Modern Information Retrieval: A Brief Overview

Amit Singhal
Google, Inc.
singhal@google.com

Abstract

For thousands of years people have realized the importance of archiving and finding information. With the advent of computers, it became possible to store large amounts of information; and finding useful information from such collections became a necessity. The field of Information Retrieval (IR) was born in the 1950s out of this necessity. Over the last forty years, the field has matured considerably. Several IR systems are used on an everyday basis by a wide variety of users. This article is a brief overview of the key advances in the field of Information Retrieval, and a description of where the state-of-the-art is at in the field.

1 Brief History

The practice of archiving written information can be traced back to around 3000 BC, when the Sumerians designated special areas to store clay tablets with cuneiform inscriptions. Even then the Sumerians realized that proper organization and access to the archives was critical for efficient use of information. They developed special classifications to identify every tablet and its content. (See `http://www.libraries.gr` for a wonderful historical perspective on modern libraries.)

The need to store and retrieve written information became increasingly important over centuries, especially with inventions like paper and the printing press. Soon after computers were invented, people realized that they can be used for storing and mechanically retrieving large amounts of information. In 1945 Vannevar Bush published a ground breaking article titled "As We May Think" that gave birth to the idea of automatic access to large amounts of stored knowledge. [5] In the 1950s, this idea materialized into more concrete descriptions of how archives of text could be searched automatically. Several works emerged in the mid 1950s that elaborated upon the basic idea of searching text with a computer. One of the most influential methods was described by H.P. Luhn in 1957, in which (put simply) he proposed using words as indexing units for documents and measuring word overlap as a criterion for retrieval. [17]

Several key developments in the field happened in the 1960s. Most notable were the development of the SMART system by Gerard Salton and his students, first at Harvard University and later at Cornell University; [25] and the Cranfield evaluations done by Cyril Cleverdon and his group at the College of Aeronautics in Cranfield. [6] The Cranfield tests developed an evaluation methodology for retrieval systems that is still in use by IR systems today. The SMART system, on the other hand, allowed researchers to experiment with ideas to

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

improve search quality. A system for experimentation coupled with good evaluation methodology allowed rapid progress in the field, and paved way for many critical developments.

The 1970s and 1980s saw many developments built on the advances of the 1960s. Various models of document retrieval were developed and advances were made along all dimensions of the retrieval process. These new models/techniques were experimentally proven to be effective on small text collections (several thousand articles) available to researchers at the time. However, due to lack of availability of large text collections, the question whether these models and techniques would scale to larger corpora remained unanswered. This changed in 1992 with the inception of Text Retrieval Conference, or TREC[1]. [11] TREC is a series of evaluation conferences sponsored by various US Government agencies under the auspices of NIST, which aims at encouraging research in IR from large text collections.

With large text collections available under TREC, many old techniques were modified, and many new techniques were developed (and are still being developed) to do effective retrieval over large collections. TREC has also branched IR into related but important fields like retrieval of spoken information, non-English language retrieval, information filtering, user interactions with a retrieval system, and so on. The algorithms developed in IR were the first ones to be employed for searching the World Wide Web from 1996 to 1998. Web search, however, matured into systems that take advantage of the cross linkage available on the web, and is not a focus of the present article. In this article, I will concentrate on describing the evolution of modern textual IR systems ([27, 33, 16] are some good IR resources).

2 Models and Implementation

Early IR systems were boolean systems which allowed users to specify their information need using a complex combination of boolean ANDs, ORs and NOTs. Boolean systems have several shortcomings, e.g., there is no inherent notion of document ranking, and it is very hard for a user to form a good search request. Even though boolean systems usually return matching documents in some order, e.g., ordered by date, or some other document feature, relevance ranking is often not critical in a boolean system. Even though it has been shown by the research community that boolean systems are less effective than ranked retrieval systems, many power users still use boolean systems as they feel more in control of the retrieval process. However, most everyday users of IR systems expect IR systems to do ranked retrieval. IR systems rank documents by their estimation of the usefulness of a document for a user query. Most IR systems assign a numeric score to every document and rank documents by this score. Several models have been proposed for this process. The three most used models in IR research are the vector space model, the probabilistic models, and the inference network model.

