

Spreading activation search is a search technique for network graphs. It is characterized by the concept of 'activation energy' that is spread in the graph from 'activated nodes' to other nodes by means of outbound edges.

This search technique is motivated by models of neurophysiological activity. Neurons in a neural network that are activated are said to 'fire', transmitting (electrical) activation energy to other neurons with which they are connected. If the activation energy received by a neuron is sufficiently high, the neuron itself begins to 'fire', further spreading the activation in the neural network. According to Anderson (1983), it is unclear whether the concept of spreading activation can be applied to individual neurons, or whether it should rather be applied to sets of neurons. Spreading activation is the core of several theories concerning the organization of human memory, and has been applied extensively for search in semantic and associative networks.

A network graph $G = (V, E)$ consists of nodes $V = \{v_1, \ldots, v_n\}$ and directed edges $E \subseteq V \times V$. A weight matrix $C \in M_{|V|\times|V|}(\mathbb{R})$ contains the weight of each edge, with c_{ij} signifying the weight of the edge from node v_i to v_j (or 0 if no such edge exists.) Edge weights are usually positive; negative edge weights may be used to simulate inhibitory links.

A node v_i is said to be activated if at time t its activation energy $a_i^t > 0$. In the following, spreading activation is described for discrete time scales only; the continous case is of little interest for applications in IR.

Spread of activation occurs in several iterations, called 'pulses' in analogy to the operation of neurons. Each iteration consists of four steps, as detailed in figure 3.3:

1. pre-adjustment, decay:
   In order to determine the output energy for a node, a function f_o is applied to the activation level in the previous iteration:

   $$o_i^t = f_o(a_i^{t-1})$$

   The function f_o determines how energy is spread to neighbouring nodes.
2. spreading:
   input energy i_i^t is accumulated for each node in the network:

   $$i_i^t = \sum_{j=1}^{|V|} o_j^t c_{ji}$$

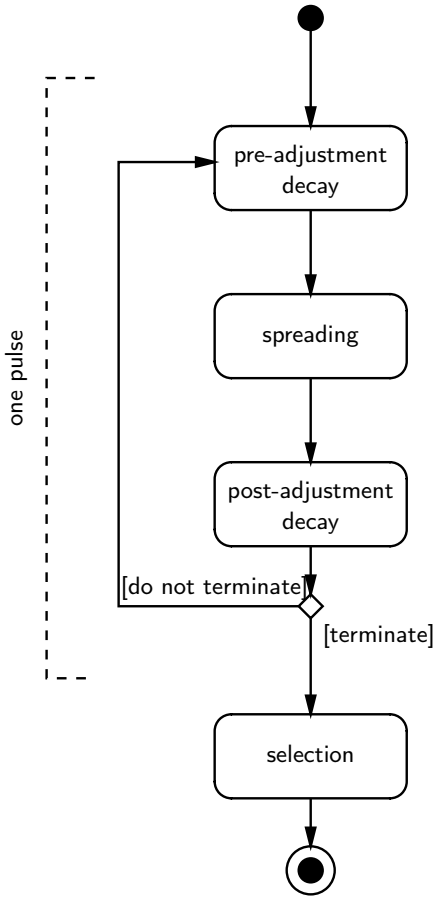


Figure 3.3: Activity diagram for spreading activation (after Preece, 1981)

3. post-adjustment, decay:
   The activation level for each node is determined from its input energy and its activation level at the last iteration:

$$a_i^t = f_a(a_i^{t-1}) + f_i(i_i^t)$$

4. termination check:
   When a fixed number of iterations has been reached, or other conditions have been met (for example, the amount of activation energy dissipated in the current iteration is zero, or is lower than a fixed threshold), the algorithm stops.

Spreading activation energy is an extremely flexible model. By carefully chosing the edge weights, the functions f_a, f_i and f_o, and the termination condition, a number of different search strategies can be implemented.

Preece (1981) distinguishes three types of pre-adjustment:

full strength spreading: each neighbouring node receives the full activation energy of the source node:

$$o_i^t = a_i^{t-1}$$

unit spreading: From each activated node, neighbouring nodes receive a fixed amount of energy, regardless of its activation energy:

$$o_i^t = \begin{cases} 1 & \text{if } a_i^{t-1} > 0 \\ 0 & \text{otherwise} \end{cases}$$

equal distribution spreading: Each receiving node gets an equal share of the output energy of the source node:

$$o_i^t = \frac{a_i^{t-1}}{\sum_{\substack{1 \le j \le |V| \\ (v_i, v_j) \in E}} c_{ij}}$$

During post-adjustment, the effect of the received energy is changed at the destination node. Several stratagies are used in this stage to limit the spread of activation:

retention: One may choose to not retain the previous activation level ($f_a \equiv 0$), or to decrease it by a fixed factor ($f_a(a) = \lambda a$ for $0 < \lambda < 1$).

thresholding: If the received energy is smaller than a predefined threshold ω, it may be dropped. A threshold function on the received energy acts as a 'noise filter', canceling out small changes in activation energy and limiting the number of activated nodes.

Popular choices for the threshold function are the Heaviside function ($\Theta(x) = 0$ for $x \le 0$, $\Theta(x) = 1$ otherwise), or sigmoidal functions like the tangens hyperbolicus:

$$f_i(i_i^t) = i_i^t \Theta(i_i^t - \omega) \text{ or } f_i(i_i^t) = i_i^t \tanh(i_i^t - \omega)$$

inverse destination frequency spreading: The input energy is divided by the sum of the weights of the incoming edges over which it was received:

$$f_i(i_i^t) = \frac{i_i^t}{\sum_{\substack{1 \le j \le |V| \\ (e_j, e_i) \in E}} c_{ji}}$$

A system with equal distribution spreading, no retention of activation energy, and no thresholding conserves the total energy in the system.

When using spreading activation search in a dense graph, one must take care to limit the spread of activation through the graph; otherwise, the whole graph will be activated after a few pulses. **Constrained spreading activation** adds further methods of limiting the dissipation of activation energy in the network. Commonly used heuristics include (Crestani, 1997):

distance constraint: Spread should stop after a certain distance from the originally activated nodes has been reached; this prevents the search from arriving at nodes that only have a tenuous connection, via several links, to the original nodes.

fan-out constraint: Spread should stop at nodes with a high fan-out, since these usually denote a very general concept, and further exploration from this concept is unlikely to lead to helpful results.

path constraint: The spread of activation energy should prefer links that contain more meaningful information if possible, and resort to links of certain other categories only if no other are available.

activation constraint: Activation is disseminated only from nodes whose initial activation value exceeds a certain threshold, which may differ depending on the node type.

The application of spreading activation search for text retrieval was studied by Salton and Buckley (1988a), who found its performance to be comparable to vector-space methods. Pirolli et al. (1996) used spreading activation to unify content-based and link-based information for searching the World Wide Web. It was also used by Crestani and Lee (2000) as part of the WebSCSA system, an agent browser that follows outgoing links from visited web pages and correlates them with the user's past interests.

An overview of spreading activation in information retrieval is found in (Crestani, 1997), who notes that the effectiveness of spreading activation search depends crucially on the structure of the network graph. Despite various prototype systems described in the literature, no commercial system implementing spreading activation search is available.

3.7 Retrieval Performance Evaluation

The evaluation of retrieval performance consists of an **evaluation scenario** (or **setting**), an **evaluation task**, and **evalution metrics** which provide a measure of the

D (total documents)	$PPos$ (predicted positive examples)	$PNeg = D \setminus PPos$ (predicted negative examples)
Pos (positive examples)	$TP = Pos \cap PPos$ (true positives)	$FN = PNeg \setminus Neg$ (false negatives)
$Neg = D \setminus Pos$ (negative examples)	$FP = PPos \setminus Pos$ (false positives)	$TN = Neg \cap PNeg$ (true negatives)

Figure 3.4: The confusion matrix lists possible subdivisions of the sets D, Pos and $PPos$.

performance. The choice of setting also determines the appropriate evaluation metrics.

We can distinguish broadly between interactive evaluation scenarios and batch scenarios. In an interactive setting, we measure the ability of a user to solve the evaluation task, using the information retrieval system under evaluation. An example task for this setting is finding out the answers for a questionnaire. We would measure the performance of the system in terms of the average number of questions attempted, the average number of questions answered correctly, and the time taken to fill out the questionnaire.

In a batch (or non-interactive) setting, we measure the ability of the system to find relevant documents as regards a query and rank them accordingly. Because the performance evaluation does not depend on the abilities of the users, experiments using batch settings are easily repeatable and comparable. A number of standard test collections for batch retrieval exist.

For a non-interactive setting, the individual evaluation task consists of a set of documents D and an information request q. The set of documents relevant to this query $Pos \subset D$ is usually determined by a human expert. The information retrieval system returns a set of answers $d_1, \ldots, d_k \in PPos \subset D$ in respect to the information request q, as well as a ranking function $rank : PPos \to \{1, \ldots, k\}$. The ranking function imposes an order on the returned documents.

The document sets D, Pos and $PPos$ can be further subdivided as seen in figure 3.7. True positives are documents deemed relevant by both the human expert and the information retrieval system. False positives are returned by the IR system, but were reckoned irrelevant to the query by the human expert. False negatives are documents relevant to the query which are not found by the system. True negatives are not returned by the system and are considered irrelevant by the human expert.

In cases where the ranking of the result list is not unique, interval arithmetic (Hayes, 2003) may be used. This phenomenon occurs in scoring information retrieval systems, when two or more documents are assigned the same score, and the desired document is one of these documents. Take for example, an IR system which returns three documents with the same score on rank 3, 4 and 5 of the result list, and returns the desired document d on rank 5. In this case, reporting the rank of d as 5 would be pessimistic, since the system might have returned it on rank 3 or 4 under other circumstances. Instead, the

rank is an interval: $\text{rank}(d) = [3,5]$ or $\text{rank}(d) = 4 \pm 1$. Average rank and other statistics need to be computed using interval arithmetic.

Alternatively, one may choose to report only the midpoint of the interval, or the actual rank reported by the IR system. Where detailed information about the ranking mechanisms is not available, the latter may be the only option.

Important evaluation metrics for non-interactive retrieval are introduced in the following subsections.

3.7.1 Precision and Recall

Precision and recall measure the performance of batch information retrieval systems and have been in use for this purpose for at least thirty years.

Precision is defined as the fraction of documents returned by the IR system that is actually relevant:

$$\text{Prec} = \frac{\text{TP}}{\text{PPos}}$$

Recall is the fraction of relevant documents returned by the IR system:

$$\text{Recall} = \frac{\text{TP}}{\text{Pos}}$$

Precision and recall do not take a ranking of the documents into account. They presume that an IR system returns a fixed set of answers for a given query, and that all returned documents are subsequently examined by the user – an appropriate assumption for early boolean retrieval systems.

In order to adapt this measure to ranking models, the result list is examined in order of increasing rank, and precision is measured when a specified **recall level** has been reached, ie. when a specified fraction of the relevant documents has been seen. An **average precision** is computed by averaging over the precision at certain standard recall levels. (For example, at a recall of 75%, 50% and 25%; interpolation may be necessary if the precision at the exact recall level cannot be determined.) **Average precision at seen relevant documents** averages over the precision at every relevant document in the result list.

For very large collections, the set of relevant documents Pos can be difficult (and expensive) to determine, as this task presumes knowledge of all documents in the collection.

Precision and recall are similar to the ROC (receiver operating characteristics) model (Hanley and McNeil, 1982) popular in machine learning. In difference to the ROC model, precision and recall do not take the number of true negatives into account. True negatives usually dominate in information retrieval, since only few documents in the document collection are relevant to a specific query. (See also (Fürnkranz and Flach, 2003) for a comparison of evaluation metrics.)

3.7.2 Metrics for Known-item Retrieval

In the **known-item retrieval task**, only a single document from the collection is presumed relevant to the query; the objective is to find this document as quickly as possible. The known-item task is especially popular in spoken document retrieval and retrieval of OCR documents; it was used at TREC-5 and TREC-6 in this function (Kantor and Voorhees, 1996; Garofolo et al., 1997).

Performance is measured by the rank at which the desired document appears in the result list. The **average rank** for a set of queries $q_1, \ldots, q_k$ and relevant documents $d_1, \ldots, d_k$ is

$$\overline{\text{rank}} = \frac{1}{k} \sum_{i=1}^{k} \text{rank}(d_i)$$

Another popular measure is the harmonic mean of the rank at which the desired document occurs; this is also called **inverse average inverse rank** in the known item retrieval context and is defined as

$$\text{IAIR} = \frac{k}{\sum_{i=1}^{k} (\text{rank}(d_i))^{-1}}$$

Both average rank and inverse average inverse rank score 1.0 for perfect retrieval; inverse average inverse rank has the advantage of rewarding systems that return the desired document early in the result list.

3.8 Summary

We describe three different models for information retrieval: Vector-space retrieval treats documents and queries as vectors in a term vector space; relevance measures are based on the similarity of document vectors and the query vector. Weighting schemes like the $\text{tf} \cdot \text{idf}$ scheme improve the performance of vector-space retrieval. Associative retrieval models terms and documents as nodes in an associative network and uses graph-based search techniques. In hyperlinked environments, spectral methods on the adjacency matrix of the hyperlink graph are common.

One such spectral method is described in detail: The PageRank algorithm is a well-known algorithm for link analysis and is commonly used for web retrieval. We describe the model it is based on, methods for computing it, and how to integrate it into an IR system.

A brief overview of personalized and collaborative retrieval describes existing attempts at integrating a user model into IR systems. User models are based on past interactions of the users with the system.

Statistical network analysis describes the properties of naturally occuring networks and devises models for them. Measures describing the characteristics of networks such as the web graph or social networks are introduced.

Semantic and associative networks are formalisms for knowledge representation. Together with spreading activation search, they form the basis for many early models of the human mind. Spreading activation search, especially when used with constraints, is a current IR technique for many applications.

We conclude the chapter by describing methods for evaluating the performance of an IR system. We distinguish between interactive and non-interactive evaluation scenarios and describe metrics for two variants of non-interactive evaluation.

Chapter 4

Related Work

As noted by Romano et al. (1999), the connection between information retrieval and social processes has not been extensively researched to date. Even though information seeking has long been recognized as a social process (Wilson, 1981, 1994), few projects support social interaction in the information retrieval process or exploit social networks to achieve better performance; Romano et al. (1999) calls this the 'IR paradox'.

This section describes systems that exhibit some characteristics of a social IR system.

4.1 Google

Google[1] was one of the first web search engines to incorporate analysis of the web graph into its ranking algorithms. The PageRank algorithm (see Brin and Page, 1998; Page et al., 1999) was a novelty among search engines at the time and was quickly singled out among independent observers as the main factor for its success. The publication of a tool for determining the PageRank value of a specific page (on a scale from one to ten) led to a frenzy among 'search engine optimizers' – consultants concerned with achieving a high rank for a specific page and query on leading search engines. The quest for a high PageRank value shaped the topological nature of the web graph. Common tactics include selling links from high PageRank sites to promote sites with lower PageRank, and installing 'link farms': autonomous networks of highly interlinked web sites with little and highly similar content, all for the purpose of increasing the PageRank value of a given web page.

Google today is a successful publicly traded corporation with a market capitalization of more than 80 billion U. S. dollar. It provides numerous free services, for example an email service[2], a UseNet archive[3], a photo organizer[4], and several specialized search engines for images[5], scholarly articles[6], weblogs[7] and others. The main source of revenue

[1] http://www.google.com/, last visit on 2005/09/18.
[2] http://www.gmail.com/
[3] http://news.google.com/
[4] http://picasa.google.com
[5] http://images.google.com/
[6] http://scholar.google.com/
[7] http://blogsearch.google.com/

for Google is its advertisement service that allows clients to place text-only ads on the result pages of the search engine.

The impact of PageRank on the quality of Google's search results is not known; as is common for a web search engine, the innards of its scoring algorithm are kept secret. Several other factors may account for its singular position among search engines today:

- Google's homepage is very clean and uncluttered, compared to competitors like Yahoo![8]. This may account for its popularity among users and its perceived quality.
- For a long time, Google crawled a much larger portion of the web than any of its competitors, thereby enabling it to find pages buried much deeper in the web. The depth of Google's index was only recently surpassed by Yahoo![9]; its result quality however, by popular opinion, was not.
- Google implemented a highly scalable and easily adaptable processing and storage architecture, centered around the 'map-reduce' paradigm borrowed from functional programming languages, and GoogleFS, a fault-tolerant distributed filesystem. The size of Google's compute grid is estimated to comprise between 10 000 and 100 000 CPUs, thus ensuring consistently high performance and availability.

To summarize, Google pioneered link analysis in information retrieval and managed to incorporate it into a highly successful product.

4.2 ReferralWeb

ReferralWeb (Kautz et al., 1997b,a) is a system for mining social relations from the web and exploring social networks. The authors describe it as 'combining of social networks and collaborative filtering'; its focus is extracting a social network from web pages, finding experts for a topic and linking the searcher to the expert by a path in the social network.

The ReferralWeb prototype bootstraps the social network by searching for web pages with an individual's name. From the result pages, proper names are extracted, using techniques from information extraction (Sundheim and Grishman, 1995). Social links between two individuals are determined by the ratio of web pages containing both names and web pages containing only a single name. This process is repeated recursively to determine the social neighbourhood of an individual. Social networks are also extracted from Usenet archives, coauthorships of scientific publications and organization charts.

Several operations are supported on the resulting social network. Paths from one person to another are used to determine a chain of referrals that links a searcher to an expert for a specific topic. A user can search for an expert on a topic either on the whole

[8] http://www.yahoo.com
[9] http://www.ysearchblog.com/archives/000172.html, last visit on 2005/09/19.

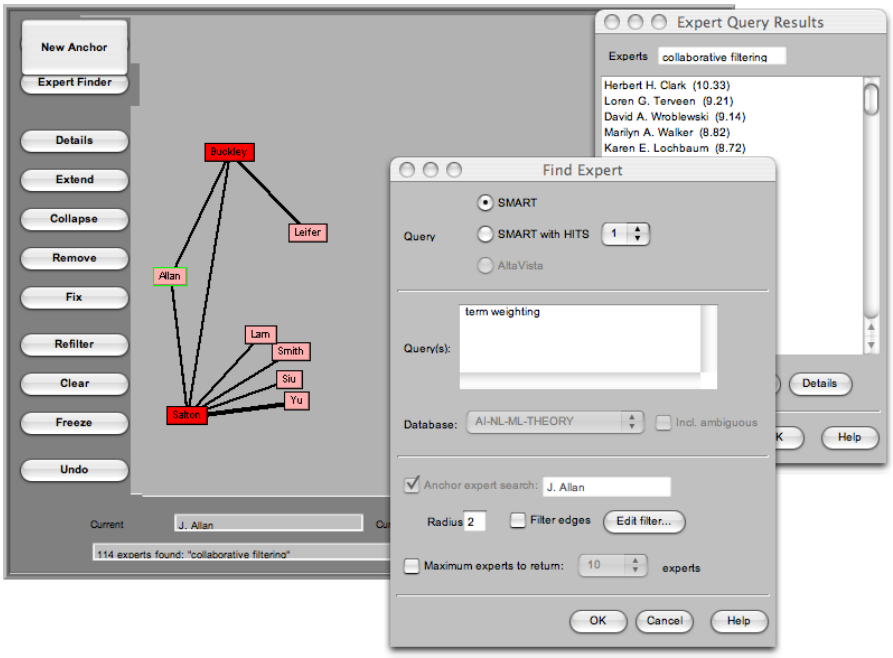


Figure 4.1: Screenshot of the ReferralWeb 2.0 prototype

social network, or just in his neighbourhood. The system also supports visualizing and exploring the social network in an interactive, graphical manner.

