Inf Retrieval (2006) 9:311–330
DOI 10.1007/s10791-006-1682-6

Evolving local and global weighting schemes in information retrieval

Ronan Cummins · Colm O'Riordan

Received: 25 September 2004 / Revised: 11 February 2005 / Accepted: 25 April 2005

Abstract This paper describes a method, using Genetic Programming, to automatically determine term weighting schemes for the vector space model. Based on a set of queries and their human determined relevant documents, weighting schemes are evolved which achieve a high average precision. In Information Retrieval (IR) systems, useful information for term weighting schemes is available from the query, individual documents and the collection as a whole.

We evolve term weighting schemes in both local (within-document) and global (collection-wide) domains which interact with each other correctly to achieve a high average precision. These weighting schemes are tested on well-known test collections and are compared to the traditional *tf-idf* weighting scheme and to the BM25 weighting scheme using standard IR performance metrics.

Furthermore, we show that the global weighting schemes evolved on small collections also increase average precision on larger TREC data. These global weighting schemes are shown to adhere to Luhn's resolving power as both high and low frequency terms are assigned low weights. However, the local weightings evolved on small collections do not perform as well on large collections. We conclude that in order to evolve improved local (within-document) weighting schemes it is necessary to evolve these on large collections.

Keywords Genetic Programming · Information Retrieval · Term-Weighting Schemes

1. Introduction

The ability to retrieve information based on a user's need has become increasingly important with the emergence of the World Wide Web and the huge increase in information available

R. Cummins (✉)
Department of Information Technology, National University of Ireland, Galway, Ireland
e-mail: ronan.cummins@nuigalway.ie

C. O'Riordan
Department of Information Technology, National University of Ireland, Galway, Ireland
e-mail: colmor@it.nuigalway.ie

on-line. Despite advances in web retrieval, problems with the retrieval of relevant information exist. The complexity of the problem is further increased by the fact that more and more of this information appears in natural language and not in structured formats. Information Retrieval (IR) is primarily concerned with the retrieval of information rather than data. The study of IR techniques has increased greatly since the advent of the World Wide Web and many paradigms and models have been introduced to help solve the IR problem.

A very popular IR model used in recent years has been the vector space model (Salton et al., 1975). It has been recognized that the effectiveness of the vector space model depends crucially on the term weights applied to the terms of the document vectors (Salton and Buckley, 1988). These term weights are found using a term weighting scheme based on the frequency of the terms in the document and the collection. Terms that occur more often in a document are treated as more important, i.e. they better describe the document content, and thus are given a higher weight. Terms that occur less frequently throughout a collection are given a higher weight because they are deemed better able to distinguish between documents. Measures derived from knowledge of the document are regarded as local measures, and measures derived from knowledge of the collection are regarded as global measures. This distinction is beneficial in the analysis of weighting schemes. The number of ways these measures can be combined is vast, even using a small number of operators (e.g. $+$, $-$, $\times$, and $/$).

Genetic Programming (Koza, 1992) is a heuristic stochastic searching algorithm, inspired by natural selection (Darwin, 1859), efficient for navigating large complex search spaces. In the Genetic Programming paradigm, solutions are initially created at random. These solutions are evaluated and the fitter solutions are chosen to undergo crossover and mutation to create new solutions. These new solutions undergo the same process for a number of generations. Genetic Programming is efficient for searching large spaces because it probes, in parallel, many points in the search space. The advantage of this evolutionary approach is that it can help solve problems that are extremely complex or problems where the traditional algorithm is computationally infeasible. In modelling a genetic program, often all that is required to know is how to evaluate a solution to a problem. In many cases this is a difficult problem in itself. Genetic Programming helps in problems where the roles of variables are not fully understood. Genetic Programming can be used to automatically derive functions whose variables combine and react in complex ways.