2.1 Vector Space Model

In the vector space model text is represented by a vector of *terms*. [28] The definition of a term is not inherent in the model, but terms are typically words and phrases. If words are chosen as terms, then every word in the vocabulary becomes an independent dimension in a very high dimensional vector space. Any text can then be represented by a vector in this high dimensional space. If a term belongs to a text, it gets a non-zero value in the text-vector along the dimension corresponding to the term. Since any text contains a limited set of terms (the vocabulary can be millions of terms), most text vectors are very sparse. Most vector based systems operate in the positive quadrant of the vector space, i.e., no term is assigned a negative value.

To assign a numeric score to a document for a query, the model measures the *similarity* between the query vector (since query is also just text and can be converted into a vector) and the document vector. The similarity between two vectors is once again not inherent in the model. Typically, the angle between two vectors is used as a measure of divergence between the vectors, and cosine of the angle is used as the numeric similarity (since

[1] `http://trec.nist.gov`

cosine has the nice property that it is 1.0 for identical vectors and 0.0 for orthogonal vectors). As an alternative, the inner-product (or dot-product) between two vectors is often used as a similarity measure. If all the vectors are forced to be unit length, then the cosine of the angle between two vectors is same as their dot-product. If $\vec{D}$ is the document vector and $\vec{Q}$ is the query vector, then the dot-product similarity between document D and query Q (or score of D for Q) can be represented as:

$$Sim(\vec{D}, \vec{Q}) = \sum_{t_i \in Q, D} w_{t_{iQ}} \cdot w_{t_{iD}}$$

where $w_{t_{iQ}}$ is the value of the ith component in the query vector $\vec{Q}$, and $w_{t_{iD}}$ is the ith component in the document vector $\vec{D}$. (Since any word not present in either the query or the document has a $w_{t_{iQ}}$ or $w_{t_{iD}}$ value of 0, respectively, we can do the summation only over the terms common in the query and the document.) How we arrive at $w_{t_{iQ}}$ and $w_{t_{iD}}$ is not defined by the model, but is quite critical to the search effectiveness of an IR system. $w_{t_{iD}}$ is often referred to as the *weight* of term-i in document D, and is discussed in detail in Section 4.1.

2.2 Probabilistic Models

This family of IR models is based on the general principle that documents in a collection should be ranked by decreasing probability of their relevance to a query. This is often called *the probabilistic ranking principle* (PRP). [20] Since true probabilities are not available to an IR system, probabilistic IR models *estimate* the probability of relevance of documents for a query. This estimation is the key part of the model, and this is where most probabilistic models differ from one another. The initial idea of probabilistic retrieval was proposed by Maron and Kuhns in a paper published in 1960. [18] Since then, many probabilistic models have been proposed, each based on a different probability estimation technique.

Due to space limitations, it is not possible to discuss the details of these models here. However, the following description abstracts out the common basis for these models. We denote the probability of relevance for document D by $P(R|D)$. Since this ranking criteria is monotonic under log-odds transformation, we can rank documents by $log\frac{P(R|D)}{P(\bar{R}|D)}$, where $P(\bar{R}|D)$ is the probability that the document is non-relevant. This, by simple bayes transform, becomes $log\frac{P(D|R) \cdot P(R)}{P(D|\bar{R}) \cdot P(\bar{R})}$. Assuming that the prior probability of relevance, i.e., $P(R)$, is independent of the document under consideration and thus is constant across documents, $P(R)$ and $P(\bar{R})$ are just scaling factors for the final document scores and can be removed from the above formulation (for ranking purposes). This further simplifies the above formulation to: $log\frac{P(D|R)}{P(D|\bar{R})}$.

Based on the assumptions behind estimation of $P(D|R)$, different probabilistic models start diverging at this point. In the simplest form of this model, we assume that terms (typically words) are mutually independent (this is often called the *independence assumption*), and $P(D|R)$ is re-written as a product of individual term probabilities, i.e., probability of presence/absence of a term in relevant/non-relevant documents:

$$P(D|R) = \prod_{t_i \in Q, D} P(t_i|R) \cdot \prod_{t_j \in Q, \bar{D}} (1 - P(t_j|R))$$

which uses probability of presence of a term t_i in relevant documents for all terms that are common to the query and the document, and the probability of absence of a term t_j from relevant documents for all terms that are present in the query and absent from the document. If p_i denotes $P(t_i|R)$, and q_i denotes $P(t_i|\bar{R})$, the ranking formula $log(\frac{P(D|R)}{P(D|\bar{R})})$ reduces to:

$$log\frac{\prod_{t_i \in Q, D} p_i \cdot \prod_{t_j \in Q, \bar{D}} (1 - p_j)}{\prod_{t_i \in Q, D} q_i \cdot \prod_{t_j \in Q, \bar{D}} (1 - q_j)}$$