ReferralWeb differs from other social networking applications because it extracts social links from publicly available information on the web; it does not require the user to sign up with a service and explicitly name his colleagues and collaborators.

The prototype system was developed as part of Mehul A. Shah's master's thesis (Shah, 1997); a second implementation was performed by Yooki Park and is available on the web[10] (see also figure 4.1). A formal evaluation of ReferralWeb's effectiveness, as compared to other information retrieval systems, was not conducted to our knowledge.

4.3 Collaborative Information Retrieval Environment

The 'Collaborative Information Retrieval Environment' (CIRE) by Romano et al. (1999) combines features of information retrieval system and group support systems. A group support system is defined as a 'computer-based information system to support intellectual collaborative work'. Group support systems provide features to facilitate communication, deliberation, problem solving and decision making processes in groups.

CIRE is implemented on top of a conventional web search engine, AltaVista in this case. The user interface of the underlying search engine is augmented by additional interface

[10] http://foraker.research.att.com/refweb/version2/RefWeb.html, last visit on 2005/09/19.

elements to access the collaborative features of the system. The user's familiarity with this interface ensures a gentle learning curve for new users.

CIRE stores information about past queries, past results, and the browsing history of its users, as well as comments and relevance judgements for individual pages, thereby serving as a search memory for users. This information is also shared between users.

The asynchronous nature of the system allows users to search collaboratively even if they are geographically or temporally distributed. By accessing other users' queries and annotations, one can continue a research task where another user left off. The search memory also allows novice users to gain familiarity with the way experts use the system.

Romano et al. (1999) note that the collaborative features of the system were often ignored or forgotten by users; this is attributed to the non-intrusive nature of the system's user interface.

4.4 I-SPY

I-SPY is an experimental meta search engine developed at University College, Dublin, Ireland. As a meta search engine, it does not maintain its own index of web pages; instead, it relies on another web search engine (Google in I-SPY's case) for results. Results from the underlying search engine are re-ranked and presented to the user.

I-SPY implements collaborative ranking, borrowing ideas from collaborative filtering: It aggregates relevance judgements from a community of people and uses them in later searches for the same keywords to boost pages which are known to be good. The result list is stratified: Previously ranked pages are displayed first, followed by other pages from the result list of the underlying search engine. If the impact of relevance judgements is not discounted with time, this may lead to a fossilization of the result lists, presenting pages as relevant which have long since changed or become out of date.

Users are required to join a specific community before executing a query. Anonymity is thus ensured, as the usage data is aggregated among one community and cannot be traced back to one specific member. One user can only be part of one community at a time, requiring the user to change the community as the subject matter of his search changes. Since this step must be executed consciously by the user, it often leads to 'communities of one': Communities which consist only of one user (with names such as 'Pete's searches') and used only to track the search history – a task which could also be accomplished using simpler personalization systems.

I-SPY does not facilitate the formation of a community. It does not use information about the social relations between its users, and does not facilitate the formation of such relations.

The influence of collaborative ranking methods on user performance was evaluated in (Freyne and Smyth, 2004): Students were issued with a questionnaire of 25 questions and were asked to solve it, using I-Spy as a web search engine. A training group did not use collaborative ranking, but the usage data from the training group was fed into

the collaborative ranking process. The test group solved the same questionnaire, using collaborative ranking with usage data from the training group.

Using a training group of 45 students and a test group of 47 students, it was concluded that the test group indeed benefitted from the usage information from the training group: both the number of attempted questions and correctly solved questions increased, and the average position of results clicked significantly decreased.

4.5 Summary

We describe four systems that pioneered components of a social IR system:

Google is currently one of the most successful web retrieval companies. Google pioneered link analysis for web retrieval, which was quickly determined as one of the factors responsible for its success.

ReferralWeb mines social relations from the web and visualizes them. Its primary purpose is finding experts on a topic and finding paths of referrals between individuals.

The Collaborative Information Retrieval Environment CIRE combines information retrieval systems and group support systems, allowing users to collaborate on a retrieval task.

The I-SPY search engine implements collaborative ranking by keeping track of previously entered queries and documents, and re-ranking documents accordingly. Documents are promoted to the top of the rank list if they were previously selected for a similar query.

Chapter 5

Models

This chapter introduces a domain model for social information retrieval. The domain model identifies entities pertinent to the retrieval task, as well as their relations. It forms the basis for the retrieval techniques described in chapter 6.

5.1 A Domain Model for Social IR

Social information retrieval is defined as the incorporation of information about social networks and relationships into the information retrieval process:

$$\text{social information retrieval} = \text{social networks} + \text{information retrieval}$$

The traditional models for information retrieval concern themselves with documents, queries, and their relations to each other: A document is relevant to a query, a document references other documents, a query is similar to other queries. Likewise, social network analysis models individuals and their relations with each other: friends and family, acquaintances, collaborators, or sexual relationships. Information retrieval systems traditionally do not model individuals, neither in their role as users of the system, nor as authors of the retrieved documents, and social networks do not incorporate retrievable content. A simplistic view of the domains is pictured in figure 5.1.

Social IR combines the two models with each other. By incorporating individuals into the model, we gain a greater insight into their role in the information retrieval and production process (figure 5.2). New associations between the entities become apparent: Individuals appear in their role as information producers or information consumers, queries relate to an individual's information needs, or describe a topic about which an individual possesses knowledge.

A social IR system is characterized by the presence of all three types of entities: documents, queries, and individuals. Most systems will only use a subset of the possible associations between the entities, depending on the domain of the system. Modeling the relations between individuals is mandatory for a social IR system; all other types of associations are optional, as long as all three entities have an association with at least one other.

The motivation for social IR is rooted in the belief that an information producer and his product cannot be separated: Information does not spring into existence spontaneously;

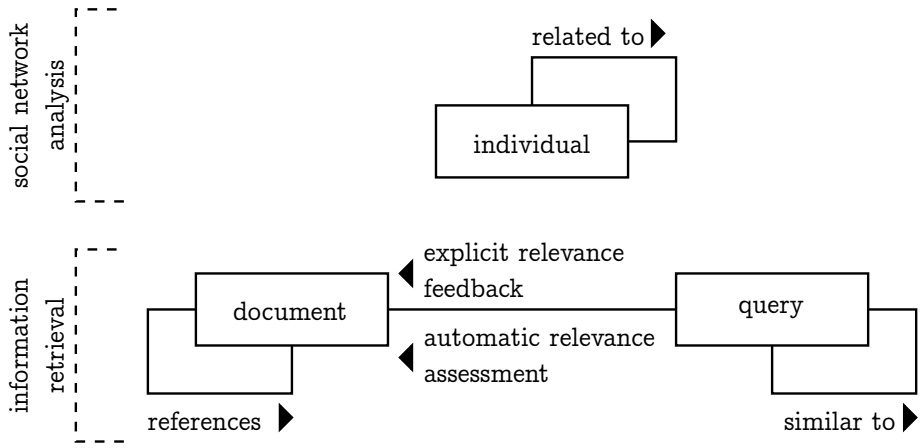


Figure 5.1: Traditional domains of information retrieval and social network analysis

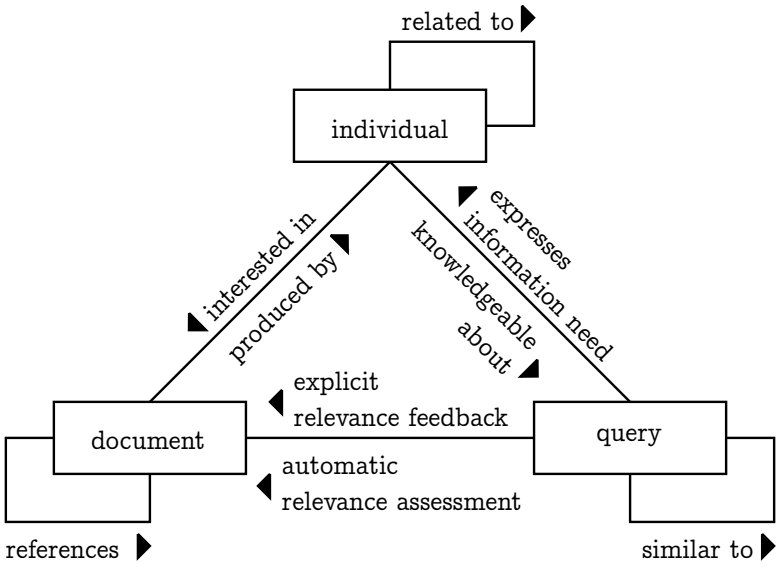


Figure 5.2: A domain model for social information retrieval

it is always produced by an individual as an expression of his state of mind. This implies that any judgement about the product can be used to infer a judgement about the producer, and vice-versa. Social IR uses this observation to apply judgements about the information producer to his products. Judgements about the producer are derived from an analysis of the social network. The producer's relationship with his peers is used to draw conclusions about the nature of his products.

Understanding the social fabric in which information production takes place is especially important when only limited understanding of the documents or the information needs is available. Traditional information retrieval techniques which are based solely on analysing document content, while very successful in many contexts, fail badly when the information need is underspecified, and when a large number of relevant documents exist. In this sense, social IR can be understood as a formalization of search techniques we commonly use to assess the quality of information – by looking at the author's standing in his community.

An example may serve to clarify this point: Suppose that we try to find an authoritative scientific paper on a certain topic. A search in a database reveals publications by five different authors. Further searches reveal that the first author has collaborated with three of the others, whereas the last author is a 'lone ranger' and has not collaborated with any of the other authors. When trying to choose where to begin, we would probably chose a paper by the first author, because he has the best standing in the community of his peers, and as such may be presumed to be the most authoritative source. We would probably disregard the last author, since he has no connections to other people studying the same field.

The same principle can be applied to other instances of information production in a social environment: We tend to favour authors who engage in active collaboration and exchange of ideas; this is usually seen as a sign of thorough and diligent work.

5.2 Mediums for Social IR

Referring to the examples of social information spaces in the introduction (section 1.2), we now identify entities, roles, and associations for selected examples, and show that these mediums are valid domains for social IR:

- The semantic web (as described in section 1.2.6) is a prime example for social search, as it supports the explicit modeling of all three entity types: The FOAF standard specifies a type <foaf:Person> for modeling individuals, and a property <foaf:knows> for relations between individuals. The Dublin Core standard specifies the <dc:creator> relationship for linking documents to their author; FOAF specifies a similar relationship via the <foaf:made>/<foaf:maker> property. The <dc:relation> property identifies associations between documents, and the <dc:subject> property contains the topic and keywords for a document. A per-

son's interests are expressed with the <foaf:interest> property. The OpenSearch standard allows one to publish information about which documents are relevant to a specific query. Combined, these three standards allow for a complete and machine-readable description of all parts of the domain.

- Mailing list archives (section 1.2.4) contain less explicit information. Individual users are identified by their email address; thus, authorship information is easily available for each message. References between messages are extracted from the `In-Reply-To:` and `References:` header lines; these are not fully supported by every email client and may lead to information loss. Information extraction techniques may also be used to identify follow-up messages. Relations between individuals are identified based on whether two individuals corresponded with each other on the mailing list. A conventional text search engine provides relevance assessments for queries and documents. Note that in this case, no explicit relation between queries and individuals is extracted; rather, a person's interests are characterized by the union of the messages he wrote.

- While the previous two examples were primarily concerned with modeling the information production process, the ISKODOR system (introduced in section 1.2.5) contains a different subset of the social IR model: the information retrieval process. Individuals submit queries to the system and thereby express their information needs. Explicit feedback determines which document is relevant to a query. By recording the individual who submitted the feedback, associations between individuals and documents are stored. Measures like peer relevance (Gül, 2004) express relations between users. ISKODOR lacks associations between documents or between queries; it nevertheless contains all components to enable the use of social search techniques.

From this comparison, we see that the social IR model is applicable in diverse situations. Its key concepts can be identified in information retrieval environments as well as information production and sharing environments.

5.3 Additional Aspects

This section discusses additional aspects of applying social retrieval methods. We discuss practical issues for their implementation as well as social and philosophical ramifications resulting from their application.

5.3.1 Bootstrapping the Social Network

Social information retrieval presupposes the existence of a social network between the content producers. In some cases, this network can be inferred based on the content;

for example, the coauthorship network of scientific publications is such a social network. One may also use social networks that are completely unrelated to the produced content, for example by asking the authors to name their peers in the network explicitly.

In the latter case, it can be difficult to persuade users of an existing system to enter data about their personal contacts, especially if they do not perceive an immediate advantage in doing so. On the other hand, experiences with existing systems, for example social networking services like Friendster[1], Orkut[2] or openBC[3], show that participants readily connect to other users of the same system, forming an intricate and reasonably complete social network. The perceived value of disclosing one's social neighbourhood seems to outweigh potential privacy concerns. A combination of such systems and an information retrieval system constitutes a fertile ground for the application of social search techniques.

Unsuccessful formation of the social network results in performance degradation of the retrieval process. In particular, algorithms that use the global structure of the social network are ineffective if the network is very sparse or fractured into many different components; one such network is described in section 7.1.2. In this case, one has to resort to other IR methods, or limit the influence of the social component of the retrieval system.

5.3.2 Privacy, Anonymity, and Plausible Deniability

Privacy matters quickly become a concern as more and more information can be tied to a specific person, especially when this information concerns said person's interests and social ties to other people. Therefore, a user of such a system needs to be aware of the information that is available on him and how it is used. He needs to know that information about certain actions is recorded and is available to others.

We describe techniques that actively make use of the fact that information can be tied to a specific user and can be made available to others – in order to identify relevant individuals and their content. As such, we believe that they should only be used in environments where the information is publicly available anyway. We oppose to them being used in combination with information gathering techniques such as evaluating browsing histories: In such an environment, the user has no direct control about the information that is published about him.

In other applications, measures need to be taken to ensure anonymity of the individual users and, preferably, plausible deniability (an individual user can plausibly deny that a specific piece of information originated from him.) How to employ our techniques in such an environment is not subject of this thesis.

[1] http://www.friendster.com/, last visit on 2005/10/11.
[2] http://www.orkut.com/, last visit on 2005/10/11.
[3] http://www.openbc.com/, last visit on 2005/10/11.

5.4 Classification and Comparison to Other Approaches

In order to classify the social information retrieval approach, we adopt a domain model of information retrieval as in figure 5.3. In difference to process models for IR, for example (Baeza-Yates and Ribeiro-Neto, 1999, p. 10), or earlier domain models as in (Frakes and Baeza-Yates, 1992, p. 2), this domain model omits implementation details. It focuses on three main aspects of information retrieval: Aspects of human-computer interaction, the retrievable content, and the user's information needs.

User aspects like interface design and information-seeking behaviour are an important factor for the effectiveness of an IR system. While the inclusion of individuals into the IR system is an important characteristic of social IR, we do not describe the implementation of a complete IR system and as such do not deal directly with issues of user interaction.

As regards the retrievable content, most of the literature is concerned with operations on natural-language text, like language detection, stopword removal, stemming, and term selection. A computer's understanding of natural language is limited, and those language techniques which allow a deeper understanding of the semantics of a text are prohibitively expensive and are only used for information retrieval in limited settings: For example, shallow parsing is used in question-answering systems.

Metadata – data *about* data – aids the evaluation of documents by providing context. Metadata takes diverse forms, the simplest being descriptive data, for example author or publication date. The analysis of the associations of a particular document is especially important in web retrieval, due to the hyperlinked nature of the web.

Social IR takes association analysis one step further, by not only analysing associations between documents, but also between documents and their authors, and between authors. It works on a deeper level than bibliographic reference analysis and is able to infer evidence about documents where other methods of association analysis fail.

The social IR model (as in figure 5.2) also incorporates associations between individuals and queries, and is able to model collective information needs. Adopting the notion that the same representation can be used for queries and documents (as it is present, for example, in the vector space model), the techniques for retrievable content can also be applied to queries. We have already described integrated information sharing systems like ISKODOR that treat queries in a similar way to retrievable content.

Social IR also shares features with systems that do not fit in the domain model for information retrieval. Notable examples are the following:

Collaborative filtering systems, also called **recommender systems**, tailor the results of the search to one specific user. A profile of this user must be available to the system, which may either be gathered from previous usage data, or explicitly constructed. Associations between individual users of the system are either not present at all (for systems that take the complete data into account when producing recommendations), or generated based on the agreement of the users' profiles. Collaborative filtering systems make no use of social relations between users.

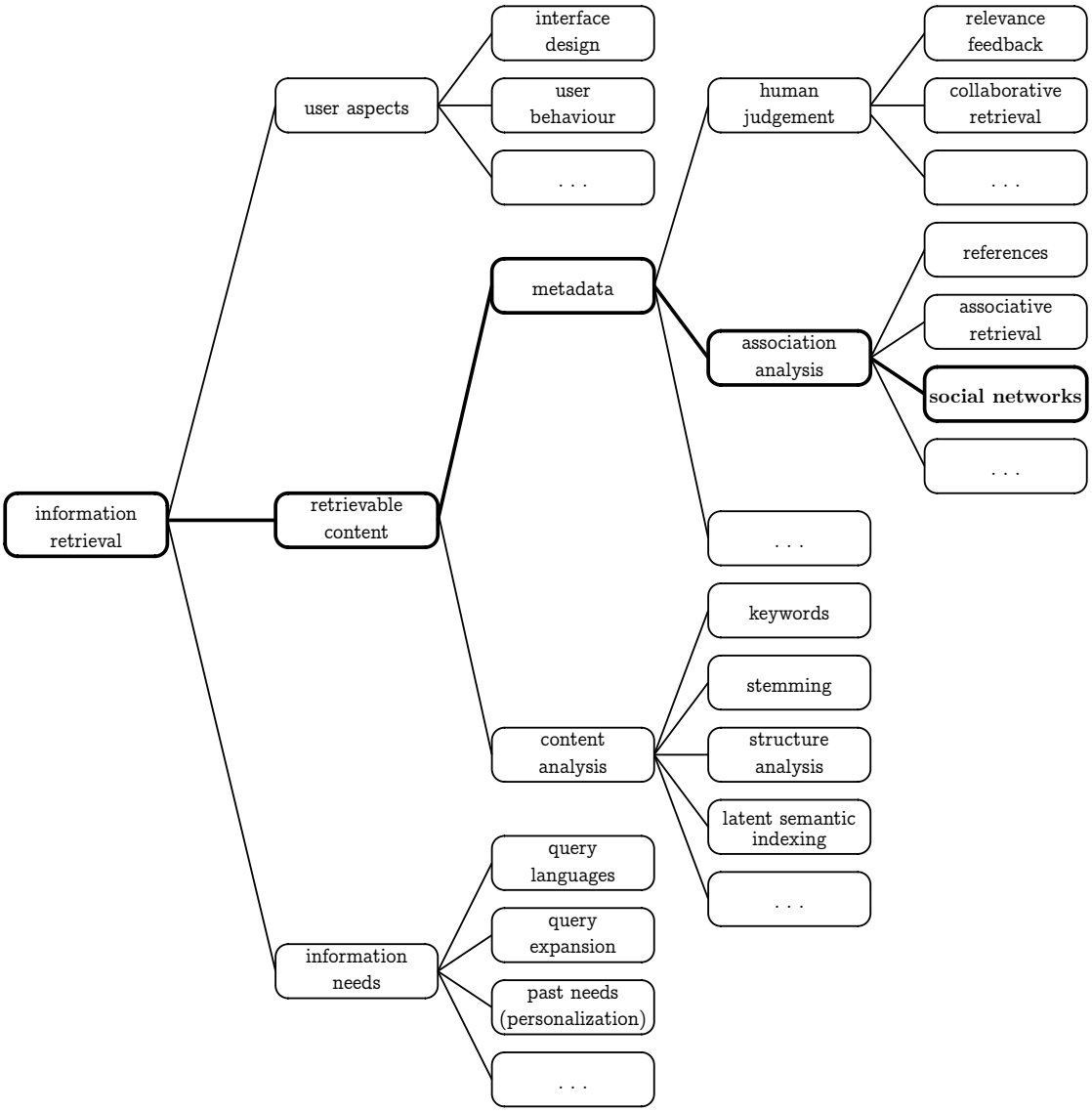


Figure 5.3: Classification of social networks in information retrieval. Social networks are a form of metadata for retrievable content, closely related to other association analysis like bibliographic reference analysis.