This paper presents a framework based on the vector space model that allows different weighting schemes to be evaluated. Genetic Programming is used to selectively breed fitter schemes. We present weighting schemes evolved in a global domain that achieve a high average precision and that are consistent with Luhn's theory of resolving power. Local weighting schemes are evolved which are combined with the global schemes to create a complete weighting scheme for the vector space model. We analyse these weighting schemes in both a local and global domain and compare them to the traditional *tf-idf* and BM25 (Robertson et al., 1998) schemes on standard test collections.

Section 2 introduces background information in the areas of Information Retrieval and Genetic Programming. Existing approaches applying evolutionary computation to IR are also reviewed in this section. Section 3 discusses the design of the framework and the design of the experiments. The aims of the experiments are also outlined in this section. The results and analysis of each experiment are outlined in Section 4 and finally, our conclusions and proposed future work are discussed in Section 5.

2. Background

2.1. Information retrieval

This section presents background material on the vector space model. Term weighting schemes are introduced and Luhn's (1958) and Zipf's (1949) contributions to IR are summarised.

The vector space model is one of the most widely known and studied IR models. The classic vector space model represents each document in the collection as a vector of terms with weights associated with each term. The weight of each term is based on the frequency of the term in the document and collection. The query (user need) is also modelled as a vector. A matching function is used to compare each document vector to the query vector. Once the documents are compared, they are sorted into a ranked list and returned to the user. The *tf-idf* family of weighting schemes (Salton and Buckley, 1988) is one of the most widely used weighting schemes for the vector space model. The term-frequency *(tf)* is a document specific local measure and is calculated as follows:

$$tf = \frac{rtf}{max\_freq} \tag{1}$$

where *rtf* is the raw term frequency and the *max_freq* is the frequency of the most common term in the document. The *max_freq* measure is used as the normalisation factor because longer documents tend to have more terms and hence higher raw term frequencies (*rtf*). The idf part of the weighting scheme, first introduced by Sparck Jones (1972), is an Inverse Document Frequency measure. The main heuristic behind the idf factor is that a term that occurs infrequently is good at discriminating between documents. The *idf* of a term is a global measure and is determined in the *tf-idf* solution as follows:

$$idf_t = \log\left(\frac{N}{df_t}\right) \tag{2}$$

where N is the number of documents in the collection and df_t is the number of documents in which the term t appears. The weight then assigned to a term is a product of *tf* and *idf*.

The Okapi-BM25 weighting scheme, developed by Robertson et al. (1998), is a weighting scheme based on the probabilistic model. Okapi-*tf* is calculated as follows:

$$Okapi\text{-}tf = \frac{rtf}{rtf + k_1\left((1-b) + b\frac{l}{l_{avg}}\right)} \tag{3}$$

where *rtf* is the raw term frequency and l and l_{avg} are the length and average length of the document vectors respectively. k_1 and b are tuning parameters. The *idf* of a term as determined in the BM25 formula is as follows:

$$BM25\text{-}idf_t = \log\left(\frac{N - df_t + 0.5}{df_t + 0.5}\right) \tag{4}$$

The weight assigned to a term in the BM25 scheme is a product of Okapi-*tf* and BM25-*idf*. The BM25 scheme has been shown to achieve a higher average precision than *tf-idf* on

large document collections. The standard matching function for the BM25 scheme can be described as the inner-product (Li et al., 2002).

$$\sum_{t \in q \cap d} (Okapi\text{-}tf \times BM25\text{-}idf_t \times qrtf) \tag{5}$$

where *qrtf* is the raw term frequency in the query and t is a term in the query q and document d.

Another modern matching function is the pivoted document length normalisation scheme. This scheme is often calculated as follows (Singhal, 2001; Singhal et al., 1996):

$$\sum_{t \in q \cap d} \left(\frac{1 + \log(1 + \log(rtf))}{(1 - s) + s \frac{l}{l_{avg}}} \times \log\left(\frac{N+1}{df_t}\right) \times qrtf \right) \tag{6}$$

where s is the slope and is a constant value of usually about 0.2.