For a given query, we can add to this a constant $log(\prod_{t_i \in Q} \frac{1-q_i}{1-p_i})$ to transform the ranking formula to use only the terms present in a document:

$$log \prod_{t_i \in Q,D} \frac{p_i \cdot (1-q_i)}{q_i \cdot (1-p_i)} \qquad or \qquad \sum_{t_i \in Q,D} log \frac{p_i \cdot (1-q_i)}{q_i \cdot (1-p_i)}$$

Different assumptions for estimation of p_i and q_i yield different document ranking functions. E.g., in [7] Croft and Harper assume that p_i is the same for all query terms and $\frac{p_i}{1-p_i}$ is a constant and can be ignored for ranking purposes. They also assume that almost all documents in a collection are non-relevant to a query (which is very close to truth given that collections are large) and estimate q_i by $\frac{n_i}{N}$, where N is the collection size and n_i is the number of documents that contain term-i. This yields a scoring function $\sum_{t_i \in Q,D} log \frac{N-n_i}{n_i}$ which is similar to the inverse document frequency function discussed in Section 4.1. Notice that if we think of $log \frac{p_i \cdot (1-q_i)}{q_i \cdot (1-p_i)}$ as the weight of term-i in document D, this formulation becomes very similar to the similarity formulation in the vector space model (Section 2.1) with query terms assigned a unit weight.

2.3 Inference Network Model

In this model, document retrieval is modeled as an inference process in an inference network. [32] Most techniques used by IR systems can be implemented under this model. In the simplest implementation of this model, a document instantiates a term with a certain strength, and the credit from multiple terms is accumulated given a query to compute the equivalent of a numeric score for the document. From an operational perspective, the strength of instantiation of a term for a document can be considered as the *weight* of the term in the document, and document ranking in the simplest form of this model becomes similar to ranking in the vector space model and the probabilistic models described above. The strength of instantiation of a term for a document is not defined by the model, and any formulation can be used.

2.4 Implementation

Most operational IR systems are based on the *inverted list* data structure. This enables fast access to a list of documents that contain a term along with other information (for example, the weight of the term in each document, the relative positions of the term in each document, etc.). A typical inverted list may be stored as:

$$t_i \rightarrow < d_a, ... >, < d_b, ... >, ..., < d_n, ... >$$

which depicts that term-i is contained in d_a, d_b, ..., d_n, and stores any other information. All models described above can be implemented using inverted lists. Inverted lists exploit the fact that given a user query, most IR systems are only interested in scoring a small number of documents that contain some query term. This allows the system to only score documents that will have a non-zero numeric score. Most systems maintain the scores in a heap (or another similar data structure) and at the end of processing return the top scoring documents for a query. Since all documents are indexed by the terms they contain, the process of generating, building, and storing document representations is called *indexing* and the resulting inverted files are called the *inverted index*.

Most IR systems use single words as terms. Words that are considered non-informative, like function words (*the, in, of, a, ...*), also called *stop-words*, are often ignored. Conflating various forms of the same word to its root form, called *stemming* in IR jargon, is also used by many systems. The main idea behind stemming is that users searching for information on `retrieval` will also be interested in articles that have information about `retrieve, retrieved, retrieving, retriever`, and so on. This also makes the system susceptible to errors due to poor stemming. For example, a user interested in `information retrieval` might get an article titled `Information on Golden Retrievers` due to stemming. Several stemmers for various languages have been developed over the years, each with its own set of stemming rules. However, the usefulness of stemming for improved search quality has always been questioned in the research community, especially for English. The consensus is that, for English, on average stemming yields small improvements in search effectiveness; however, in cases where it causes poor retrieval, the user can be considerably annoyed. [12] Stemming is possibly more beneficial for languages with many word inflections (like German).