Collaborative ranking systems aggregate explicit relevance judgements from a group of people. These relevance judgements are used for ranking of result lists. Collaborative ranking systems usually treat the group as uniform: A relevance judgement of any member is worth the same as any other.

Expert location may be considered the inverse of social IR: In expert location, one tries to determine the authority of a person based on the content they produce, whereas in social IR, we try to determine the quality of content based on the authority of the author.

Trust models treat links in a social network as a measure of trust or distrust. In difference to this, links in social IR are not an explicit measure of trust, but rather one of social interaction.

5.5 Summary

We describe a domain model for social information retrieval that includes three prominent types of entities: documents, queries and individuals. Possible associations between these entities are given in order to justify their inclusion in the retrieval model. We illustrate the application of the domain model, using mediums from section 1.2 as examples.

Practical issues for the application of the domain model are discussed, as well as privacy concerns and acceptance problems.

Social information retrieval is classified as a kind of metadata analysis and related to other types of association analysis. It is compared to similar approaches that lack important characteristics of a social IR system.

Chapter 6

Techniques

As detailed in chapter 5, what sets social IR apart from other information retrieval settings is the inclusion of a social network. The constituents of the social network are not the objective of the retrieval process; instead, they provide additional information about the retrievable items. This information needs to be integrated in the retrieval process in a meaningful way.

6.1 Associative Network Model

Based on the domain model in figure 5.2, we use an associative network (section 3.5) as the underlying representation. An associative network is a graph of information items, with unlabeled, weighted, directed or undirected edges ('associations') between nodes. In agreement with the domain model, we use three kinds of nodes: for individuals, documents, and queries.

For modeling one specific social IR task, we use only a subset of the possible associations, as in figure 6.1. We model a domain that includes documents, associations with a document's author, and a social network between authors. The relevance of a document as regards a query is determined automatically, using standard text retrieval methods.

For a set of individuals I and a set of documents D, the domain is represented by a weighted, directed graph $G = (V, E)$, where $V = I \uplus D$ and $E \subseteq V \times V$. A weight matrix $C \in M_{V\times V}(\mathbb{R}_{\geq 0})$ contains the weight of the edges. For edges between individuals $e \in I \times I$, the weight function expresses the strength of a social relationship between two individuals; for other types of edges, suitable weight are chosen. This model is also able to accomodate for unweighted social networks (by using a uniform weight function) and undirected networks (by using two directed edges (v, v') and (v', v) for every undirected edge $\{v, v'\}$).

The task on this domain is the retrieval of documents from keyword queries. This task is the most common task in information retrieval, which ensures comparability with other systems. Systems that store associations between users and queries, or between queries and documents, are mostly found in the experimental field of personalized and collaborative retrieval; they have not found their way into the mainstream of IR yet.

Limiting ourselves in this way allows us to formulate definite goals and develop algorithms which can be compared with mainstream IR systems.

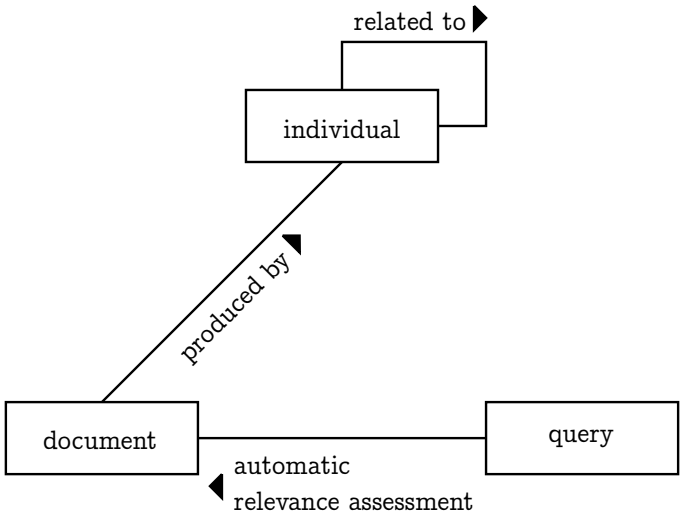


Figure 6.1: A model for a concrete social IR task, using only a subset of the associations present in the general domain model.

We describe two techniques for this task: one global technique, based on the PageRank algorithm, and one local technique, based on spreading activation search. The global technique is motivated by the idea that we would be more interested to read what an authoritative person has to say about a topic, regardless of what the topic is. The local techniques implements the notion that an author is knowledgeable about a subject if he is connected to other authors working in the same field.

6.2 Vector-Space Model

The vector-space model is not a technique for social information retrieval, as it does not include social networks. We use the vector-space model for two purposes: It is a method for matching keyword queries against documents in the collection and is used in this role in social IR techniques described later in this chapter. We also use it as a baseline method for measuring the performance of social IR in chapter 7.

We do not to re-implement vector-space search, but use a freely available implementation instead. Specifically, our system is based on the the Lucene[1] library, an open-source information retrieval library. Lucene uses a modified vector-space model; the main scoring formula is

$$\text{score}(q, d) = \frac{\sum_{t \in q} \sqrt{\text{tf}(t, d)} \cdot \text{idf}(t)^2}{\sqrt{\sum_{t \in q} \text{idf}(t)^2} \sqrt{\sum_{t \in d} \text{tf}(t, d)}} \tag{6.1}$$

[1] http://lucene.apache.org/, last visit on 2005/09/09.

where

$$\mathrm{idf}(t) = \log \frac{|D|}{\mathrm{df}(t) + 1} + 1$$

Scores are normalized to fall in a range of 0.0 to 1.0.

This weighting scheme is easily related to the standard vector-space model by using $\sqrt{\mathrm{tf}(t,d)}$ instead of $\mathrm{tf}(t,d)$ and defining $\mathrm{tf}(t,q) \equiv 1$. Then

$$\begin{aligned}
\mathrm{score}(q,d) = \cos\angle(\mathbf{q},\mathbf{d}) &= \frac{\mathbf{q}\cdot\mathbf{d}}{\|\mathbf{q}\|\cdot\|\mathbf{d}\|} \\
&= \frac{\sum_{t\in T}\left(\sqrt{\mathrm{tf}(t,q)}\,\mathrm{idf}(t)\right)\left(\sqrt{\mathrm{tf}(t,d)}\,\mathrm{idf}(t)\right)}{\sqrt{\sum_{t\in T}\left(\sqrt{\mathrm{tf}(t,q)}\,\mathrm{idf}(t)\right)^2}\sqrt{\sum_{t\in T}\left(\sqrt{\mathrm{tf}(t,d)}\,\mathrm{idf}(t)\right)^2}} \\
&= \frac{\sum_{t\in q}\sqrt{\mathrm{tf}(t,d)}\,\mathrm{idf}(t)^2}{\sqrt{\sum_{t\in q}\mathrm{idf}(t)^2}\sqrt{\sum_{t\in d}\mathrm{tf}(t,d)\,\mathrm{idf}(t)^2}}
\end{aligned}$$

By omitting the term $\mathrm{idf}(t)^2$ from the term $\sqrt{\sum_{t\in d}\mathrm{tf}(t,d)\,\mathrm{idf}(t)^2}$ in the denominator, one arrives at the main scoring formula in equation (6.1). Omitting the inverse document frequency from the document normalization factor allows one to precompute this factor and store it in the index; otherwise it would be necessary to recompute the normalization factors every time a document is added or deleted from the index.

Lucene includes various modifications of the scoring method to account for partial matches and for the proximity of terms; these are not used in our experiments in section 7 and are not described here.

The exact formulation of the weighting formula is not crucial; one could also replace the baseline method by different version of the vector-space model, or use a probabilistic approach. This fact allows us to replace the text retrieval component by a state of the art implementation.

6.3 PageRank

PageRank is a global authority measure for graphs; how to compute it is described in section 3.2. Its primary use is an authority measure for web pages.

6.3.1 Applicability of PageRank

In this section, we compare the web graph with social networks and determine the applicability of PageRank to social networks.

The PageRank algorithm in its formulation as in equation (3.1) is equivalent to the power method for computing the dominant eigenvector of a matrix. The speed of convergence for the power method depends on the quotient $\frac{\lambda_2}{\lambda_1}$; for stochastic matrices, λ_1

is always 1. Therefore, the convergence of PageRank depends on the magnitude of λ_2, which is small compared to λ_1 for power-law graphs.

The power law holds true for the web graph; recent studies have determined that the in-degree of nodes on the web graph follows a power-law distribution with an exponent of about $\gamma \approx 2.1$, thus ensuring rapid convergence of the PageRank computation. The same power law is applicable to social networks, making them similarly suited for PageRank analysis.

Another prerequisite for convergence of the PageRank algorithm is that the underlying Markov chain is ergodic, ie. that the random walker has a finite probability of re-visiting every node. This is usually ensured by introducing the 'teleportation step'; but even with teleportation, unintentional effects occur on graphs with several connected components. For example, a small component that is heavily interlinked (or even a single node linking to itself) may have a disproportionate amount of PageRank bestowed on it, compared to nodes in larger connected components. Increasing the parameter ϵ in equation (3.1) ameliorates this problem, but does not solve it.

Empirical analysis of the web graph (Broder et al., 2000) showed that 91% of all surveyed pages are part of a single giant weak component. This number is well in agreement with random graph theory, which predicts that a random graph with more than $\log |V|$ edges per node will consist of one 'giant connected component' of size $\Theta(|V|)$ (see for example Janson et al., 1993).

If one takes the direction of hyperlinks into account, the largest strongly connected component contains only 28% of all nodes, and that the probability of a path existing from randomly chosen source and destination nodes is just 24%. This is a significant deviation from PageRank's premise that every page can be reached from every other page. It is still unclear whether this structure is an artifact of the web, or whether it is indeed typical for random directed power-law graphs. Preliminary results from Newman et al. (2001) indicate that the 'bow-tie structure' of the web (a term coined by Broder et al. (2000)) is close to that of a random directed power-law graph.

Similar analysis of social networks (from Newman, 2001) was conducted on scientific collaboration networks. For a collaboration network extracted from the MEDLINE database, 91% of all authors are part of a single connected component. Most models treat social networks as undirected graphs, which accounts for the larger percentage of nodes that are reachable from each other. No survey is available that examines directed social networks; if the 'bow-tie structure' is indeed a characteristic of random directed power-law graphs, it is also to be expected for social networks.

The similarities in structure of the web graph and social networks suggest the use of PageRank as an importance measure for individuals in a social network. For the web graph, PageRank has a very intuitive interpretation; namely, it is the amount of time a random surfer would spend on a given page. For social networks, especially in the context of information production, there is no such intuitive interpretation. One might imagine a 'book of knowledge' that is passed along social links, for every author

Table 6.1: PageRank scores for the coauthorship network of the SIGIR corpus. Scores are normalized and are computed with a teleportation probability of $\epsilon = 0.3$.

rank	name	PageRank
1.	Bruce W. Croft	7.929
2.	Clement T. Yu	4.716
3.	James P. Callan	4.092
4.	Norbert Fuhr	3.731
5.	Susan T. Dumais	3.731
6.	Mark Sanderson	3.601
7.	Nicholas J. Belkin	3.518
8.	Vijay V. Raghavan	3.303
9.	James Allan	3.200
10.	Jan O. Pedersen	3.135
11.	Justin Zobel	2.992
12.	Jian Yun Nie	2.982
13.	Stephen E. Robertson	2.959
14.	C. J. van Rijsbergen	2.856
15.	Peter Bruza	2.779
16.	Alistair Moffat	2.757
17.	Maristella Agosti	2.588
18.	Yasushi Ogawa	2.544
19.	Gareth J. Jones	2.493
20.	Sung Hyon Myaeng	2.492

to look at while it is in his possession. Under this interpretation, the PageRank value of an author would be the amount of time that this book is in his possession, ie. the amount of time he has to copy material from the 'book of knowledge'. (The hungarian mathematician Paul Erdős frequently referred to 'The Book', an imaginary book which contains all the most elegant mathematical proofs.)

In order to get an idea of the application of PageRank to a social network, it is instructive to compute the PageRank scores for a well-known social network. We computed PageRank scores for a coauthorship network extracted from 25 years of SIGIR proceedings (from 1978–2003); the twenty highest-ranking authors are listed in table 6.1. For anyone working in information retrieval, most if not all of the names in the list will be very familiar and will be recognized as authorities of the field. We also computed PageRank scores for the social network extracted from a mailing list archive of the 'origami-l' mailing list; the highest-ranking individuals are listed in table 6.2 and will be equally recognizable if one is familiar with the mailing list.

As far as we can conclude from these examples, PageRank is a measure that corre-

Table 6.2: PageRank scores for the social network of the 'origami-l' corpus. Scores are normalized and are computed with a teleportation probability of $\epsilon = 0.3$.

rank	name	address	PageRank
1.	David Lister	DLister891@...	8.831
2.	Joseph Wu	josephwu@...	8.377
3.	Michael Ujin Sanders	fightflipnfold@...	8.179
4.	Jose Tomas Buitrago	buitrago@...	7.452
5.	Mark Kennedy	KennedyM@...	6.898
6.	Julia Palffy	jupalffy@...	6.681
7.	Candice Bradley	candice.bradley@...	6.173
8.	Lar deSouza	fresco@...	5.943
9.	Dorothy Engleman	FoldingCA@...	5.007
10.	Leong Cheng Chit	leongccr@...	4.670
11.	Dorothy Kaplan	DORIGAMI@...	4.660
12.	J. C. Neal	jcneal@...	4.308
13.	Joshua Koppel	Skiffy1@...	4.285
14.	Juan Carlo Rodrigues	juancarlor@...	4.102
15.	Rick Beech	Ricknbeech@...	4.068
16.	Zack Brown	zbrown@...	4.067
17.	'Nathan'	rockmanex6@...	3.905
18.	Janet Hamilton	mikeinnj@...	3.894
19.	Marilyn Lewis	Abbmackdes@...	3.880
20.	Kenneth Kawamura	MadHawn@...	3.847

sponds with our intuitive sense of authority.

6.3.2 Applying PageRank to Social IR

For the implementation of PageRank, we use only the social network, that is, the graph $G[I]$. We compute a PageRank score r_i for every node i in the social network, as in section 3.2. We ignore the fact that several disconnected components may exist in the social network: Since they are small compared to the giant component, they can be expected to contribute little to the document set, which means that documents produced by individuals not in the giant component will only be relevant for very few of the expected queries. We use a bias of $\epsilon = 0.3$, further ameliorating the problem.

Another possibility is to employ topic-sensitive PageRank (Haveliwala, 2002) in order to bias the PageRank computation against nodes in smaller components. The uniform teleportation step is replaced by a non-uniform teleportation probability that depends on the size of the component of which the target node is a member.

At this stage, one may choose to normalize the PageRank scores, so that $\bar{r} = 1$; since $\sum_{i \in I} r_i = 1$, this is equivalent to multiplying each r_i by $|I|$. Since we use PageRank in a way that is invariant to normalization, we skip this step.

The score r_i is then assigned to to the documents:

$$\forall\, d \in D\ \forall\, i \in I : (i, d) \in E \Rightarrow r_d = r_i$$

If a document has more than one author, one has the option of either accumulating the PageRank scores ($r_d = \sum_{(i,d) \in E} r_i$), or of chosing either the maximum, minimum, or average of the PageRank scores of the authors. If the edges between nodes for individuals and document nodes are non-uniform in weight, one can also incorporate this weight information when transferring PageRank scores from authors to documents.

r_d is a global score expressing the 'importance' of each document (as derived from the 'importance' of its author or authors.) It needs to be combined with a conventional text retrieval system in order to produce results that are relevant to a specific query.

6.3.3 Integrating PageRank

As described in section 6.2, we employ a modified vector-space model. For a query q, the text retrieval system produces a set of relevant document $D_q \subset D$ as well as a score $\text{score}(q, d)$ for every document. The inclusion of r_d does not affect the result set D_q; it only influences the ranking of the documents, enabling the user to find relevant documents more quickly.

There are several models for combining PageRank with a text retrieval system. The simplest method is to sort the documents $d \in D_q$ by their PageRank score, and present those with the highest r_d to the user first. However, this method only works when a high precision of the result set is ensured (as noted by Page et al., 1999). For example, when

browsing an ontology or a document catalog, one may choose to order the documents in one category by their score r_d, in order to display the most important documents first.

A linear combination of relevance scores

$$\alpha \, \text{score}(q, d) + \beta r_d$$

offers a rich potential for optimizing the impact of the PageRank score in regard to the relevance score. Because of the differing distributions of score and r, it may be necessary to transform the PageRank scores. Zaragoza et al. (2004) suggest using $\log r$ or $(1 + \exp(-\log r + b))^{-1}$ instead of r, after normalizing the PageRank scores. The parameters α, β and b need to determined by experimentation.

A very simple method of combining PageRank and relevance scores is

$$r_d \cdot \text{score}(q, d) \tag{6.2}$$

For our purposes, this method has the advantage of not having tunable parameters, and being invariant to normalization. We choose this method for the experiments in chapter 7.

6.4 Spreading Activation Search

Spreading activation search (see section 3.6) is a very flexible formalism for expressing search techniques on graphs. It is based on the notion of 'activation energy' which spreads from node to node via outgoing edges. Spread of activation occurs in discrete timesteps called 'pulses', after each of which the received activation is accumulated and added to the residual activation from the last iteration.

A pure, unconstrained spreading activation search results in the complete network being activated after a low number of pulses. Small-world networks such as social networks aggravate this effect, due to their low average path length. Therefore, the spread of activation must be carefully limited and directed – mimicking a kind of inference process.

In information retrieval systems, spreading activation search is often used in an interactive fashion: The user is presented with a set of activated nodes after each pulse, at which point he can choose to stop the search process, to drop nodes not matching his information needs, or to guide the activation towards a suitable direction. In difference, automatic spreading activation search proceeds according to predetermined activation rules and stops when a termination condition has been met. Due to our choice of evaluation scenario (section 7), interactive spreading activation search is not a viable option.