Zipf (1949) recognized that the frequency of terms in a collection when placed in rank order approximately follows a log curve. The Zipfian distribution for terms in a collection states that the product of the rank order of terms in a collection and their frequencies is approximately constant. Luhn (1958) further proposed that terms that occur too frequently in a collection have little power to distinguish between documents and that terms that appear infrequently are also of little use in distinguishing between documents. Luhn used the Zipfian characteristics to devise 2 cut-off points for determining terms with a high resolving power. Terms that appear outside these points are considered as having a low distinguishing (resolving) power. Thus in Figure 1 (Schultz, 1968), the bell-shaped curve in the graph relates the frequency of terms to their resolving power.

Salton and Yang (1973) validate much of Luhn's and Zipf's work with empirical analysis. The standard *tf-idf* weighting scheme stems from these ideas. As previously discussed the *tf* part of the scheme identifies terms that appear more frequently as more important within a document (i.e. on a local level). This curve would follow the Zipfian curve in a local context. The *idf* part of this scheme weights the high frequency terms lower on a global scale. Thus, the curve of the *idf* measure follows an inverse of the Zipfian curve in Figure 1. Yu and Salton (1976) suggest that the best distinguishing terms are terms that occur with a high frequency in certain documents but whose overall frequency across a collection is low (low document frequency). They conclude from this that term weighting should vary directly with term frequency and inversely with document frequency. However, these weighting schemes

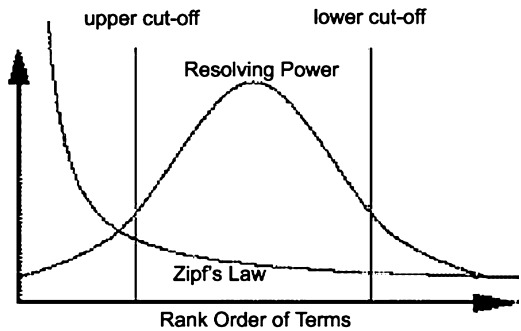

Fig. 1 Zipf's law and Luhn's proposed cut-off points

are not fully consistent with Luhn's observations that the resolving power of terms with a low frequency is also low.

2.2. Genetic programming

Inspired by the successes in traditional Genetic Algorithms (GA), John Koza (1992) developed Genetic Programming (GP) in the early 1990s. The GP approach has helped solve problems in a wide variety of areas. GP is inspired by the Darwinian theory of natural selection (1859), where individuals that have a higher fitness will survive longer and thus produce more offspring. These offspring will inherit characteristics similar to that of their parents and through successive generations, the useful characteristics will survive. GP can be thought of as an artificial way of selective breeding. In GP, solutions are encoded as trees with operators (functions) on internal nodes and operands (terminals) on the leaf nodes. These nodes are often referred to as genes and their values as alleles. The coded version of a solution is called its genotype, as it can be thought of as the genome of the individual, while the solution in its environment is called its phenotype. The fitness is evaluated on the phenotype of a candidate solution while reproduction and crossover is performed on the genotype.

The basic flow of a GP is shown in Figure 2. Initially, a random population of solutions is created. These solutions are encoded as trees. The depth of each tree determines the maximum size or length of the solution. Each solution is rated based on how it performs in its environment. This is achieved using a fitness function. Having assigned the fitness values, selection can occur. Goldberg (1989) uses the roulette wheel example where each solution is represented by a segment on a roulette wheel proportionately equal to the fitness of the solution to explain how selection occurs. Thus, solutions with a higher fitness will produce more offspring. Tournament selection is the most common selection method used. In tournament selection, a number of solutions are chosen at random and these solutions compete with each other. The fittest solution is then chosen as a parent. The number of solutions chosen to compete in the tournament is the tournament size and this can be increased or decreased to increase or decrease the speed of convergence. Once selection has occurred,