Some IR systems also use multi-word phrases (e.g., "information retrieval") as index terms. Since phrases are considered more meaningful than individual words, a phrase match in the document is considered more informative than single word matches. Several techniques to generate a list of phrases have been explored. These range from fully linguistic (e.g., based on parsing the sentences) to fully statistical (e.g., based on counting word cooccurrences). It is accepted in the IR research community that phrases are valuable indexing units and yield improved search effectiveness. However, the style of phrase generation used is not critical. Studies comparing linguistic phrases to statistical phrases have failed to show a difference in their retrieval performance. [8]

3 Evaluation

Objective evaluation of search effectiveness has been a cornerstone of IR. Progress in the field critically depends upon experimenting with new ideas and evaluating the effects of these ideas, especially given the experimental nature of the field. Since the early years, it was evident to researchers in the community that objective evaluation of search techniques would play a key role in the field. The Cranfield tests, conducted in 1960s, established the desired set of characteristics for a retrieval system. Even though there has been some debate over the years, the two desired properties that have been accepted by the research community for measurement of search effectiveness are *recall*: the proportion of relevant documents retrieved by the system; and *precision*: the proportion of retrieved documents that are relevant.[6]

It is well accepted that a good IR system should retrieve as many relevant documents as possible (i.e., have a high recall), and it should retrieve very few non-relevant documents (i.e., have high precision). Unfortunately, these two goals have proven to be quite contradictory over the years. Techniques that tend to improve recall tend to hurt precision and vice-versa. Both recall and precision are set oriented measures and have no notion of ranked retrieval. Researchers have used several variants of recall and precision to evaluate ranked retrieval. For example, if system designers feel that precision is more important to their users, they can use precision in top ten or twenty documents as the evaluation metric. On the other hand if recall is more important to users, one could measure precision at (say) 50% recall, which would indicate how many non-relevant documents a user would have to read in order to find half the relevant ones. One measure that deserves special mention is *average precision*, a single valued measure most commonly used by the IR research community to evaluate ranked retrieval. Average precision is computed by measuring precision at different recall points (say 10%, 20%, and so on) and averaging. [27]

4 Key Techniques

Section 2 described how different IR models can implemented using inverted lists. The most critical piece of information needed for document ranking in all models is a term's weight in a document. A large body of work has gone into proper estimation of these weights in different models. Another technique that has been shown to be effective in improving document ranking is query modification via *relevance feedback*. A state-of-the-art ranking system uses an effective weighting scheme in combination with a good query expansion technique.

4.1 Term Weighting

Various methods for weighting terms have been developed in the field. Weighting methods developed under the probabilistic models rely heavily upon better estimation of various probabilities. [21] Methods developed

tf	is the term's frequency in document
qtf	is the term's frequency in query
N	is the total number of documents in the collection
df	is the number of documents that contain the term
dl	is the document length (in bytes), and
avdl	is the average document length

Okapi weighting based document score: [23]

$$\sum_{t \in Q,D} ln \frac{N - df + 0.5}{df + 0.5} \cdot \frac{(k_1 + 1)tf}{(k_1(1-b) + b\frac{dl}{avdl}) + tf} \cdot \frac{(k_3+1)qtf}{k_3 + qtf}$$

k_1 (between 1.0–2.0), b (usually 0.75), and k_3 (between 0–1000) are constants.

Pivoted normalization weighting based document score: [30]

$$\sum_{t \in Q,D} \frac{1 + ln(1 + ln(tf))}{(1-s) + s\frac{dl}{avdl}} \cdot qtf \cdot ln \frac{N+1}{df}$$

s is a constant (usually 0.20).

Table 1: Modern Document Scoring Schemes

under the vector space model are often based on researchers' experience with systems and large scale experimentation. [26] In both models, three main factors come into play in the final term weight formulation. a) Term Frequency (or tf): Words that repeat multiple times in a document are considered salient. Term weights based on *tf* have been used in the vector space model since the 1960s. b) Document Frequency: Words that appear in many documents are considered common and are not very indicative of document content. A weighting method based on this, called inverse document frequency (or *idf*) weighting, was proposed by Sparck-Jones early 1970s. [15] And c) Document Length: When collections have documents of varying lengths, longer documents tend to score higher since they contain more words and word repetitions. This effect is usually compensated by normalizing for document lengths in the term weighting method. Before TREC, both the vector space model and the probabilistic models developed term weighting schemes which were shown to be effective on the small test collections available then. Inception of TREC provided IR researchers with very large and varied test collections allowing rapid development of effective weighting schemes.