6.4.1 Adjustments and Constraints

When applying spreading activation to social IR, we mimic an inference process, similar to the process one would apply to infer the authority of an author from his collaborators:

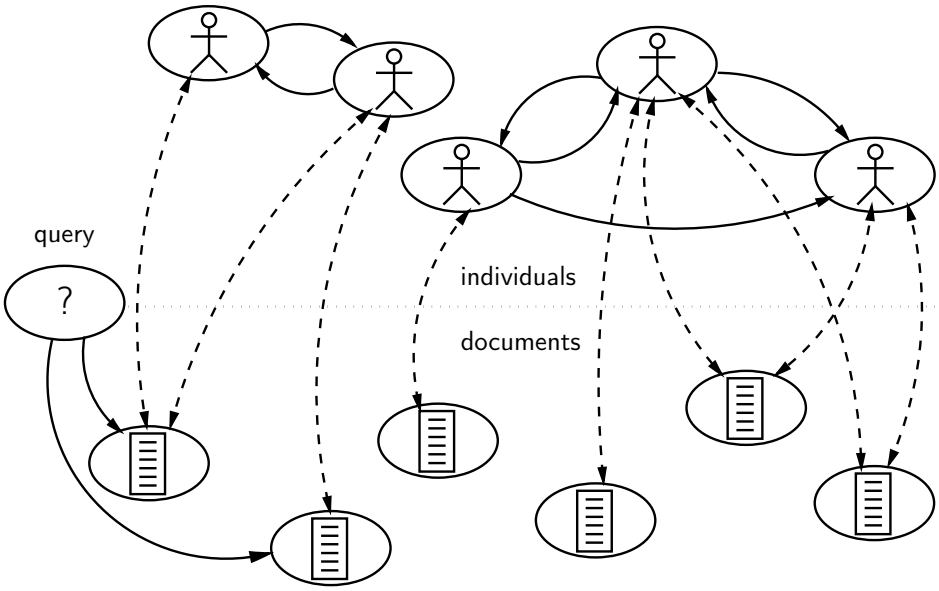


Figure 6.2: An associative network models the relationship between users as well as between items of content. Query nodes (denoted by '?') are transient nodes introduced into the network to express the relevance of a document as regards a query.

- The initial relevance of a document as regards a query is determined using an automatic information retrieval system; we use a system based on the vector-space model as in section 6.2.
- Authors of relevant documents are presumed to be experts as regards the query topic.
- An author is presumed to be authoritative if he has social ties with many experts. Likewise, he is presumed to be authoritative if he has written many documents about the topic.
- The relevance of a message depends on both its initial relevance (as estimated by a text retrieval system) and the authority of the author.

We implement spreading activation search on an associative network as in figure 6.2. The user's information needs are represented by a query node (denoted by '?' in figure 6.2), which stores the query keywords. The underlying text retrieval system estimates the relevance of the documents as regards the query keywords and adds edges from the query nodes to the document nodes accordingly. The edges are weighted according to the relevance score produced by the underlying retrieval system and are in a range between 0.0 and 1.0.

The query node is initially activated with a fixed amount of activation energy. Spreading activation proceeds according to the following rules and constraints:

- We constrain spread to nodes with a distance of two links or less to the initial query node. This constraint allows activation of neighbouring document nodes of the query node (with a distance of one) and their author nodes (with a distance of two).
- Spreading terminates after four iterations. Combined with a distance constraint of two, four iterations spread the activation energy from the query node to the author nodes, and back to the document nodes.
- In the pre-adjustment stage, we use full strength spreading, as this type of pre-adjustment rewards nodes with a high degree: Nodes with a high fan-out serve as multipliers and increase the amount of activation spreading through the network. We do not use equal distribution spreading since it conserves the amount of activation energy, and penalizes nodes with a high fan-out.
- We use an activation decay of 0.1: After each pulse, the residual activation of a node is reduced to one tenth before adding it to the incoming activation energy. This factor limits the effect of the initial activation.

Because the spreading activation algorithm does not distinguish between node types, we use custom parameters in two of the five iterations:

- During the first pulse, we use an activation decay of 0.0. At this time, only the query node is activated, and activation spreads from the query node to the initial document nodes. By using an activation decay of 0.0, we ensure that the query node's activation is 0.0 after the first pulse, and that it does not re-activate the initial document nodes.
- During the third pulse, we constrain spreading to edges between author nodes. At this stage, activation has arrived at author nodes, and activation energy from multiple messages by the same author has been accumulated. By constraining spread to author nodes in this activation, we emphasize the importance of the social network.

After the fourth pulse, the initially activated documents are returned, sorted by their activation level.

The described adjustments and constraints are chosen to express an intuitive notion of social search. Different applications will require a different set of constraints. The large number of possible constraints and parameters make it infeasibel to search through this configuration space in a systematic manner. Small changes in parameters or the stage at which adjustments and constraints are applied can have a profound effect on the resulting activation levels. Whether other parameter sets for social search exist with a similar or better performance is subject to extensive experimentation.

6.4.2 Example

A schematic depiction of the spread of activation through the associative network is in figure 6.3. The domain is similar to the one described in the example in section 5.1: It contains five individuals $i_1, \ldots, i_5$ and five documents $d_1, \ldots, d_5$. The first individual i_1 has social ties with the next three individuals i_2, i_3 and i_4, whereas i_5 has no social ties with another individual. A query node q points to all five documents, and every document is associated with its author.

We see that activation spreads from the query node to the document nodes in the first iteration, after which the activation of the query node drops to zero. After the second pulse, activation arrives at author nodes; in the third iteration, document nodes retain part of their activation level, while activation accumulates in the dominant nodes of the social network. In the fourth iteration, this accumulated activation is spread back to the document nodes.

As described in the last subsection, we use an activation adjustment in the post-adjustment phase of $f_a^{2-4}(x) = 0.1x$. In the first iteration, we use $f_a^1(x) \equiv 0$. The distance constraint is not active in the example network, as all nodes have a distance of two or less from the query node.

- Initially, the only activated node is the query node q with an activation energy of 100:

$$a_q^0 = 100$$

- In the first iteration, the full activation energy is spread from the query node q to the document nodes $d_1, \ldots, d_5$. No preadjustments are active, causing the document nodes to receive the full activation energy. Because an activation decay of 0.0 is active in this iteration, the activation of the query node drops to zero, causing it to become deactivated:

$$\begin{aligned} a_q^1 &= f_a^1(a_q^0) = 0 \\ a_{d_1}^1 &= a_q^0 = 100 \\ &\vdots \\ a_{d_5}^1 &= a_q^0 = 100 \end{aligned}$$

- In the second iteration, the energy of the document nodes $d_1, \ldots, d_5$ is spread to their respective author nodes $i_1, \ldots, i_5$. Because an activation decay of 0.1 is

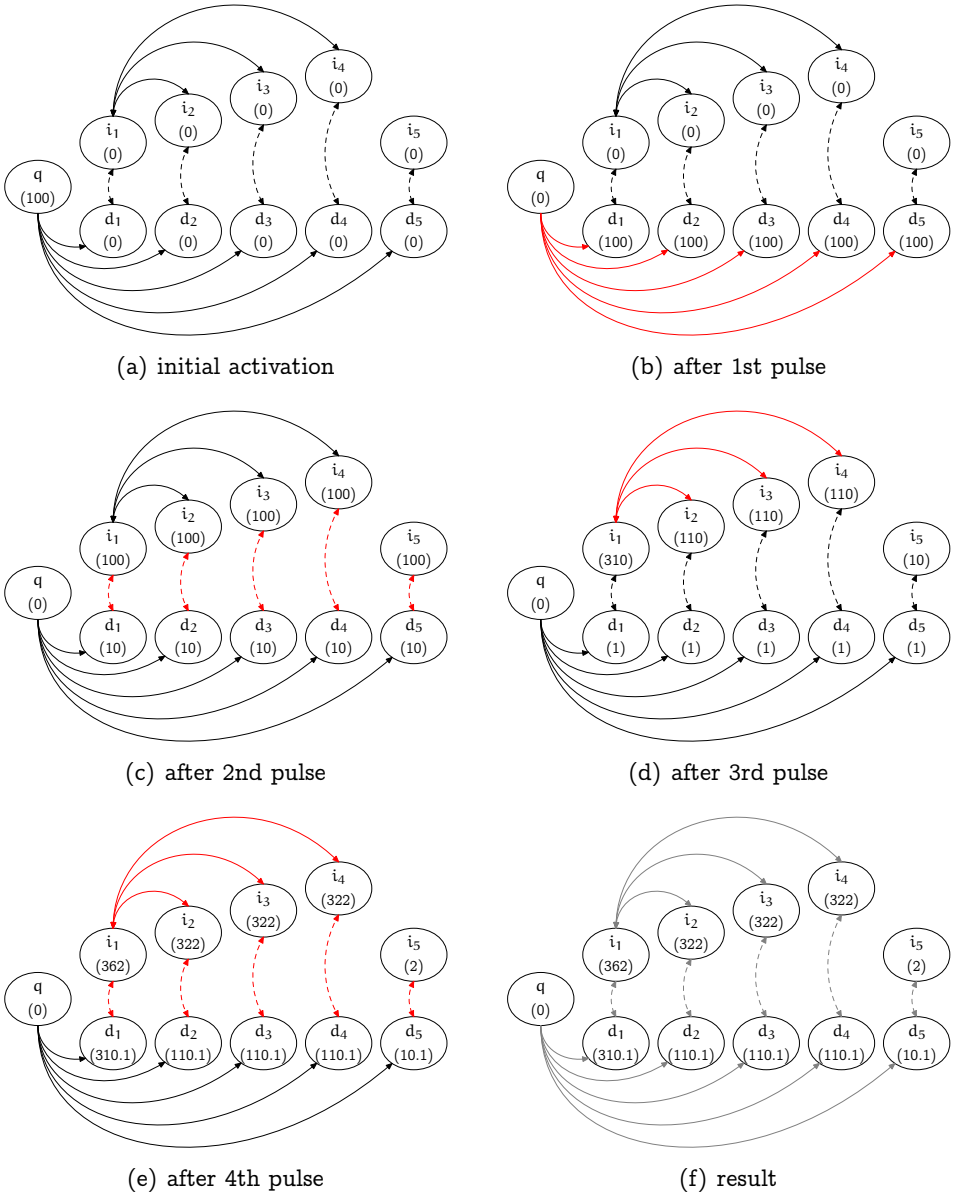


Figure 6.3: Schematic depiction of activation spread through the associative network. Numbers in parentheses are the activation level of the node in this iteration; red arrows signify activated edges.

active, the energy of the document nodes drops to one tenth of its value:

$$\begin{aligned}
a^2_{i_1} &= a^1_{d_1} = 100 \\
&\vdots \\
a^2_{i_5} &= a^1_{d_5} = 100 \\
a^2_{d_1} &= f^2_a(a^1_{d_1}) = 0.1 \cdot 100 = 10 \\
&\vdots \\
a^2_{d_5} &= f^2_a(a^1_{d_5}) = 0.1 \cdot 100 = 10
\end{aligned}$$

- In the third iteration, spread is constrained to author nodes: i_1 receives additional activation from i_2, i_3 and i_4, in addition to its initial activation of a_{i_4} reduced to one tenth. The activation of document nodes drops to one tenth, as does the activation of all other nodes:

$$\begin{aligned}
a^3_{i_1} &= a^2_{i_2} + a^2_{i_2} + a^2_{i_2} + f^3_a(a^2_{i_1}) \\
&= 100 + 100 + 100 + 0.1 \cdot 100 = 310 \\
a^3_{i_2} &= a^2_{i_1} + f^3_a(a^2_{i_2}) = 100 + 0.1 \cdot 100 = 110 \\
&\vdots \\
a^3_{i_4} &= a^2_{i_1} + f^3_a(a^2_{i_4}) = 100 + 0.1 \cdot 100 = 110 \\
a^3_{i_5} &= f^3_a(a^2_{i_5}) = 0.1 \cdot 100 = 10 \\
a^3_{d_1} &= f^3_a(a^2_{d_1}) = 0.1 \cdot 10 = 1 \\
&\vdots \\
a^3_{d_5} &= f^3_a(a^2_{d_5}) = 0.1 \cdot 10 = 1
\end{aligned}$$

- In the last iteration, energy is spread from the author nodes back to the document

nodes. Author nodes also receive activation from document nodes:

$$
\begin{aligned}
a^4_{i_1} &= a^3_{i_2} + a^3_{i_3} + a^3_{i_4} + a^3_{d_1} + f^4_a(a^3_{i_1}) = 110 + 110 + 110 + 1 + 0.1 \cdot 310 = 362 \\
a^4_{i_2} &= a^3_{i_1} + a^3_{d_2} + f^4_a(a^3_{i_2}) = 310 + 1 + 0.1 \cdot 110 = 322 \\
&\vdots \\
a^4_{i_4} &= a^3_{i_1} + a^3_{d_4} + f^4_a(a^3_{i_4}) = 310 + 1 + 0.1 \cdot 110 = 322 \\
a^4_{i_5} &= a^3_{d_5} + f^4_a(a^3_{i_5}) = 1 + 0.1 \cdot 10 = 2 \\
a^4_{d_1} &= a^3_{i_1} + f^4_a(a^3_{d_1}) = 310 + 0.1 \cdot 1 = 310.1 \\
a^4_{d_2} &= a^3_{i_2} + f^4_a(a^3_{d_2}) = 110 + 0.1 \cdot 1 = 110.1 \\
&\vdots \\
a^4_{d_4} &= a^3_{i_4} + f^4_a(a^3_{d_4}) = 110 + 0.1 \cdot 1 = 110.1 \\
a^4_{d_5} &= a^3_{i_5} + f^4_a(a^3_{d_5}) = 10 + 0.1 \cdot 1 = 10.1
\end{aligned}
$$

We see that spreading activation search achieves our desired result of promoting authors with many social links, while penalizing solitary authors.

6.5 Summary

We describe an associative network model for one concrete information retrieval task, namely keyword-based retrieval on a domain where author information and a social network between authors is available. Vector-space retrieval is used as the underlying text retrieval method, and is used as a baseline performance measure in evaluation.

Two techniques are described, one based on a global authority measure for the social network, and one based on exploring local links in the associative network.

The global technique is based on the PageRank authority measure; PageRank scores are computed for the social network and combined with relevance scores from vector-space retrieval to determine the ranking of results. The suitability of PageRank as a measure of authority is demonstrated on two example networks.

Spreading activation search is used as a local technique; it is based on exploring the social neighbourhood of a relevant document's author. An intuitive method of assessing a document's relevance based on the author's social network is given. This method is implemented as a set of constraints and adjustments for spreading activation search.

Chapter 7

Evaluation

In this section, we evaluate the effectiveness of social retrieval techniques, as described in chapter 6, in comparison with conventional retrieval techniques. Due to the absence of standard corpora with suitable characteristics, we use two locally compiled corpora.

We evaluate the techniques in a known-item retrieval setting and compare them to the baseline technique described in section 6.2 using the metrics average rank and inverse average inverse rank as in section 3.7.2. Evaluation based on the precision and recall metrics as in section 3.7.1 requires labour-intensive screening of the complete corpora, as well as the collaboration of several experts in the domain of the corpora. In comparison, a known-item retrieval setting reduces the amount of manual labour required and allows a semi-automatic selection of items, as described in the following sections.

By comparing with a baseline technique on the same index, we eliminate external factors that may account for differences in performance; this allows us to gauge the impact of social retrieval techniques on retrieval performance.

7.1 Corpora

The domain model for social information retrieval as in figure 6.1 requires that a social network between individuals is present in the evaluation corpus, as well as associations between individuals and documents. In this section, we describe two corpora that satisfy these requirements and which are used for evaluating the effectiveness of social IR techniques. We explain how a full-text index is constructed and how the social network between the authors of the document is extracted. We also include statistical characteristics of the corpora and of the extracted social networks. We examine whether the social networks display the expected characteristics from statistical network analysis as described in section 3.4.

7.1.1 Mailing List Archives

The mailing list corpus contains messages from the 'origami-l' mailing list[1] archive from the years 1997–2005 and was collected by the author. The full source of each message

[1] http://origami.kvi.nl/, last visit on 2005/05/02.

in RFC 822 format (Crocker, 1982) is available. For evaluation, two different subsets of the corpus are used, one containing messages from 2000–2005, and one from 2004.

For the full-text index, the following strategy is used:

1. Both the message body and the `Subject:` line are included in the full-text index. When choosing evaluation queries as in section 7.2.1, this ensures that the desired document for known-item retrieval is found in any case.

2. Heuristics are used to detect common types of markup for signatures and quoted text; these parts are removed. This step ensures that only content actually produced by the author of the messages is included in the full-text index.

3. Remaining content is tokenized and lowercased.

4. Stopwords are removed, using a stopword list by Jacques Savoy[2].

5. For statistical purposes, bi- and trigrams are extracted; they are not used for searching.

In addition to the full-text index, an associative network is constructed from the messages:

- An author node is constructed for each email address. No effort is made to reconcile different email addresses of one person.

- Every message is linked to its author, and every author is linked to his messages.

- Messages are linked to their follow-ups, and vice-versa. Whether a message is a follow-up to another is determined from the `In-Reply-To:` and `References:` header lines. No attempt is made to match messages to their follow-ups by textual means.

- Authors are linked to each other based on how often they respond to one another's messages.

Statistics of the two subsets are listed in table 7.1; the degree distribution for the social network is in figure 7.1. When calculating statistics for the social network, we use the underlying undirected graph, ie. we treat all social links as undirected links. This is in accordance with the usual techniques in social network analysis, which are mostly concerned with undirected graphs.

The social networks extracted from the corpus share typical characteristics with other social networks examined by Newman (2001). The giant connected components comprise about 70% of all nodes; less than the more than 90% commonly cited for the network of movie actors or the coauthorship network for the MEDLINE database, but on par with

[2] http://www.unine.ch/info/clef/englishST.txt, last visit on 2005/08/17.

Table 7.1: Statistical characteristics of the 'origami-l' corpus.

	2000–2005	2004
no. of messages	44108	4411
no. of email addresses	1834	464
mean neighbours per address	7.959	4.838
exponent γ	1.093	1.078
size of giant connected component (GCC)	1271	331
as percentage [%]	69.3	71.3
size of next-largest component	2	2
average shortest path length in GCC	2.983	3.108
diameter of GCC	9	6
mean clustering coefficient	0.647	0.578

smaller networks. The size of the next-largest weak components is very small compared to the size of the largest component. The average shortest path length in the giant connected component is very low at about three, and the diameters are 6 for the smaller corpus and 9 for the larger corpus. This makes the social network of the mailing list corpus a very small world.

The degree distribution seems to follow a power law with an exponent of $\gamma \approx 1.1$, similar to smaller coauthorship networks surveyed by Newman (2001). A graphical plot of the degree distribution is in figures 7.1 and 7.2. The regression curves were fitted to the data using nonlinear least squares regression.