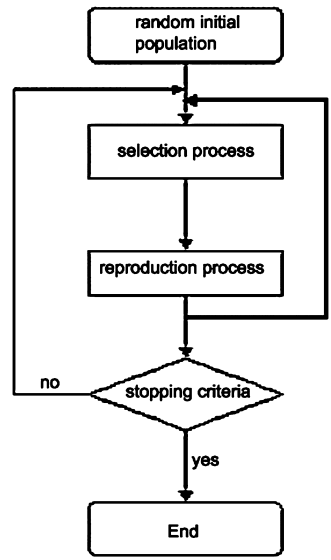

Fig. 2 Flow of a basic GP

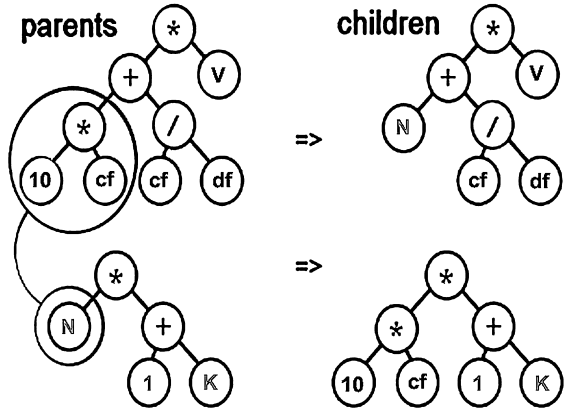

Fig. 3 Example of crossover in GP

reproduction can start. Reproduction (recombination) can occur in variety of ways. The most common form is sexual reproduction, where two different individuals (parents) are selected and two separate children are created by combining the genotypes of both parents. Mutation (asexual reproduction) is the random change of an allele of a gene to create a new individual. Selection and recombination occurs until the population is replaced by newly created individuals. Usually, the number of solutions from generation to generation remains constant. Once the recombination process is complete, each individual's fitness in the new generation is evaluated and the selection process starts again. The process usually ends after a predefined number of generations, once convergence of the population is achieved or after an individual is found with an acceptable fitness.

2.3. Existing evolutionary approaches to IR

There have been several approaches that apply ideas from evolutionary computation to the domain of Information Retrieval. These approaches can be broadly categorized as follows: evolving weights or weighting schemes for existing IR models; learning relationships between terms and document features; evolving optimal queries for a given information need; and learning optimal weighting of tags in semi-structured documents.

Oren (2002) presents interesting work where a genetic programming technique is used to evolve weighting functions, which outperform the traditional *tf-idf* weighting scheme in a vector space framework. Fan, Gordon and Pathak (2004b) also adopt a genetic programming approach to evolving new matching functions. These approaches evolve useful weighting schemes. However, they lack a detailed analysis of the weighting schemes presented. Fan et al. (2004a) explore different fitness functions for use in a similar framework. Gordon (1988) adopts a genetic algorithm to modify document representations (a set of keywords) based on user interaction. Vrajitoru (2000) models the whole document collection as the genome of an individual and evolves a better representation for the whole collection.

Term-oriented models focus on finding the best discriminatory terms for improved document retrieval. For term-oriented approaches, a query is evolved so that the optimal document set is returned for a query. Horng and Yeh (2000) use this approach to extract keywords from a subset of relevant documents to construct this query and then adapt the weights to best suit the relevant documents. This approach is useful in building user profiles so that the system can learn a set of optimal terms that best describe the user's needs. Yang and Korfhage

(1993) adopt a genetic algorithm approach to modify document representations by altering the weights associated with keywords.

Query expansion and adaptation techniques, resulting in better retrieval performance have been developed through evolutionary computation techniques. Vrajitoru (1998) uses genetic algorithms to evolve queries and introduces a modified crossover operator to create new queries. Exploiting relationships between terms and features of documents has been used in information retrieval in weighting query and document features to improve performance. These relationships are often defined using knowledge of a language or derived empirically by analysis of the document collection. Work by Bergstrom et al. (2000) attempts to automatically evolve patterns for relation extraction from collections of web pages.