Soon after first TREC, researchers at Cornell University realized that using raw *tf* of terms is non-optimal, and a dampened frequency (e.g., a logarithmic *tf* function) is a better weighting metric. [4] In subsequent years, an effective term weighting scheme was developed under a probabilistic model by Steve Robertson and his team at City University, London. [22] Motivated in part by Robertson's work, researchers at Cornell University developed better models of how document length should be factored into term weights. [29] At the end of this rapid advancement in term weighting, the field had two widely used weighting methods, one (often called *Okapi weighting*) from Robertson's work, and the second (often called *pivoted normalization weighting*) from the work done at Cornell University. Most research groups at TREC currently use some variant of these two weightings. Many studies have used the phrase *tf-idf weighting* to refer to any term weighting method that uses *tf* and *idf*, and do not differentiate between using a simple document scoring method (like $\sum_{t \in Q,D} tf \cdot ln\frac{N}{df}$) and a state-of-the-art scoring method (like the ones shown in Table 1). Many such studies claim that their proposed methods are far superior than *tf-idf weighting*, often a wrong conclusion based on the poor weighting formulation used.

4.2 Query Modification

In the early years of IR, researchers realized that it was quite hard for users to formulate effective search requests. It was thought that adding synonyms of query words to the query should improve search effectiveness. Early research in IR relied on a thesaurus to find synonyms.[14] However, it is quite expensive to obtain a good general purpose thesaurus. Researchers developed techniques to automatically generate thesauri for use in query modification. Most of the automatic methods are based on analyzing word cooccurrence in the documents (which often produces a list of strongly related words). Most query augmentation techniques based on automatically generated thesaurii had very limited success in improving search effectiveness. The main reason behind this is the lack of query context in the augmentation process. Not all words related to a query word are meaningful in context of the query. E.g., even though `machine` is a very good alternative for the word `engine`, this augmentation is not meaningful if the query is `search engine`.

In 1965 Rocchio proposed using relevance feedback for query modification. [24] Relevance feedback is motivated by the fact that it is easy for users to judge some documents as relevant or non-relevant for their query. Using such relevance judgments, a system can then automatically generate a better query (e.g., by adding related new terms) for further searching. In general, the user is asked to judge the relevance of the top few documents retrieved by the system. Based on these judgments, the system modifies the query and issues the new query for finding more relevant documents from the collection. Relevance feedback has been shown to work quite effectively across test collections.

New techniques to do meaningful query expansion in absence of any user feedback were developed early 1990s. Most notable of these is *pseudo-feedback*, a variant of relevance feedback. [3] Given that the top few documents retrieved by an IR system are often on the general query topic, selecting related terms from these documents should yield useful new terms irrespective of document relevance. In pseudo-feedback the IR system assumes that the top few documents retrieved for the initial user query are “relevant”, and does relevance feedback to generate a new query. This expanded new query is then used to rank documents for presentation to the user. Pseudo feedback has been shown to be a very effective technique, especially for short user queries.

5 Other Techniques and Applications

Many other techniques have been developed over the years and have met with varying success. **Cluster hypothesis** states that documents that cluster together (are very similar to each other) will have a similar relevance profile for a given query. [10] Document clustering techniques were (and still are) an active area of research. Even though the usefulness of document clustering for improved search effectiveness (or efficiency) has been very limited, document clustering has allowed several developments in IR, e.g., for browsing and search interfaces. **Natural Language Processing** (NLP) has also been proposed as a tool to enhance retrieval effectiveness, but has had very limited success. [31] Even though document ranking is a critical application for IR, it is definitely not the only one. The field has developed techniques to attack many different problems like information filtering [2], topic detection and tracking (or TDT) [1], speech retrieval [13], cross-language retrieval [9], question answering [19], and many more.

6 Summing Up

The field of information retrieval has come a long way in the last forty years, and has enabled easier and faster information discovery. In the early years there were many doubts raised regarding the simple statistical techniques used in the field. However, for the task of finding information, these statistical techniques have indeed proven to be the most effective ones so far. Techniques developed in the field have been used in many other areas and have yielded many new technologies which are used by people on an everyday basis, e.g., web search engines, junk-email filters, news clipping services. Going forward, the field is attacking many critical problems that users face in todays information-ridden world. With exponential growth in the amount of information available, information retrieval will play an increasingly important role in future.

References

[1] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study: Final report. In *Proceedings of DARPA Broadcast News Transcription and Understanding Workshop*, pages 194–218, 1998.

[2] N. J. Belkin and W. B. Croft. Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM*, 35(12):29–38, 1992.

[3] Chris Buckley, James Allan, Gerard Salton, and Amit Singhal. Automatic query expansion using SMART: TREC 3. In *Proceedings of the Third Text REtrieval Conference (TREC-3)*, pages 69–80. NIST Special Publication 500-225, April 1995.