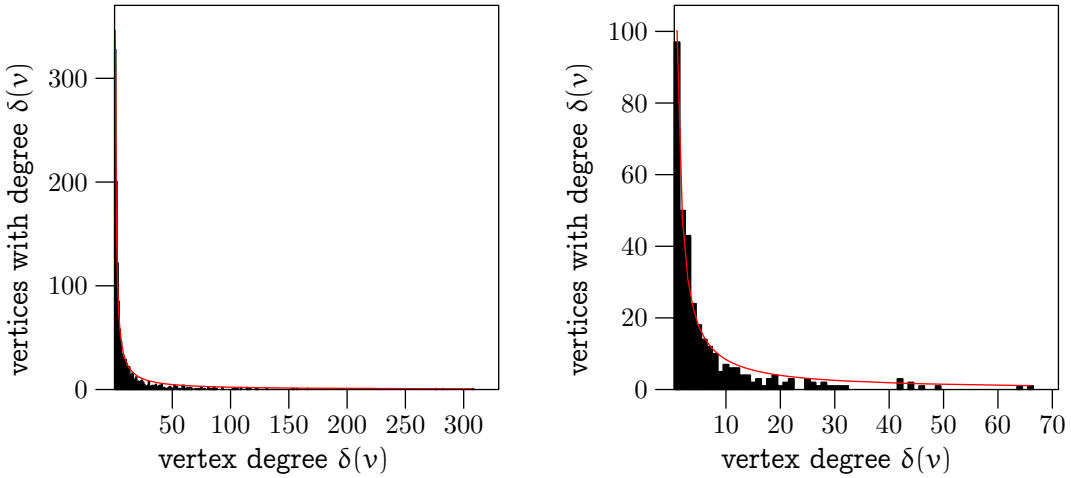
7.1.2 SIGIR Corpus

The 'SIGIR corpus' is a collection of conference proceedings of the annual ACM SIGIR (Special Interest Group on Information Retrieval) conference from 1978–2003.

This corpus contains author and title information about every document published in the proceedings of the SIGIR conference, as well as a full-text index. References to other documents in the corpus are also present. The database containing author names, titles, abstracts and year of publication was originally prepared for Smeaton et al. (2002) and was graciously provided by the authors. It was enhanced locally; citation information was extracted from the full text of the documents using information extraction techniques.

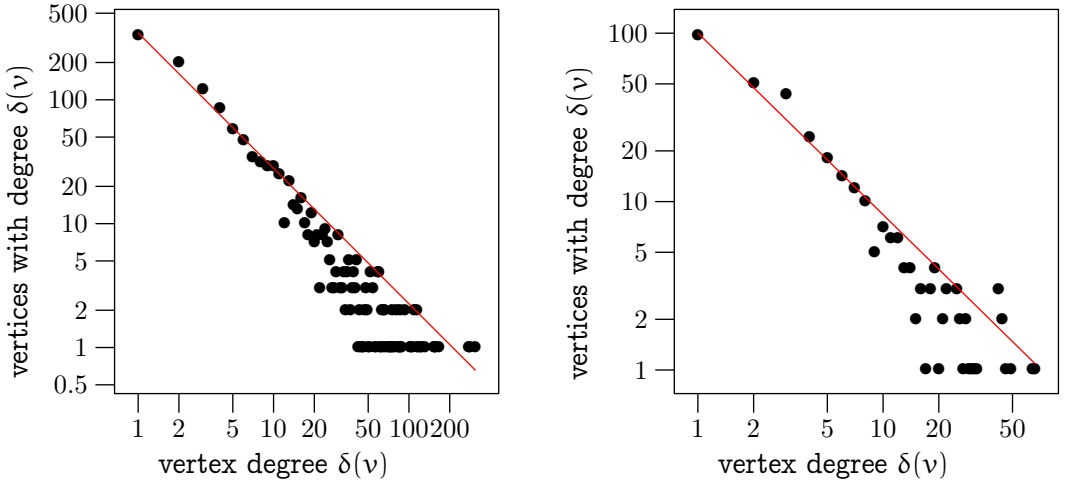
For the full-text index, electronic versions of the proceedings (available in PDF format) are converted to plain text. Plain text files are tokenized and lowercased; stopwords are removed as in section 7.1.1. Titles and abstracts are retrieved from the database and added to separate fields of the full-text index.

The social network of this corpus is the coauthorship network: Two authors are presumed to have a social relation if they authored a publication together. This method of



(a) 2000–2005 (b) 2004

Figure 7.1: The distribution of vertex degrees for the social network of the 'origami-l' corpus. Vertices with degree $\delta(v) = 0$ are omitted in the graph. The red line is a regression curve for the power-law distribution $\Pr(\delta(v) = k) \sim k^{-\gamma}$ with γ as in table 7.1.



(a) 2000–2005 (b) 2004

Figure 7.2: The distribution of vertex degrees for the social network of the 'origami-l' corpus, plotted on a log-log scale; again, vertices with degree $\delta(v) = 0$ were omitted. The logarithmic scale makes it evident that the graph follows a power-law distribution, as the regression curve becomes a straight line. The slope of the regression curve is the same as the exponent γ in table 7.1.

Table 7.2: Statistical characteristics of the SIGIR corpus.

no. of documents	1041
no. of authors	1397
mean collaborators per author	2.863
exponent γ	—
size of giant connected component (GCC)	312
as percentage [%]	22.3
size of next-largest component	146
average shortest path length in GCC	6.303
diameter of GCC	16
mean clustering coefficient	0.902

constructing the social network implies that the network is undirected.

As can be seen from the statistics for the SIGIR corpus in table 7.2, the corpus is rather small at just over one thousand documents; furthermore, it contains more authors than documents. Each author has on average less than three collaborators. The giant connected component is fairly small, comprising 22% of the coauthorship graph; this figure is markedly lower than the corresponding figure for the mailing list corpus, and also lower than the figures reported by Newman (2001). We presume that this is due to the small size of the corpus. Average shortest path length and diameter of the giant connected component are higher than the figures for the mailing list corpus, but comparable to figures reported for larger coauthorship networks.

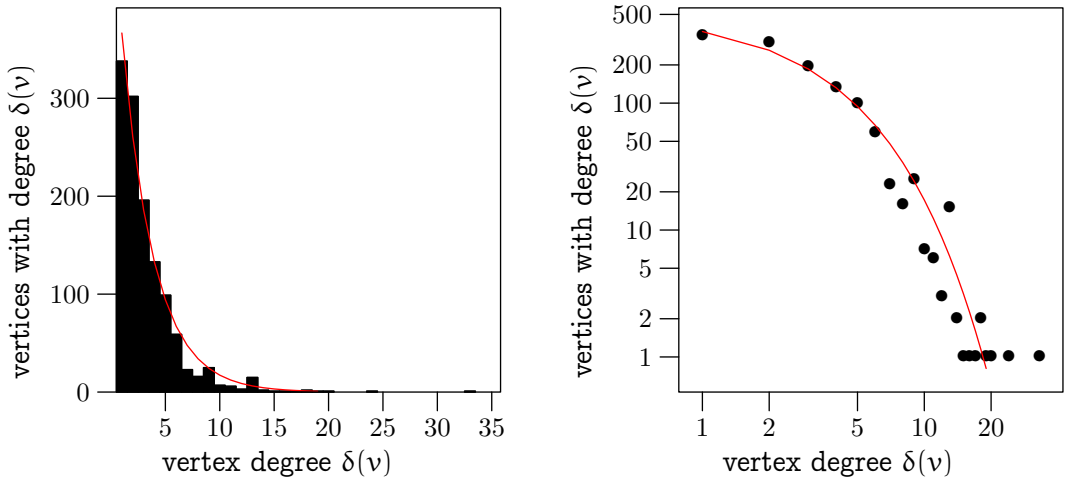
The degree distribution of the social network does not appear to follow a power law; instead, the probability of a vertex having degree k appears closer to

$$\Pr(\delta(v) = k) \sim \exp\left(-\frac{k}{k_c}\right)$$

where k_c is a constant. This may result from the small size of the corpus: Newman (2001) observed degree distributions for smaller social networks that were closer to a power law with an exponential cutoff, that is

$$\Pr(\delta(v) = k) \sim k^{-\gamma} \exp\left(-\frac{k}{k_c}\right);$$

He speculates that it is a result of the underlying distribution following a power law, with an external constraint that limits the maximum degree of a node. In our case, the constraint arises from the limited time frame and the small number of documents: An author can only have a limited number of publications in the SIGIR proceedings, and thus can only collaborate with a limited number of other individuals. As can be seen from figure 7.3, the degree distribution can be adequately explained using exponential decay (with $k_c \approx 2.9$); the power law hardly seems to affect the distribution.



(a) linear scale

(b) log-log scale

Figure 7.3: The degree distribution of the SIGIR corpus does not appear to follow a power law; it seems closer to an exponential distribution. Vertices with degree $\delta(v) = 0$ are omitted in the graph. The red line is a regression line for an exponential distribution $\Pr(\delta(v) = k) \sim \exp(-\frac{k}{k_c})$ with $k_c \approx 2.9$.

7.2 Methodology for Choosing Search Queries

Choosing representative search queries and relevant documents is a central part of the known-item retrieval scenario; it is usually performed by experts in the subject matter with a reasonably complete knowledge of the documents in the corpus. We extract known items and search queries in a semi-automatic manner, due to a limited amount of manpower available for the evaluation. Since objective criteria are used for choosing search queries, we prevent a personal bias from affecting the evaluation results. Where a human judgement is necessary, two different experts choose relevant documents independent from each other.

7.2.1 Mailing List Archives

For choosing appropriate query terms for known-item retrieval in the case of mailing list archives, the following strategy is used:

From the `Subject:` lines of email messages, frequent bi- and trigrams are extracted. `Subject:` lines are a good indicator of user information needs, as many threads on a mailing list start with a question, and the question is usually summarized in the subject. Bi- and trigrams are especially apt candidates, because 'real-world' queries have been found to average between two and three words (Silverstein et al., 1999).

Selecting n-grams by frequency alone is sub-optimal, as some frequent n-grams correlate highly with the author of the containing messages (for example, periodic announcements usually contain the same `Subject:` line and are by the same author.) In order to remove these n-grams, the mutual information of the occurence of a specific n-gram in the `Subject:` line and the author of the messages is determined.

Mutual Information, also called **information gain** in the context of machine learning, measures the amount of information shared by two random variables. The mutual information of two random variables X and Y is usually defined as

$$I(X,Y) = \sum_x \sum_y \Pr(X=x, Y=y) \log_2 \frac{\Pr(X=x, Y=y)}{\Pr(X=x)\Pr(Y=y)}$$

An equivalent definition is (Hamming, 1980)

$$I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

where the entropy of X is

$$H(X) = -\sum_x \Pr(X=x) \log_2 \Pr(X=x)$$

and the conditional entropy of X given Y is

$$H(X|Y) = \sum_y \Pr(Y=y) H(X|Y=y)$$

A high mutual information between the occurrence of a specific n-gram and the author of the containing messages is an indicator for an idiom that is used exclusively by few authors, and is not a good query phrase for evaluation. A desirable n-gram for use as a query phrase therefore has a low mutual information with the author, and a high document frequency at the same time. We sort n-grams by information gain divided by the frequency and use the n-grams with the lowest score for evaluation:

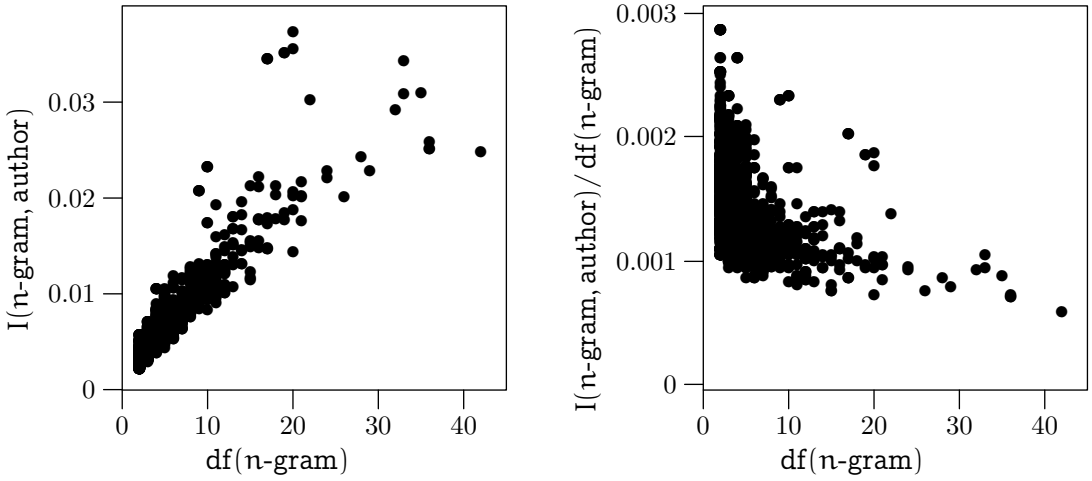
$$\text{score}(\text{n-gram}) = \frac{I(\text{n-gram}, \text{author})}{\text{df}(\text{n-gram})} \tag{7.1}$$

Figure 7.2.1 shows the correlation between messages containing a specific n-gram and the author of the messages, as regards the document frequency. As the document frequency decreases, the correlation decreases as well, since distributions become more ordered. In the figure, one can discern a number of n-grams that have an unusually high correlation to one specific author.

Table 7.3 lists the n-grams with the lowest score for the `Subject:` lines, for the 'origami-l' corpus for 2004. Terms printed in italics are chosen as query terms for known-item retrieval. In the case of overlap between n-grams, the longest n-gram is chosen.

Table 7.3: n-grams from `Subject:` headers, sorted by score. Terms in italics are selected as query terms for known-item retrieval. (data: 'origami-l' archives for 2004, scores as in equation 7.1.)

n-gram	df(n-gram)	score(n-gram) $\times 10^{-4}$
origami sighting	98	5.148
crease patterns	42	5.884
5 favorite	36	6.966
favorite models	36	6.966
5 favorite models	36	6.966
rose polygon	20	7.143
art craft	36	7.158
roses project	15	7.618
cp short	26	7.692
cp short cruel	26	7.692
current model	15	7.704
favorite current	15	7.704
favorite current model	15	7.704
cruel punishment	29	7.842
short cruel	29	7.842
short cruel punishment	29	7.842
collector accumulator	15	8.122
folding clothes	11	8.161
tension folding	13	8.199
rolling ball	10	8.214
identify image	12	8.321
nick robinson	21	8.326
teaching origami	12	8.341
lang origami	11	8.595
fish model	17	8.612



(a) information gain vs. doc. frequency

(b) information gain divided by doc. frequency vs. doc. frequency

Figure 7.4: Measuring the correlation between the occurence of a specific n-gram in the subject of emails and the author of the message. (data: 'origami-l' archives for 2004)

For each of the ten queries, one message is chosen as the 'known item', the objective of this search: Only messages from 2004 are considered as relevant, and only those messages are assessed that actually contain the sequence of query terms in the `Subject:` line. The criteria for relevance are selected to mimic a searcher looking for an item he has seen before: He would probably remember the subject of the message, and from the pool of messages with a matching subject, the most memorable one is chosen.

The items to be retrieved are chosen by the author, who is an expert in the subject matter, and by a complete novice as regards paperfolding. Using two different relevance assessments allows us to evaluate whether a social IR system caters more to novice users who desire more general results of high quality, but know next to nothing about the authors, or expert users who may have more specific interests, and can judge a person's authority within the community without assistance of the social IR system.

7.2.2 SIGIR Corpus

Due to the small size of the SIGIR corpus, the statistical approach for choosing search queries used in the section 7.2.1 is not applicable. Instead, we use the following approach:

We determine the number of citations for a document in the corpus and use this as a measure of importance of a document within the corpus. This is a very simple method of citation analysis; it ignores important factors such as citations outside the corpus or the publication date of the cited document and the citing documents. The result are

Table 7.4: The ten most-cited documents in the SIGIR corpus, with query phrases derived from the document title

document	query phrase	citations
A language modeling approach to information retrieval (Ponte and Croft, 1998)	language modeling	13
Reexamining the cluster hypothesis: scatter/gather on retrieval results (Hearst and Pedersen, 1996)	scatter gather	11
A hidden Markov model information retrieval system (Miller et al., 1999)	hidden markov model	11
Relevance feedback revisited (Harman, 1992)	relevance feedback	11
Pivoted document length normalization (Singhal et al., 1996)	document length normalization	11
Automatic phrase indexing for document retrieval (Fagan, 1987)	automatic phrase indexing	10
Information retrieval as statistical translation (Berger and Lafferty, 1999)	statistical translation	10
Inference networks for document retrieval (Turtle and Croft, 1990)	inference networks	9
Probabilistic Models of Indexing and Searching (Robertson et al., 1981)	probabilistic models	9
Towards interactive query expansion (Harman, 1988)	interactive query expansion	8

further biased by inadequacies in the information extraction methods used to extract the citations.

We select the ten most-cited documents as the most influential (or authoritative) documents in the corpus, and use those as 'known items' in a known-item retrieval setting. Query phrases are determined from the title of the publication. Table 7.4 lists the selected documents, as well as query phrases and the number of citations. The most-cited publications include documents from 1981–1999; we surmise that newer documents have not been available long enough to acquire a significant number of citations.

7.3 Evaluation Tasks

This section contains results from known-item retrieval experiments on the two corpora described in section 7.1, using queries and documents chosen as described in section 7.2.

Rankings are not necessarily unique, as they rely on sorting according to a numerical score. In case of ambiguities, we report the ranks and derived metrics as intervals.

7.3.1 Known-item Retrieval on Mailing List Data

Detailed results from these experiments are in tables 7.5 and 7.6.

For items chosen by an expert searcher, the combination of PageRank and the vector-space model performs better than the vector space model alone for four of ten queries on the 2004 corpus; in one case, the result is a draw. While the average rank of the found documents increases for PageRank search, the inverse average inverse rank decreases: The average rank increases by 21.7%2.4, but the inverse average inverse rank decreases by $6.2\% \pm 0.5$. This means that some documents are found considerably later than with vector-space search, but for those documents in the earlier parts of the result list, PageRank combined with vector space performs better. This effect is even more pronounced on the 2000–2005 corpus, where the average rank increases by $69.9\% \pm 2.3$, but the inverse average inverse rank decreases by $24.6\% \pm 0.5$. On the 2000–2005 corpus, the combination performs better for six out of ten queries.

For the novice searcher, results are less pronounced. On the smaller corpus from 2004, both the average rank and inverse average inverse rank decrease (average rank by $13.1\% \pm 1.5$, IAIR by $1.5\% \pm 0.3$), whereas on the larger corpus, the average rank is identical the same, but the IAIR increases sharply (by $58.4\% \pm 0.4$.) On the smaller corpus, PageRank times vector space performs better for five out of ten queries, with one draw; for the larger corpus, it performs better for four out of ten queries, also with one draw.

This mirrors the results from Page et al. (1999), who report that 'the benefits of PageRank are the greatest for underspecified queries' and that 'for more specific searches where recall is more important, the traditional information retrieval scores and the PageRank should be combined.' The very nature of the known-item retrieval task places an emphasis on recall, since the objective is finding one *specific* document instead of just one of several that satisfy the information need.