In information retrieval, potentially useful information is available from document tags in collections of semi-structured (e.g. html) documents. Several information retrieval systems exist which pay more attention to content associated with certain tags (e.g. title, author, keywords). Kim and Zhang (2001) attempt to learn the optimal set of tags, and their associated weights, using a genetic algorithm and demonstrate an improvement in retrieval performance.

3. Design and experimental setup

This section describes the experimental setup, outlines the aims of each of the planned experiments and outlines the novelty of this approach.

3.1. Document test collections

The three small document collections used in this research are the Medline, CISI and Cranfield collections.[1] These small collections are used for testing and training. The NPL collection is a medium sized collection available from the same source. The larger document collections used are subsets of the TREC-9 filtering track (OHSUMED collection (Hersh et al., 1994)).[2] This collection consists of abstracts from the Medline database from 1988 to 1991. The first subset consists of 70,825 documents from 1988 (OHSU88). The second collection consists of the documents from 1989 (OHSU89). The largest subest consists of documents from 1990 and 1991 which we call OHSU90-91. Each collection consists of 63 queries although 2 queries have no relevant documents from 1988. We ignore queries that have no relevant documents associated with them. The relevance assessments for the OHSUMED collection are graded as *definitely* or *possibly* relevant. We make no distinction between *definitely* and *possibly* relevant documents in our tests and regard both grades as relevant. All documents and queries are pre-processed by removing standard stop-words from the Brown corpus[3] and are stemmed using Porter's stemming algorithm (Porter, 1980). Table 1 shows some characteristics of the test collections used in this research.

3.2. Training times

Due to the nature of the GP approach, efficiency is of prime concern. Typically for the GP, thousands of weighting schemes need to be evaluated over a document collection for

[1] ftp://ftp.cs.cornell.edu/pub/smart

[2] http://trec.nist.gov/data/t9_filtering.html

[3] http://www.lextek.com/manuals/onix/stopwords1.html

Table 1 Characteristics of document collections

Collection	Docs	Qrys	Vocab	Qry_terms	Avg_doc_len	Avg_qry_len
Medline	1,033	30	10,975	249	56.8	11
Cranfield	1,400	225	9,014	639	59.6	8.8
CISI	1,460	76	8,342	1024	47.8	26.8
NPL	11,429	93	7,759	331	18.78	6.78
OSHU88	70,825	61	175,021	195	49.40	5.05
OSHU89	74,869	63	185,304	197	50.45	4.97
OHSU90-91	148,162	63	287,807	197	52.87	4.97

many queries. For example, consider using the smallest document collection (Medline) as the training set, and choosing an initial population of 1000 solutions running for 50 generations. In this case, 50,000 weighting schemes need to be evaluated. Each evaluation requires the processing of 30 queries over 1033 separate documents. Thus, the system will process 1.5 million queries on the collection of 1033 documents. This requires searching and determining the relevance of over 1.5 billion documents. This test typically takes 6 hours on the Medline collection using a standard desktop PC with a 2.0 GHz processor and 500Mbs of RAM. Tree depth and query length also contribute to the training time as longer solutions and queries take longer to evaluate.