[4] Chris Buckley, Gerard Salton, and James Allan. Automatic retrieval with locality information using SMART. In *Proceedings of the First Text REtrieval Conference (TREC-1)*, pages 59–72. NIST Special Publication 500-207, March 1993.

[5] Vannevar Bush. As We May Think. *Atlantic Monthly*, 176:101–108, July 1945.

[6] C. W. Cleverdon. The Cranfield tests on index language devices. *Aslib Proceedings*, 19:173–192, 1967.

[7] W. B. Croft and D. J. Harper. Using probabilistic models on document retrieval without relevance information. *Journal of Documentation*, 35:285–295, 1979.

[8] J. L. Fagan. The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval. *Journal of the American Society for Information Science*, 40(2):115–139, 1989.

[9] G. Grefenstette, editor. *Cross-Language Information Retrieval*. Kluwer Academic Publishers, 1998.

[10] A. Griffiths, H. C. Luckhurst, and P. Willett. Using interdocument similarity in document retrieval systems. *Journal of the American Society for Information Science*, 37:3–11, 1986.

[11] D. K. Harman. Overview of the first Text REtrieval Conference (TREC-1). In *Proceedings of the First Text REtrieval Conference (TREC-1)*, pages 1–20. NIST Special Publication 500-207, March 1993.

[12] David Hull. Stemming algorithms - a case study for detailed evaluation. *Journal of the American Society for Information Science*, 47(1):70–84, 1996.

[13] G. J. F. Jones, J. T. Foote, K. Sparck Jones, and S. J. Young. Retrieving spoken documents by combining multiple index sources. In *Proceedings of ACM SIGIR'96*, pages 30–38, 1996.

[14] K. Sparck Jones. *Automatic Keyword Classification for Information Retrieval*. Butterworths, London, 1971.

[15] K. Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[16] K. Sparck Jones and P. Willett, editors. *Readings in Information Retrieval*. Morgan Kaufmann, 1997.

[17] H. P. Luhn. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, 1957.

[18] M. E. Maron and J. L. Kuhns. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM*, 7:216–244, 1960.

[19] Marius Pasca and Sanda Harabagiu. High performance question/answering. In *Proceedings of the 24th International Conference on Research and Development in Information Retrieval*, pages 366–374, 2001.

[20] S. E. Robertson. The probabilistic ranking principle in IR. *Journal of Documentation*, 33:294–304, 1977.

[21] S. E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, May-June 1976.

[22] S. E. Robertson and S. Walker. Some simple effective approximations to the 2–poisson model for probabilistic weighted retrieval. In *Proceedings of ACM SIGIR'94*, pages 232–241, 1994.

[23] S. E. Robertson, S. Walker, and M. Beaulieu. Okapi at TREC–7: automatic ad hoc, filtering, VLC and filtering tracks. In *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*, pages 253–264. NIST Special Publication 500-242, July 1999.

[24] J. J. Rocchio. Relevance feedback in information retrieval. In Gerard Salton, editor, *The SMART Retrieval System—Experiments in Automatic Document Processing*, pages 313–323, Englewood Cliffs, NJ, 1971. Prentice Hall, Inc.

[25] Gerard Salton, editor. *The SMART Retrieval System—Experiments in Automatic Document Retrieval*. Prentice Hall Inc., Englewood Cliffs, NJ, 1971.

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

[27] Gerard Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw Hill Book Co., New York, 1983.

[28] Gerard Salton, A. Wong, and C. S. Yang. A vector space model for information retrieval. *Communications of the ACM*, 18(11):613–620, November 1975.

[29] Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted document length normalization. In *Proceedings of ACM SIGIR'96*, pages 21–29. Association for Computing Machinery, New York, August 1996.

[30] Amit Singhal, John Choi, Donald Hindle, David Lewis, and Fernando Pereira. AT&T at TREC-7. In *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*, pages 239–252. NIST Special Publication 500-242, July 1999.

[31] T. Strzalkowski, L. Guthrie, J. Karlgren, J. Leistensnider, F. Lin, J. Perez-Carballo, T. Straszheim, J. Wang, and J. Wilding. Natural language information retrieval: TREC-5 report. In *Proceedings of the Fifth Text REtrieval Conference (TREC-5)*, 1997.

[32] Howard Turtle. *Inference Networks for Document Retrieval*. Ph.D. thesis, Department of Computer Science, University of Massachusetts, Amherst, MA 01003, 1990. Available as COINS Technical Report 90-92.

[33] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 1979.