Spreading activation search shows a clear improval on the smaller corpus with messages from 2004: Using the combination of adjustments and constraints described in section 6.4 lowers the inverse average inverse rank by $49.5\% \pm 0.3$ for the expert searcher, and by $18.0\% \pm 0.2$ for the novice searcher. Average rank increases by $6.8\% \pm 2.1$ for the expert searcher and decreases by $29.1\% \pm 1.2$ for the novice searcher. Development of average rank and inverse average inverse rank as compared to the baseline method mirrors the trends found for social search using PageRank.

In difference to our results for social search with PageRank, the trends for spreading activation search do not carry over to the larger corpus with messages from 2000–2005. For the expert searcher, the average rank and inverse average inverse rank double when using spreading activation search (average rank 44.85 ± 0.05 versus 24.4 ± 0.3 for the

baseline method, inverse average inverse rank 14.018 ± 0.089 versus 8.787 ± 0.040 for the baseline method.) Comparable results are achieved for the novice searcher (average rank 52.9 versus 39.35 ± 0.35 for the baseline, inverse average inverse rank 14.358 versus 4.962 ± 0.013 for the baseline.)

As noted by Crestani (1997), the effectiveness of spreading activation search depends crucially on the structure of the associative network. In particular, nodes with a high degree, which are found more frequently in the larger subset of the mailing list corpus, often need special treatment. Further experiments are needed to determine suitable procedures for the treatment of nodes with a high degree in the social network.

7.3.2 Known-item Retrieval on the SIGIR Corpus

Known-item retrieval on the SIGIR is performed using the documents and query terms from table 7.4; detailed results are found in table 7.7. Three different scenarios are evaluated, using terms from the title of the documents, terms from the abstracts, and a search over the full text of the documents. When searching the abstracts, the desired document is not found in two out of ten cases (denoted by '—' in the corresponding cells of the result table), because the abstract does not contain the query terms taken from the document title.

No improvement can be detected in any of the scenarios when comparing social search using PageRank to the baseline method: at best, the average rank remains the same, at worst, it increases by 138.4%, while the inverse average inverse increases by at least $20.3\% \pm 20.5$ and at most 41.4% when using social search. Similar results were achieved using spreading activation search.

When reviewing the result lists for the experiments on the full-text index, we find that in all cases where the known item is not at the top of the result list, there is a publication by W. Bruce Croft at the top – who has the highest PageRank in the social network, as we see from table 6.1. In other words, we seem to have constructed a highly effective search engine for finding publications by W. Bruce Croft.

One interpretation for this phenomenon is that PageRank identifies hubs in the social network. Hubs are important to the network in their role as 'multipliers' or disseminators of information. Bruce Croft is an example of a very successful multiplier: He co-authored thirty-six publication in twenty-five years of conference proceedings, and collaborated with thirty-three authors on a large variety of topics.

By selecting documents based on how often they are cited, we bias the desired documents away from multipliers and towards innovators: When selecting which publication to cite, one often goes back to the work originally introducing an idea, neglecting to cite subsequent work that popularized the topic.

Indeed, Granovetter (1973) notes that innovators are often at the margin of the social network, because they do not conform to the norms of the community – which may be a trait that allows them to innovate. The early adopters of a new idea however are

Table 7.5: Known-item retrieval on mailing list data, using messages from 2004. Columns labelled 'VS' contain ranks from vector-space search as in section 6.2, columns labelled 'PR×VS' contain ranks scored by pagerank times vector space score as in equation 6.2 in section 6.3, and columns labelled 'SA' contain ranks using spreading activation search as detailed in section 6.4. Rows '$\overline{\text{rank}}$ change' and 'IAIR change' contain the change compared to the baseline method 'VS' in percent.

method:	VS	PR×VS	SA	VS	PR×VS	SA
searcher:	expert	expert	expert	novice	novice	novice
crease patterns	16 ± 0	44 ± 0	40 ± 0	37.5 ± 0.5	24 ± 0	36 ± 0
5 favourite models	29 ± 0	25 ± 0	11 ± 0	34.5 ± 0.5	34 ± 0	13 ± 0
rose polygon	5 ± 0	4 ± 0	5 ± 0	5 ± 0	4 ± 0	5 ± 0
art craft	31.5 ± 0.5	25 ± 0	35 ± 0	45 ± 0	35 ± 0	31 ± 0
roses project	13.5 ± 0.5	15.5 ± 0.5	9.5 ± 0.5	1 ± 0	1 ± 0	11 ± 0
favourite current model	8 ± 0	14 ± 0	14 ± 0	8 ± 0	14 ± 0	14 ± 0
short cruel punishment	25.5 ± 1.5	29 ± 0	24 ± 0	20 ± 2	17 ± 0	2 ± 0
collector accumulator	6 ± 0	3 ± 0	1 ± 0	9 ± 0	12 ± 0	9 ± 0
folding clothes	2 ± 0	2 ± 0	1 ± 0	2 ± 0	2 ± 0	1 ± 0
tension folding	11 ± 0	18 ± 0	17 ± 0	13 ± 0	9 ± 0 2	$0 \pm$
$\overline{\text{rank}}$:	14.75 ± 0.25	17.95 ± 0.05	15.75 ± 0.05	17.5 ± 0.3	15.2 ± 0	12.4 ± 0
$\overline{\text{rank}}$ change [%]:		$+21.7 \pm 2.4$	$+6.8 \pm 2.1$		-13.1 ± 1.5	-29.1 ± 1.2
IAIR:	7.548 ± 0.032	7.082 ± 0.010	3.814 ± 0.008	4.670 ± 0.013	4.599 ± 0	3.831 ± 0
IAIR change [%]:		-6.2 ± 0.5	-49.5 ± 0.3		-1.5 ± 0.3	-18.0 ± 0.2

Table 7.6: Known-item retrieval on mailing list data, using messages from 2000–2005. Columns labelled 'VS' contain ranks from vector-space search as in section 6.2, columns labelled 'PR×VS' contain ranks scored by pagerank times vector space score as in equation 6.2 in section 6.3. Rows '$\overline{\text{rank}}$ change' and 'IAIR change' contain the change compared to the baseline method 'VS' in percent.

method:	VS	PR×VS	VS	PR×VS
searcher:	expert	expert	novice	novice
crease patterns	71 ± 0	279 ± 0	167 ± 1	145 ± 0
5 favourite models	34.5 ± 0.5	29 ± 0	48.5 ± 0.5	51 ± 0
rose polygon	5 ± 0	3 ± 0	5 ± 0	3 ± 0
art craft	40.5 ± 0.5	11 ± 0	89 ± 0	117 ± 0
roses project	15.5 ± 0.5	13.5 ± 0.5	1 ± 0	3 ± 0
favourite current model	8 ± 0	22 ± 0	8 ± 0	22 ± 0
short cruel punishment	25.5 ± 1.5	29 ± 0	20 ± 2	12 ± 0
collector accumulator	9 ± 0	2 ± 0	13 ± 0	13 ± 0
folding clothes	2 ± 0	3 ± 0	2 ± 0	3 ± 0
tension folding	33 ± 0	23 ± 0	40 ± 0	27 ± 0
$\overline{\text{rank}}$:	24.4 ± 0.3	41.45 ± 0.05	39.35 ± 0.35	39.6 ± 0
$\overline{\text{rank}}$ change [%]:		$+69.9 \pm 2.3$		$+0.6 \pm 0.9$
IAIR:	8.787 ± 0.040	6.697 ± 0.012	4.962 ± 0.013	7.86 ± 0
IAIR change [%]:		-24.6 ± 0.5		$+58.4 \pm 0.4$

Table 7.7: Known-item retrieval on the SIGIR corpus. Columns labelled 'VS' contain ranks from vector-space search as in section 6.2, columns labelled 'PR×VS' contain ranks scored by pagerank times vector space score as in equation 6.2 in section 6.3. Rows '$\overline{\text{rank}}$ change' and 'IAIR change' contain the change compared to the baseline method 'VS' in percent.

method:	VS	PR×VS	VS	PR×VS	VS	PR×VS
source:	title	title	abstract	abstract	fulltext	fulltext
language modeling	1 ± 0	1 ± 0	12 ± 0	3 ± 0	13 ± 0	3 ± 0
scatter gather	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0	2 ± 0
hidden markov model	1.5 ± 0.5	1 ± 0	1 ± 0	3 ± 0	7 ± 0	16 ± 0
relevance feedback	2.5 ± 1.5	10 ± 0	1 ± 0	6 ± 0	23 ± 0	105 ± 0
document length normalization	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0
automatic phrase indexing	1 ± 0	1 ± 0	1 ± 0	3 ± 0	1 ± 0	38 ± 0
statistical translation	1 ± 1	2 ± 0	2 ± 0	2 ± 0	14 ± 0	18 ± 0
inference networks	2 ± 1	1.5 ± 0.5	1 ± 0	1 ± 0	2 ± 0	1 ± 0
probabilistic models	1.5 ± 0.5	2 ± 0	—	—	45 ± 0	49 ± 0
interactive query expansion	1 ± 0	2 ± 0	—	—	5 ± 0	34 ± 0
$\overline{\text{rank}}$:	1.35 ± 0.35	2.25 ± 0.05	2.5 ± 0	2.5 ± 0	11.2 ± 0	26.7 ± 0
$\overline{\text{rank}}$ change [%]:		$+79.7 \pm 50.3$		0 ± 0		$+138.4 \pm 0$
IAIR:	1.159 ± 0.159	1.362 ± 0.046	1.215 ± 0	1.714 ± 0	2.465 ± 0	3.293 ± 0
IAIR change [%]:		$+20.3 \pm 20.5$		$+41.1 \pm 0$		$+33.6 \pm 0$

well-connected individuals at the center of the social network: If no hub in the network lends authority to a new idea and serves as a multiplier, it is unlikely to spread through the social network at all.

Examining the first-ranked documents, we find that in five out of eight times, the desired document is listed in the bibliography; in one case, there is a citation trail of length two between the first-ranked document and the desired document. This substantiates our claim that the highest-ranking individuals serve as disseminators of information.

We conclude that social IR is not applicable in this evaluation setting. There are several reasons limiting the effectivity of social IR in this setting, caused both by characteristics of the corpus and by the evaluation methodology: It is widely believed that the benefits of link analysis for information retrieval are greatest for underspecified queries, combined with a large document collection containing many relevant documents that differ widely in quality. Both conditions are violated by the SIGIR corpus, since it is very small, focused on a narrow domain, and contains only high-quality documents. The queries are not under-specified, but are chosen to match one specific document. The methodology of selecting known documents is biased towards innovators, whereas the social retrieval techniques are biased towards multipliers.

7.4 Summary

Two techniques for social search, one based on PageRank and one based on spreading activation search, are compared to conventional vector-space search in a known-item retrieval task. Evaluation is carried out on two corpora: A mailing list archive containg messages from the years 2000–2005 from the 'origami-l' mailing list, and a set of publications from the proceedings of the ACM SIGIR conference from 1978–2003.

Query phrases for evaluation on the mailing list corpus are derived from frequent bi- and trigrams in the `Subject:` lines of the messages; known items are selected by human experts. For the corpus of conference proceedings, the most-cited documents in the corpus are selected as known items; query phrases are derived from their titles.

Two evaluation metrics are used for comparing the performance of the retrieval methods: average rank and inverse average inverse rank, as in section 3.7.2.

On the mailing list corpus, the social retrieval method based on PageRank shows a marked improvement of inverse average inverse rank in three out of four scenarios; social search with PageRank decreases the average rank in one out of four scenarios. Spreading activation search halves the inverse average inverse rank on a subset of the mailing list corpus which contains messages from one year, and decreases the average rank by one fifth in one scenario. On the full mailing list archive, no improvement can be detected when using spreading activation search, as regards both average rank and inverse average inverse rank.

On the corpus of conference proceedings, neither social search technique shows an improvement. It is conjectured that this is an effect of the method for choosing evaluation

items, in combination with the characteristics of the corpus.

Chapter 8

Implementation Notes

This chapter describes the prototype system used for evaluation in chapter 7. We describe design critera, the technology used for implementation, the components of the system, and the configuration files.

8.1 Design Criteria

The stated purpose of the prototype system is to enable evaluation of different retrieval techniques, with a minimum of effort for implementing new approaches that fit the data model.

This purpose led to the following functional requirements:

- The system provides a full-text index with a vector-space search algorithm.
- The system provides methods for storing, retrieving, and manipulating an associative network.
- The associative network implementation is suitable for implementing spreading activation search.
- The system is suitable for evaluation using the evaluation metrics in section 7.
- Following from the last point, the system is implemented as a batch retrieval system; only preliminary interactive facilities are provided.

The following non-functional requirements also influenced design decisions:

- The system should be modular and easily extensible to allow for experimentation with a variety of different approaches.
- Platform independence is an important factor, as it allows researchers using a variety of platforms to develop and use the prototype.
- The system should be implemented using standard open-source components. Using standard components speeds up development of the prototype. It also fosters collaboration between researchers, who have access to the same, well-documented tools.

8.2 Technology

The prototype system is written in the Java language, using the J2SE 1.4.2 SDK. In addition to the components provided by the SDK, the following open-source components were used:

- Apache Lucene[1] (see Gospodnetić and Hatcher, 2005) is a text search engine library which implements the vector-space model (see section 6.2). Lucene stores the index in a set of files; the index is used for storing both the full-text index and the associative network.

- JUNG – Java Universal Graph/Network Framework[2] (see O'Madadhain et al., 2005) is a library for modeling, analyzing and visualizing a wide variety of graphs. The classes used in the prototype for representing the associative network are directly derived from appropriate JUNG classes; we also use JUNG's PageRank implementation as well as statistics.

- Sun JavaMail[3] is a framework for mail and messaging applications; it is used for parsing email archives in RFC 822 format.

- Colt[4] is a set of libraries for high-performance scientific and technical computing. It is used in the prototype for linear algebra, matrix arithmetic and descriptive statistics.

- JDOM[5] is a library for reading, manipulating and writing XML documents; it is used in the prototype for processing configuration files.

In general, we tried to find open-source components for common tasks, in order to reduce development time.

8.3 Components

The prototype is subdivided into several components, which are described in this chapter. It contains components for modeling the associative network, for reading a network from external storage, and for searching the network. Indexing and extraction of the network is separated from the storage and retrieval architecture. Evaluation is performed by dedicated classes.

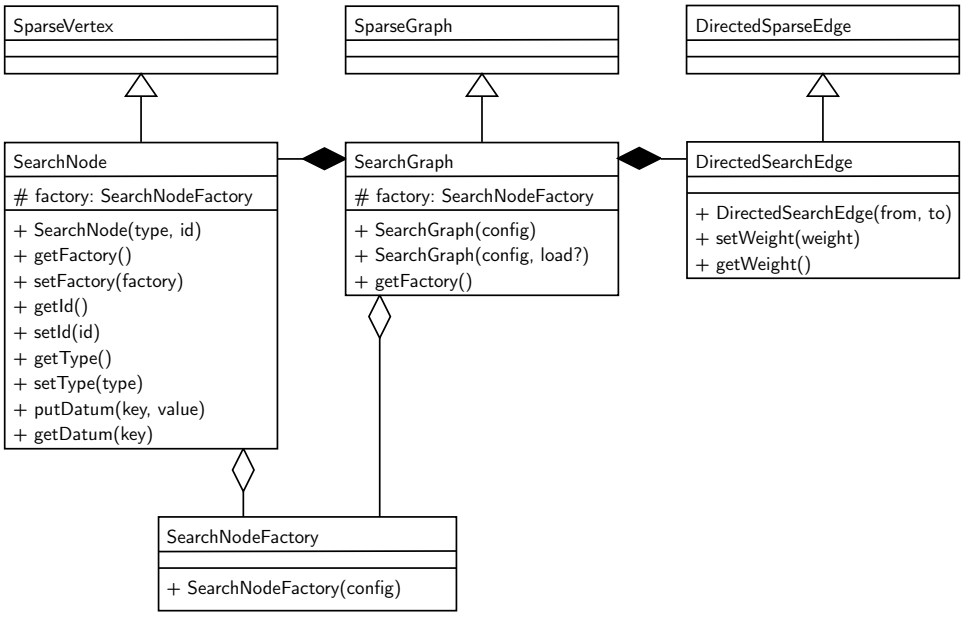


Figure 8.1: Class diagram of graph architecture

8.3.1 Associative Network

The associative network implementation of the prototype is directly derived from the corresponding classes for sparse graphs of the JUNG framework; see figure 8.1 for a class diagram. For a description of JUNG's graph model, we refer to the documentation on JUNG's web page; a conceptual overview is given in (O'Madadhain et al., 2005).

Nodes in the associative network are represented by the `SearchNode` class; they are uniquely identified by a type string and an integer id. In addition to the standard mechanisms for attaching data to classes provided by JUNG, we provide a simlar mechanism for storing explanatory data – for example the name of the person, or the title of a document.

We use directed, weighted edges for representing the network and model them using the class `DirectedSearchEdge`.

`SearchGraph` is the central class modelling the associative network. A `SearchNode` must always be associated with a `SearchGraph`, and a `DirectedSearchEdge` may only connect nodes belonging to the same `SearchGraph`.

The storage component (in section 8.3.2) uses the Factory pattern for loading parts of the associative network from external memory. A `SearchGraph` and `SearchNodes` are

[1] http://lucene.apache.org/java/, last visit on 2005/10/18.
[2] http://jung.sourceforge.net/, last visit on 2005/10/18.
[3] http://java.sun.com/products/javamail/, last visit on 2005/10/18.
[4] http://dsd.lbl.gov/~hoschek/colt/, last visit on 2005/10/18.
[5] http://www.jdom.org/, last visit on 2005/10/18.

always associated with the factory that produced them.

A `SearchGraph` may be initialized with a configuration file in XML format, in which case factories for the types declared in the configuration file will be created automatically; for the format of the configuration files see section 8.4. The storage component supports lazy loading of nodes; the boolean `load` argument of the `SearchGraph` constructor determines whether the complete graph is loaded into memory at initialization time, or whether it is loaded on demand.

8.3.2 Storage

The storage component (figure 8.2) is centered around the concept of a 'backing store' that provides access to parts of the associative network stored in external memory. A backing store provides methods to fetch a new node identified by its type and its id, as well as fetch the neighbours of a node. One backing store may provide access to several different types of nodes.

Two implementations of a backing store are available: A `JDBCBackingStore` provides access to nodes stored in a relational database, using a JDBC driver; a `LuceneBackingStore` interfaces with a Lucene index. The `JDBCBackingStore` currently does not support full-text queries, since full-text search is not a standard feature of relational databases.

A query node is produced by requesting a node of the appropriate type (stored as a constant in the `SearchNodeBackingStore` interface) from the backing store. The query phrase is attached to the query node; when fetching the neighbours of the query node, the query is automatically executed, and edges to matching documents are added.

Applications typically do not use backing stores directly; instead, a `SearchNodeFactory` is created, which initializes the backing stores and registers itself to the `SearchNodeFactoryManager`. The factory manager provides access to the factory for a given type. A factory also includes a cache for nodes which have already been fetched from the backing store.

8.3.3 Search

The `SocialSearch` class provides a common interface for social search algorithms (figure 8.3). The class returns a `SearchHits` object, which contains the nodes matched by the search, in order of their score. The hits can be filtered by a field value, and can be restricted to a set of nodes.

Two algorithms are implemented, one based on PageRank (described in section 6.3), and one based on spreading activation search (section 6.4).