3.3. GP parameters

All experiments are run for 50 generations with an initial population of 1000. It was seen in our prior tests that when using the largest terminal and function set, the population converges before 50 generations. Tournament selection is used and the tournament size is set to 10. The solutions are trained on an entire collection and tested for generality on the collections that are not included in training. The depth of the tree for each solution is limited to 6 (unless otherwise specified) to improve the generality of the solutions because shorter solutions are usually more general (Kuscu, 2000). This depth allows a large enough solution space to be searched in order to obtain high quality solutions. The creation type used is the standard ramped half and half creation method used by Koza (1992). In the ramped half and half creation method, half the trees initally created are of full depth and the other half created have depths from 1 up to the full depth. This produces a initial population of individual solutions of varying tree depth and shape. Elitism is used, i.e. the best individual from the current generation is copied into next generation automatically. No mutation is used in our experiments. Due to the stochastic nature of GP, a number of runs is often needed to show that the GP is converging to its best solution. We perform a number of runs of each experiment and choose the best individual from those runs. The best evolved solutions presented in the results section in this paper are simplified and re-written to aid the readability and analysis of the weighting schemes. However, the solutions presented are functionally equivalent to those output by the GP process.

3.4. Fitness function

The average precision (AP), used as the fitness function, is calculated for each scheme by comparing the ranked list returned by the system against the human determined relevant

documents for each query. Average precision is calculated using precision values for all points of recall. This is frequently used as a performance measure in IR systems.

3.5. Benchmark term-weighting schemes

We test our evolved scheme against the Okapi-BM25 scheme introduced earlier (2) (3). We use the default parameters of $b = 0.75$ and $k_1 = 1.2$. Typical values for k_1 range between 1 and 2. We also use the pivoted normalisation scheme (Piv) (6) with a slope of $s = 0.2$. We include the traditional *tf-idf* scheme for comparsion. On the collections used in this research we have found BM25 with $k_1 = 1.2$ to be the best performing scheme.

3.6. Terminal and function sets

Tables 2 and 3 show the full terminal and function set used in our experiments. It is worth noting that all the measures included are in a primitive (unprocessed) form as our intention is to allow the GP to combine these primitive measures in an unbiased way. The global and local terminals are indicated in the table. Document normalisation measures are shown as local measures as their influence acts in the local (document) domain.

Table 2 Terminal set

Terminal	Description	Domain
1	*the constant* 1	both
rtf	raw term frequency within a document	local
l	document length (no. of unique words in a document)	local
df	no. of documents a term appears in	global
N	no. of documents in a collection	global
max_freq	frequency of the most common term in a document	local
tl	total document length (no. of words in a document)	local
V	vocabulary of collection (no. of unique terms in the collection)	global
C	collection size (total number of terms in the collection)	global
cf	collection frequency (frequency of a term in the collection)	global
max_c_freq	frequency of the most common term in the collection	global

Table 3 Function Set

Function	Description
$+, \times, /, -$	addition, multiplication, division and subtraction functions
log	the natural log
sin, tan	trigonometric functions
$\sqrt{\ }$	square-root function
sq	square

3.7. Size of search space

The GP approach adopted in this work evolves the weighting scheme over a number of generations. An initial population is created randomly by combining a set of primitive measures (e.g. df_t, rtf, N) using a set of operators (e.g. $+, -, \times, /$). The number of different binary tree shapes with n internal nodes can be found using the Catalan number $C_n = (2n)!/(n!(n+1)!)$ (Lucas et al., 1993). For every full tree of $2n+1$ nodes, the total number of unique trees creates an enormous space of $C_n \times f^n \times t^{(n+1)}$ programs, where f is the number of functions and t is the number of terminals. Some of these trees may be functionally equivalent or may be equivalent based on the fitness function (Gustafson, 2004).

3.8. Experimental design

The aim of the first experiment is to show that general weighting schemes can be evolved whose performance is better than the *tf-idf* scheme and comparable to, or even better than, the more modern BM25 weighting scheme on small collections. In this experiment schemes are evolved on all 3 small collections. This experiment evolves solutions using all of the functions and terminals presented in Table 2 and Table 3.

The aim of the second experiment is to empirically measure the benefit, in terms of average precision, of each terminal presented in Table 2. We aim to show that some terminals have an obvious benefit when included in weighting schemes, while others have little or no benefit and may be disregarded in future work.