The `PageRank` class implements PageRank search; it needs to be supplied with a bias value, the type identifier of the person nodes, and the graph for which PageRank values should be computed. PageRank computation is perfomed when the class is initialized.

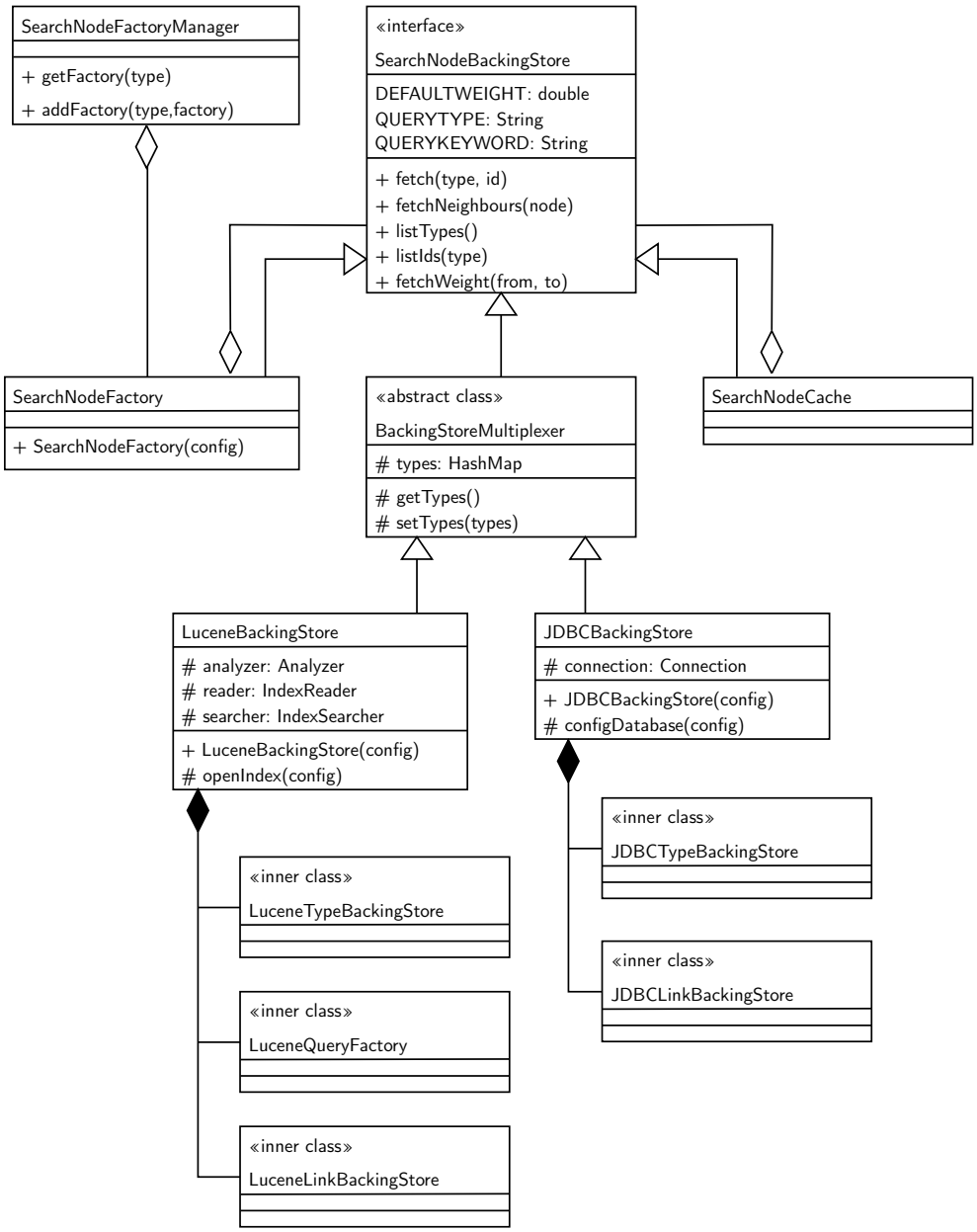


Figure 8.2: Class diagram of storage architecture

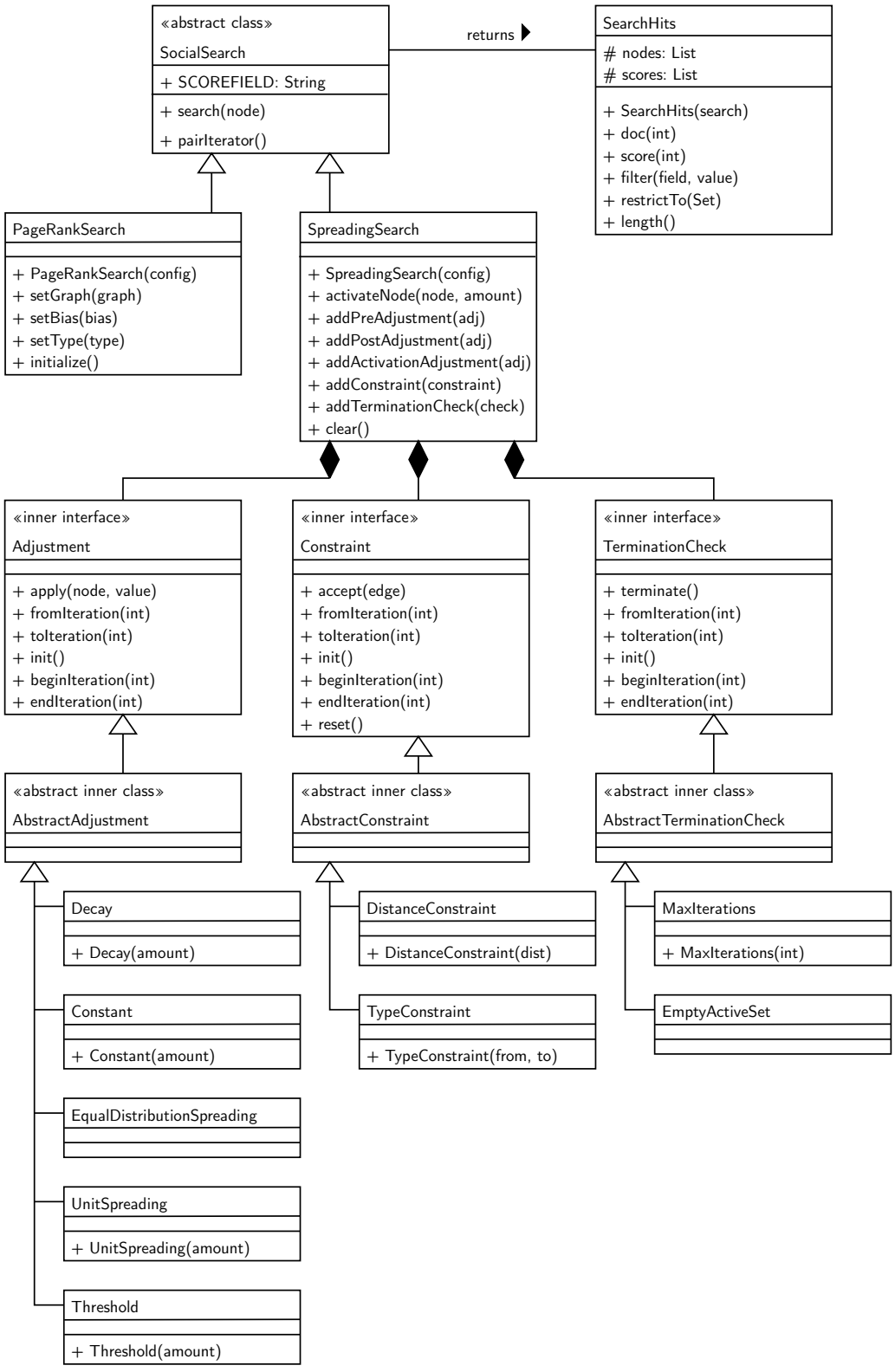


Figure 8.3: Class diagram of search architecture

The `SpreadingSearch` implements spreading activation search. Adjustments, constraints and termination checks are implemented as inner classes; they are executed in the order in which they are added to the `SpreadingSearch` object. They may be activated in specific iterations only.

An adjustment changes the activation value for a particular node; it is applied to the output energy, the input energy, or the activation energy of a node, depending on the stage to which it is added. Several adjustments may be active in one stage and are applied in the order in which they were added to the `SpreadingSearch` object.

A constraint determines whether activation spreads via an edge during the spreading stage. Constraints are also taken into account during pre- and post-adjustment, for example when determining the outgoing edges for equal distribution spreading.

8.3.4 Indexing

Indexing is separated from the storage and retrieval architecture; it is implemented in utility classes which convert from the source representation to a Lucene index.

For the mailing list archive, the source representation is a set of files which contain one email each, in RFC 822 format. Indexing of emails is a two-pass process, because the Lucene index structure does not support updating individual fields of a document once it has been written to the index.

In the first pass, messages are read from the source files, using the JavaMail API, and are added to the index. An integer identifier is assigned to each message. For each newly encountered email address, a person record is added to the index. The result of the first pass is a full-text index of the email messages, which lacks the social network between authors as well as references between messages.

In the second pass, the index from the first pass is used to resolve references between messages. A new index index is written, containing all the information in the first-pass index, with references added. The social network of authors is extracted, based on how often an author replied to messages by another. This information is also added to the second-pass index.

In the case of the SIGIR corpus, detailed information about the documents and their authors is already available in a relational database. The documents are available electronically in PDF format and have been converted to plain text. The indexer reads meta-information from the database and and adds it to the index; the document text is read from the converted PDF files.

8.3.5 Evaluation

Evaluation is performed by two classes, one performing evaluation of the baseline method and one performing evaluation of social search techniques; they are derived from a common superclass (see figure 8.4.) A known-item retrieval task consists of a dataset, a

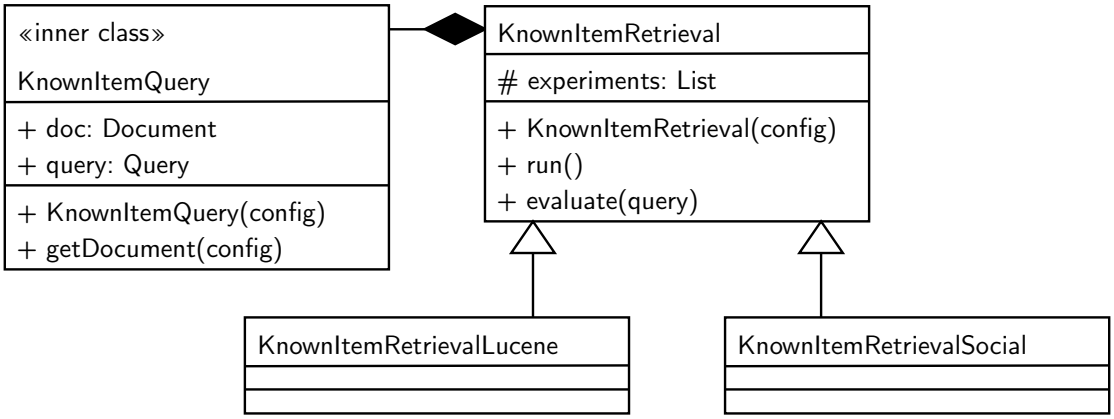


Figure 8.4: Class diagram for the evaluation classes

retrieval method, and a number of known items. A known item consists of the desired document and the associated query terms. The evaluation classes read a description of the dataset, the retrieval method and the known items from a configuration file, execute the queries and report the rank at which the desired document is found. Average rank and inverse average inverse rank are also reported; their calculation is performed using interval arithmetic.

The format of the configuration files is described in detail in section 8.4.

8.4 Configuration Files

All components of the prototype are configured using XML files. The root element of the configuration files is the `<experiments>` tag. Known items are described by the `<experiment>` tag; a known item consists of a query in the `<query>` tag; the desired document is describes by the `<item>` tag. A document is identified by one or multiple `<id>` tags, which contain a `field` attribute containing the search field, and a `value` attribute containing the field value for the desired document:

```
64 <experiments>
65   <experiment>
66     <query>+crease +patterns</query>
67     <item>
68       <id field="message-id"
69         value="&lt;BAY7-F825UhRSYSS5Eq0005cb53@hotmail.com&gt;"/>
70     </item>
71   </experiment>
```

The parameters of the retrieval method are described by the `<searchparams>` tag:

```
73   <searchparams>
```

Adjustments, constraints and termination checks for spreading activation search are defined using the corresponding tags `<adjustment>`, `<constraint>` and `<terminationcheck>`. The `type` attribute contains the type; in general, the type corresponds to the name of the implementing class in figure 8.3. The `from` and `to` tags determine in which iteration the class is active. For adjustments, the `stage` attribute determines the stage in which the adjustment is applied:

```
74      <adjustment stage="activation"
75        type="Decay"
76        value="0.1" />
77      <constraint type="DistanceConstraint" value="2" />
78      <terminationcheck type="MaxIterations" value="4" />
79      <adjustment stage="activation"
80        type="Decay"
81        value="0.0"
82        from="0"
83        to="0" />
84      <constraint type="TypeConstraint"
85        fromtype="person"
86        totype="person"
87        from="2"
88        to="2" />
```

Parameters for PageRank search are configured using the `<pagerank>` tag; it has two attributes: `type` holds the type string of the person nodes, and `bias` attribute contains the bias for PageRank computation:

```
89      <pagerank type="person" bias="0.3" />
90    </searchparams>
```

The dataset is configured using the `<backingstore>` tag; its attribute `type` is either `"lucene"` for a dataset contained in a Lucene index, or `"db"` for a dataset in a JDBC compliant database. The `<directory>` tag contains the directory where the Lucene index resides; the `<analyzer>` tag contains the class name of the analyzer to use for parsing query strings:

```
91    <backingstore type="lucene">
92      <directory>
93        data/origami-1/indexplus/
94      </directory>
95      <analyzer>
96        org.apache.lucene.analysis.MessageAnalyzer
97      </analyzer>
```

The `<type>` tags define node types contained in the dataset; the attribute `name` contains the type string, whereas the attribute `keyword` contains the keyword used to identify nodes of this type in the dataset. `<field>` tags describe fields which contain additional

information about the node. <link> tags describe links to other nodes; the target attribute contains the type of the target node, and the field tag holds the field containing the id of the target node:

```
98      <type name="person" keyword="person">
99        <field>address</field>
100       <link target="message" field="message" />
101       <link target="person" field="followuplink" />
102     </type>
103     <type name="message" keyword="message">
104       <field>from</field>
105       <field>subject</field>
106       <field>date</field>
107       <field>filename</field>
108       <link target="person" field="fromlink" />
109     </type>
```

The <query> tag describes the fields which are searched when executing full-text queries:

```
110     <query maxhits="5000">
111       <field>text</field>
112     </query>
113   </backingstore>
114 </experiments>
```

The following DTD describes the format of the configuration files in a concise manner:

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE experiments [
3    <!ELEMENT adjustment EMPTY>
4    <!ATTLIST adjustment
5      from NMTOKEN #IMPLIED
6      stage NMTOKEN #REQUIRED
7      to NMTOKEN #IMPLIED
8      type NMTOKEN #REQUIRED
9      value NMTOKEN #IMPLIED
10   >
11   <!ELEMENT analyzer (#PCDATA)>
12   <!ELEMENT backingstore
13     (((directory,analyzer)|database),type+,query)>
14   <!ATTLIST backingstore type (lucene|db) #REQUIRED>
15   <!ELEMENT constraint EMPTY>
16   <!ATTLIST constraint
17     from NMTOKEN #IMPLIED
18     fromtype NMTOKEN #IMPLIED
19     to NMTOKEN #IMPLIED
20     totype NMTOKEN #IMPLIED
21     type NMTOKEN #REQUIRED
```

```
22      value NMTOKEN #IMPLIED
23    >
24    <!ELEMENT directory (#PCDATA)>
25    <!ELEMENT database (connectstring,driver)>
26    <!ELEMENT connectstring (#PCDATA)>
27    <!ELEMENT driver (#PCDATA)>
28    <!ELEMENT experiment (query,item)>
29    <!ELEMENT experiments
30      (experiment+,searchparams,backingstore)>
31    <!ELEMENT field (#PCDATA)>
32    <!ATTLIST field EMPTY>
33    <!ELEMENT id EMPTY>
34    <!ATTLIST id
35      field CDATA #REQUIRED
36      value CDATA #REQUIRED
37    >
38    <!ELEMENT item (id+)>
39    <!ELEMENT link EMPTY>
40    <!ATTLIST link
41      field NMTOKEN #REQUIRED
42      target NMTOKEN #REQUIRED
43    >
44    <!ELEMENT pagerank EMPTY>
45    <!ATTLIST pagerank
46      bias NMTOKEN #REQUIRED
47      type NMTOKEN #REQUIRED
48    >
49    <!ELEMENT query (#PCDATA|field)*>
50    <!ATTLIST query maxhits NMTOKEN #IMPLIED>
51    <!ELEMENT searchparams
52      (adjustment|constraint|pagerank|terminationcheck)*>
53    <!ELEMENT terminationcheck EMPTY>
54    <!ATTLIST terminationcheck
55      type NMTOKEN #REQUIRED
56      value NMTOKEN #IMPLIED
57    >
58    <!ELEMENT type (field+,link+)>
59    <!ATTLIST type
60      keyword NMTOKEN #REQUIRED
61      name NMTOKEN #REQUIRED
62    >
63  ]>
```

8.5 Summary

The prototype system implements the model and the techniques described in sections 5 and 6. It is implemented in the Java programming language, using the J2SE 1.4.2 SDK. Open-source componets are employed for parts of the prototype.

The prototype is implemented as a batch retrieval system, using a modular structure which allows for rapid implementation of different retrieval methods. The architecture supports storing the dataset in two formats: In a relational database with a JDBC driver, or in a Lucene index.

Configuration of the system is performed using XML files. The configuration files contain a complete description of a known-item retrieval task, including the known-item queries, the desired documents, the parameters of the retrieval methods, and a description of the dataset.

Chapter 9

Conclusion

In this thesis, we research how to integrate social networks in the information retrieval process and whether this integration leads to a performance improvement.

We examine the process of information retrieval and production and how social interaction is present in these activities. In particular, several applications of the internet are identified as social media, for example wikis, blogs, or mailing lists.

We propose a model for social information retrieval, which integrates the domains of social network analysis and information retrieval. Meaningful associations become apparent which are not part of the traditional models. We define social information retrieval as a retrieval process which includes a well-defined subset of the constituents of the social IR model.

Two techniques are described which implement social IR. Both techniques are inspired by previous attempts at graph-based information retrieval: The PageRank algorithm, which is widely used for link analysis in the world wide web, and spreading activation search, a search technique for semantic and associative networks. The algorithms differ in that PageRank is a technique which uses global properties of the graph, whereas spreading activation search uses local links.

We evaluate the techniques in a known-item retrieval scenario. We compare the characteristics of our corpora with the web graph and with previously examined social networks. The similarities between social networks and the web graph in particular motivate the application of web retrieval techniques to social information retrieval.

We conclude that social network analysis is an important tool for information retrieval. The main argument supporting this conclusion is the importance of social interaction for information retrieval and production.

9.1 Impact

We apply graph-based techniques to social networks, using them outside their traditional domains within information retrieval, namely web retrieval and retrieval on semantic networks. We thereby extend the state of the art in graph-based retrieval techniques.

We acknowledge recent developments in statistical network analysis and theory of random graphs and apply them in the context of information retrieval. We hope that

further developments in the young field of statistical network analysis will continue to cross-pollinate information retrieval.

There is currently an indisputable interest in 'social software', exemplified by the popularity of blogs and wikis, 'social tagging' systems, and 'social bookmarking'. The number of mentions of Wikipedia, a project founded on social software principles, in reputable publications like the New York Times or the Guardian alone is witness to this trend.

The commonly cited benefits of social software, for example improved communication among group members or emergence of communities, is important but intangible. We aim to derive tangible benefits from the application of social networks, namely improved retrieval performance – by providing retrieval techniques which are tailored to the emerging field of social software. We believe that these tangible benefits will accelerate the adoption of social software.

9.2 Limitations

The main limitation of social IR follows from its domain model: it is only applicable where a social network is present in the domain, or can be derived. Furthermore, the quality of the social network is crucial: We see in section 7.3.2 that a poorly formed social network can lead to a failure of social retrieval methods.

Limitations of other graph-based retrieval methods also apply to social information retrieval. Commonly cited limitations of PageRank are that its benefits are greatest for underspecified queries with many relevant results; for spreading activation search, the structure of the network is of crucial importance.

The position of an author in the social network may be misleading as regards his authority. In particular, we see that social retrieval techniques are good at identifying multipliers, but fail to identify innovators.

9.3 Future Work

Evaluation of the prototype system was performed using non-standardized corpora and evaluation scenarios. For comparing the prototype system with current and future information retrieval systems, standardized corpora and evaluation scenarios must be constructed. Standardized scenarios also permit to tune the system for a particular retrieval task.

While the current prototype implementation as a batch retrieval system satisfies the requirements for the chosen evaluation scenarios, the implementation of an interactive prototype is indispensable for further evaluation. In particular, user studies need to be performed to find out how users react to the presence of social network information in a retrieval application. Visualization of the social network needs to be researched as part

of the result presentation of a social IR system.

The techniques and evaluation scenarios described in this thesis use only a subset of the possible relations present in the social IR domain model. It will be instructive to apply social retrieval techniques to domains exhibiting different subsets of the domain model.

When choosing algorithms for social IR, we limit the evaluation to two popular algorithms for graph-based retrieval. Other algorithms need to be examined to determine their suitability for social IR. Topic-sensitive PageRank (Haveliwala, 2002) in particular is a promising candidate, as it allows for a social authority measure tailored to a community or a single individual.

Bibliography

John R. Anderson. *The Architecture of Cognition*. Cognitive Science Series. Harvard University Press, 1983. ISBN 0-674-04425-8.

Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.

Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286:509–512, October 1999. doi: http://dx.doi.org/10.1126/science.286.5439.509.

Adam Berger and John Lafferty. Information retrieval as statistical translation. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 222–229, New York, NY, USA, 1999. ACM Press. ISBN 1-58113-096-1. doi: http://doi.acm.org/10.1145/312624.312681.

Tim Berners-Lee, Robert Cailliau, Ari Luotonen, Henrik Frystyk Nielsen, and Arthur Secret. The World-Wide Web. *Communications of the ACM*, 37(8):76–82, 1994. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/179606.179671.

Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001. URL http://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&catID=2.

Krishna Bharat. SearchPad: explicit capture of search context to support Web search. *Computer Networks*, 33(1-6):493–501, 2000. ISSN 1389-1286. doi: http://dx.doi.org/10.1016/S1389-1286(00)00047-5.

Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998. URL ftp://db.stanford.edu/pub/papers/google.pdf.

Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in

the web. *Computer Networks*, 33:309–320, 2000. URL http://www.people.cornell.edu/pages/dc288/Paper1.pdf.

Nick Craswell and David Hawking. Overview of the TREC-2004 Web track. In E. M. Voorhees and Lori P. Buckland, editors, *Proceedings of the Thirteenths Text REtrieval Conference (TREC 2004)*, number 500-261 in NIST Special Publications, Gaithersburg, MD, November 2004. U. S. National Institute of Standards and Technology. URL http://trec.nist.gov/pubs/trec13/papers/WEB.OVERVIEW.pdf.

F. Crestani. Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11(6):453–482, 1997. ISSN 0269-2821. doi: http://dx.doi.org/10.1023/A:1006569829653.

Fabio Crestani and Puay Leng Lee. Searching the web by constrained spreading activation. *Information Processing and Management*, 36(4):585–605, 2000. ISSN 0306-4573. doi: http://dx.doi.org/10.1016/S0306-4573(99)00073-4.

D. Crocker. Standard for the format of ARPA Internet text messages. RFC 822 (Standard), August 1982. URL http://www.ietf.org/rfc/rfc822.txt. Obsoleted by RFC 2822, updated by RFCs 1123, 1138, 1148, 1327, 2156.

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990. URL http://citeseer.ist.psu.edu/deerwester90indexing.html.

Peter Sheridan Dodds, Roby Muhamad, and Duncan J. Watts. An experimental study of search in global social networks. *Science*, 301:827–829, August 2003.

J. Fagan. Automatic phrase indexing for document retrieval. In *SIGIR '87: Proceedings of the 10th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 91–101, New York, NY, USA, 1987. ACM Press. ISBN 0-89791-232-2. doi: http://doi.acm.org/10.1145/42005.42016.

Gary William Flake, Kostas Tsioutsiouliklis, and Leonid Zhukov. Methods for mining web communities: Bibliometric, spectral, and flow. In Alexandra Poulovassilis and Mark Levene, editors, *Web Dynamics*, chapter 4, pages 45–68. Springer Verlag, 2004. ISBN 3-540-40676-X. URL http://research.yahoo.com/publications/4.pdf.

William B. Frakes and Ricardo Baeza-Yates, editors. *Information Retrieval. Data Structures & Algorithms*. Prentice Hall, 1992.

Jill Freyne and Barry Smyth. An experiment in social search. In Wolfgang Nejdl and Paul De Bra, editors, *Adaptive Hypermedia and Adaptive Web-Based Systems, Third International Conference, AH 2004, Eindhoven, The Netherlands, August*

23-26, 2004, Proceedings, volume 3137 of *Lecture Notes in Computer Science*, pages 95–103. Springer, 2004. ISBN 3-540-22895-0.

J. Fürnkranz and P. A. Flach. An analysis of rule evaluation metrics. In *Proceedings of the 20th International Conference on Machine Learning (ICML'03)*, pages 202–209. AAAI Press, January 2003. ISBN 1-57735-189-4. URL http://www.cs.bris.ac.uk/Publications/Papers/1000705.pdf.

John S. Garofolo, Ellen M. Voorhees, Vincent M. Stanford, and Karen Spärck Jones. TREC-6 1997 spoken document retrieval track overview and results. In E. M. Voorhees and D. K. Harman, editors, *Proceedings of the Sixth Text REtrieval Conference TREC-6*, number 500-240 in NIST Special Publications. U.S. National Institute of Standards and Technology (NIST), 1997. URL http://trec.nist.gov/pubs/trec6/papers/sdr97.ps.gz.

Melanie Gnasa, Sascha Alda, Jasmin Grigull, and Armin B. Cremers. Towards virtual knowledge communities in peer-to-peer networks. In Jamie Callan, Fabio Crestani, and Mark Sanderson, editors, *Distributed Multimedia Information Retrieval*, volume 2924 of *Lecture Notes in Computer Science*, pages 143–155. Springer, 2003. URL http://www.springerlink.com/index/NUR92TH9821N5TPJ.

Melanie Gnasa, Markus Won, and Armin B. Cremers. Three pillars for congenial web search. Continuous evaluation for enhancing web search effectiveness. *Journal of Web Engineering*, 3(3&4):252–280, 2004. ISSN 1540-9589. URL http://www.informatik.uni-bonn.de/~won/Download/wwwjournal2004.pdf.

Otis Gospodnetić and Erik Hatcher. *Lucene in Action*. Manning, 2005. URL http://www.lucenebook.com/.

Mark S. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78(6):1360–1380, May 1973. URL http://www.stanford.edu/dept/soc/people/faculty/granovetter/documents/TheStrengthofWeakTies.pdf.

Nadir Gül. MyPush – Ein kollaborativer Push Dienst für die automatische Informationsbeschaffung in einem Peer-to-Peer Netzwerk. Diploma thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, March 2004.

Richard W. Hamming. *Coding and information theory*. Prentice-Hall, Englewood Cliffs, 1980. ISBN 0-13-139139-9.

J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, April 1982. URL http://www.med.mcgill.ca/epidemiology/hanley/software/Hanley_McNeil_Radiology_82.pdf.

D. Harman. Towards interactive query expansion. In *SIGIR '88: Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 321–331, New York, NY, USA, 1988. ACM Press. ISBN 2-7061-0309-4. doi: http://doi.acm.org/10.1145/62437.62469.

Donna Harman. Relevance feedback revisited. In *SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 1–10, New York, NY, USA, 1992. ACM Press. ISBN 0-89791-523-2. doi: http://doi.acm.org/10.1145/133160.133167.

Taher H. Haveliwala. Topic-sensitive PageRank. In *WWW '02: Proceedings of the eleventh international conference on World Wide Web*, pages 517–526. ACM Press, 2002. ISBN 1-58113-449-5. doi: http://doi.acm.org/10.1145/511446.511513.

Brian Hayes. A lucid interval. *American Scientist*, 91(6):484–488, November–December 2003. URL http://www.cs.utep.edu/interval-comp/hayes.pdf.

Marti A. Hearst and Jan O. Pedersen. Reexamining the cluster hypothesis: scatter/-gather on retrieval results. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 76–84, New York, NY, USA, 1996. ACM Press. ISBN 0-89791-792-8. doi: http://doi.acm.org/10.1145/243199.243216.

Svante Janson, Donald E. Knuth, Tomasz Łuczak, and Boris Pittel. The birth of the giant component. *Random Structures & Algorithms*, 4(3):233–358, 1993. URL http://arxiv.org/pdf/math.PR/9310236.

Paul B. Kantor and Ellen M. Voorhees. Report on the TREC-5 confusion track. In E. M. Voorhees and D. K. Harman, editors, *Proceedings of the Fifth Text REtrieval Conference TREC-5*, number 500-238 in NIST Special Publications. U.S. National Institute of Standards and Technology (NIST), 1996. URL http://trec.nist.gov/pubs/trec5/papers/confusion_track.ps.gz.

Henry Kautz, Bart Selman, and Mehul Shah. Referral web: combining social networks and collaborative filtering. *Commununications of the ACM*, 40(3):63–65, 1997a. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/245108.245123.

Henry Kautz, Bart Selman, and Mehul Shah. The hidden web. *AI Magazine*, 18(2):27–36, 1997b. URL http://www.cs.washington.edu/homes/kautz/referralweb/doc/aimag.pdf.

Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999. ISSN 0004-5411. doi: http://doi.acm.org/10.1145/324133.324140.

Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. GroupLens: applying collaborative filtering to Usenet news. *Communications of the ACM*, 40(3):77–87, 1997. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/245108.245126.

F. W. Lancaster. *Information Retrieval Systems: Characteristics, Testing, and Evaluation*. Wiley, New York, 1968.

Udi Manber. Foreword. In William B. Frakes and Ricardo Baeza-Yates, editors, *Information Retrieval. Data Structures & Algorithms*, pages v–vi. Prentice Hall, 1992.

Stanley Milgram. The small-world problem. *Psychology Today*, 2:60–67, 1967.

David R. H. Miller, Tim Leek, and Richard M. Schwartz. A hidden markov model information retrieval system. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 214–221, New York, NY, USA, 1999. ACM Press. ISBN 1-58113-096-1. doi: http://doi.acm.org/10.1145/312624.312680.

M. E. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences of the United States of America*, 98(2):404–409, January 2001. doi: 10.1073/pnas.021544898. URL http://dx.doi.org/10.1073/pnas.021544898.

M. E. J. Newman and Juyong Park. Why social networks are different from other types of networks. *Physical Review E*, 68:036122, September 2003. doi: 10.1103/PhysRevE.68.036122. URL http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=pubmed&list_uids=14524847.

M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64:026118, 2001. doi: http://dx.doi.org/10.1103/PhysRevE.64.026118. URL http://link.aps.org/abstract/PRE/v64/e026118.

Joshua O'Madadhain, Danyel Fisher, Padhraic Smyth, Scott White, and Yan-Biao Boey. Analysis and visualization of network data using JUNG. *Journal of Statistical Software*, 2005. URL http://jung.sourceforge.net/doc/JUNG_journal.pdf. To appear.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the Web. Technical report, Stanford University, November 1999. URL http://dbpubs.stanford.edu:8090/pub/1999-66.

Gabriel Pinski and Francis Narin. Citation influence for journal aggregates of scientific publications: Theory, with application to the literature of physics. *Information*

Processing and Management, 12(5):297–312, 1976. doi: http://dx.doi.org/10.1016/0306-4573(76)90048-0.

Peter Pirolli, James Pitkow, and Ramana Rao. Silk from a sow's ear: extracting usable structures from the Web. In *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 118–125, New York, NY, USA, 1996. ACM Press. ISBN 0-89791-777-4. doi: http://doi.acm.org/10.1145/238386.238450. URL http://www.pitkow.com/docs/1996-CHI-Silk.pdf.

James Pitkow, Hinrich Schütze, Todd Cass, Rob Cooley, Don Turnbull, Andy Edmonds, Eytan Adar, and Thomas Breuel. Personalized search. *Commununications of the ACM*, 45(9):50–55, 2002. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/567498.567526.

Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281, New York, NY, USA, 1998. ACM Press. ISBN 1-58113-015-5. doi: http://doi.acm.org/10.1145/290941.291008.

Scott Everett Preece. *A Spreading Activation Network Model for Information Retrieval*. PhD thesis, University of Illinois at Urbana-Champaign, 1981. URL http://wwwlib.umi.com/dissertations/fullcit/8203555.

M. Ross Quillian. Semantic memory. In Marvin Minsky, editor, *Semantic Information Processing*. MIT Press, Cambridge, Mass., 1968.

Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. GroupLens: an open architecture for collaborative filtering of netnews. In *CSCW '94: Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186, New York, NY, USA, 1994. ACM Press. ISBN 0-89791-689-1. doi: http://doi.acm.org/10.1145/192844.192905.

S. E. Robertson, C. J. van Rijsbergen, and M. F. Porter. Probabilistic models of indexing and searching. In *SIGIR '80: Proceedings of the 3rd annual ACM conference on Research and development in information retrieval*, pages 35–56, Kent, UK, 1981. Butterworth & Co. ISBN 0-408-10775-8.

Nicholas C. Romano, Jr, Dmitri Roussinov, Jay F. Nunamaker, Jr, and Hsinchun Chen. Collaborative information retrieval environment: Integration of information retrieval with group support systems. In *HICSS '99: Proceedings of the Thirty-Second Annual Hawaii International Conference on System Sciences-Volume 1*, pages 1053–1062, Washington, DC, USA, 1999. IEEE Computer Society. ISBN 0-7695-0001-3.

Thorsten Ruhl. Personal Search Memory – Design und Realisierung einer Suchschnittstelle zur kombinierten Suche in früheren und neuen Suchergebnissen. Diploma thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, 2003.

Gerard Salton. Associative document retrieval techniques using bibliographic information. *Journal of the ACM*, 10(4):440–457, 1963. ISSN 0004-5411. doi: http://doi.acm.org/10.1145/321186.321188.

Gerard Salton and Chris Buckley. On the use of spreading activation methods in automatic information retrieval. In *Proceedings of the ACM SIGIR*, Grenoble, France, June 1988a. URL http://doi.acm.org/10.1145/62437.62447.

Gerard Salton and Chris Buckley. Term-weighting approaches in automatic information retrieval. *Information Processing and Management*, 24(5):513–523, 1988b.

Mehul Shah. ReferralWeb: A resource location system guided by personal relations. Master's thesis, Massachusetts Institute of Technology, May 1997.

Craig Silverstein, Hannes Marais, Monika Henzinger, and Michael Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12, 1999. ISSN 0163-5840. doi: http://doi.acm.org/10.1145/331403.331405.

Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted document length normalization. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 21–29, New York, NY, USA, 1996. ACM Press. ISBN 0-89791-792-8. doi: http://doi.acm.org/10.1145/243199.243206.

Alan F. Smeaton, Gary Keogh, Cathal Gurrin, Kieran McDonald, and Tom Sødring. Analysis of papers from twenty-five years of SIGIR conferences: What have we been doing for the last quarter of a century? *SIGIR Forum*, 36(2):39–43, 2002. ISSN 0163-5840. doi: http://doi.acm.org/10.1145/792550.792556. URL http://portal.acm.org/citation.cfm?id=792556.

Beth Sundheim and Ralph Grishman, editors. *MUC6 '95: Proceedings of the 6th conference on Message understanding*, Morristown, NJ, USA, 1995. Association for Computational Linguistics. ISBN 1-55860-402-2.

H. Turtle and W. B. Croft. Inference networks for document retrieval. In *SIGIR '90: Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 1–24, New York, NY, USA, 1990. ACM Press. ISBN 0-89791-408-2. doi: http://doi.acm.org/10.1145/96749.98006.

Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, June 1998. ISSN 0028-0836. URL http://dx.doi.org/10.1038/30918.

Etienne Wenger. How we learn. Communities of practice. The social fabric of a learning organization. *Healthcare Forum Journal*, 39(4):20–26, 1996. URL http://www.ewenger.com/pub/pubhealthcareforum.htm.

Scott White and Padhraic Smyth. Algorithms for estimating relative importance in networks. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 266–275, New York, NY, USA, 2003. ACM Press. ISBN 1-58113-737-0. doi: http://doi.acm.org/10.1145/956750.956782.

T. D. Wilson. On user studies and information needs. *Journal of Librarianship*, 37 (1):3–15, 1981. URL http://informationr.net/tdw/publ/papers/1981infoneeds.html.

T. D. Wilson. Information needs and uses: fifty years of progress. In B. C. Vickery, editor, *Fifty years of information progress: a Journal of Documentation review*, pages 15–51. Aslib, London, 1994. URL http://informationr.net/tdw/publ/papers/1994FiftyYears.html.

William A. Woods. What's in a link: Foundations for semantic networks. In Daniel G. Bobrow and Allan Collins, editors, *Representation and Understanding*. Academic Press, New York, 1975. ISBN 0-12-108550-3.

Hugo Zaragoza, Nick Craswell, Michael Taylor, Suchi Saria, and Stephen Robertson. Microsoft Cambridge and TREC-13: Web and HARD tracks. In E. M. Voorhees and Lori P. Buckland, editors, *Proceedings of the Thirteenths Text REtrieval Conference (TREC 2004)*, number 500-261 in NIST Special Publications, Gaithersburg, MD, November 2004. U. S. National Institute of Standards and Technology. URL http://trec.nist.gov/pubs/trec13/papers/microsoft-cambridge.web.hard.pdf.

Ehrenwörtliche Erklärung

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Arbeit ‘Social Information Retrieval’ selbständig angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche gekennzeichnet.

Sebastian Marius Kirsch
Bonn, den 2. November 2005