In the third and fourth experiments we combine the sources of evidence (terminals) from Table 2 in their respective domains (i.e global and local). Firstly, we evolve the global weight (gw_t) with a binary weighting on the local (within-document) weighting (lw_t). This is done so that significant terms, i.e. terms which aid retrieval, are promoted. Then, we evolve a local weighting dependant on the best performing global weight evolved. This is done so that the merit of the schemes can be correctly analysed. Thus, the local and global weighting schemes are evolved seperately in the following function:

$$w_t(d_i, q) = \sum_{t \in q \cap d} (lw_t \times gw_t \times qrtf) \tag{7}$$

where lw_t is the local weight (within-document weight), gw_t is the global weight (resolving power). The weighting scheme applied to the query terms is a simple actual term frequency weighting scheme. This query weighting is applied to all weighting schemes used in this paper. This separation of the weighting schemes into their respective parts will help uncover any important characteristics of the evolved weighting schemes.

3.9. Novelty of approach

The underlying framework adopted here is similar to those adopted by Oren (2002) and Fan et al. (2004b). However, there are some fundamental differences in the aim of the experiments. While both Oren and Fan et al. have evolved weighting schemes that show an increase in average precision over standard weighting schemes, reasons for the increase are not presented. Furthermore, the weighting schemes presented are difficult to analyse. In our approach, we gather a set of text features and provide a means of empirically measuring the benefit of each of the features with respect to relevence. This process can help to locate where (i.e. in what text features) certain relevence information may lie and can also eliminate certain sources of

information as useless. The evolution of the weighting schemes in both a local and global domain is also important in the analysis of such weighting schemes. This two step process reduces the size of the search space and also provides a means of analysis against standard *tf-idf* type solutions. The overall aims of our approach are to evolve a full weighting scheme in IR using GP and to provide a comprehensive analysis of the schemes presented.

4. Experimental results and analysis

4.1. Full function and terminal set

In this first experiment we use all of the measures from Table 2 and all of the operators from table III. The depth of the solutions is initally set to 6. This allows a sufficiently large enough search space to be searched. The training set used is the Medline collection. The following is the best formula evolved after 4 runs of the GP with an initial population of 1000 for 50 generations:

$$w_t = \frac{\frac{cf}{df} \times \left(\log(rtf) + \frac{cf}{df} \right)}{2df + l + rtf} \tag{8}$$

Table 4 shows the differences in average precision for the best solution when evolved on each document collection and tested on the other collections.

Firstly, it can be seen that the evolved weighting schemes show a significant improvement in average precision over the BM25 solution on the Medline collection. It can also be seen that the solutions achieve an average precision on the collections on which they were not trained, which is within 1% to 2% of the best solution found on the collection on which they were trained (underlined). This shows that the solutions evolved are quite general and exploit general natural language characteristics. The *cf* measure is seen to occur consistently in the fitter solutions evolved on all 3 collections.

Figure 4 shows the increase in average precision for the best individual and average of the population over the 50 generations for a population of 1000 for a typical run of the GP on the Medline collection. In all 4 tests conducted on the Medline collection the AP of the best solution from the randomly created population (0^{th} generation) does not exceed 51% average precision and is more often well below this. Population sizes of 200 and 500 can often produce similar solutions. However, to maximise the quality of the best solution obtained from a single run it is often beneficial to use a large population. Due to the fact that our collection sizes are small we can often increase the population size and still evolve a solution

Table 4 AP for best solutions found on each collection

						Training set		
Collection	Docs	Qrys	*tf-idf*	Piv	*BM25*	CISI	Medline	Cranfield
CISI	1,460	76	20.87%	22.13%	22.67%	<u>25.47%</u>	24.86%	24.03%
Medline	1,033	30	48.96%	52.65%	53.47%	56.74%	<u>58.85%</u>	56.69%
Cranfield	1,400	225	37.75%	41.77%	42.08%	41.33%	41.85%	<u>43.04%</u>

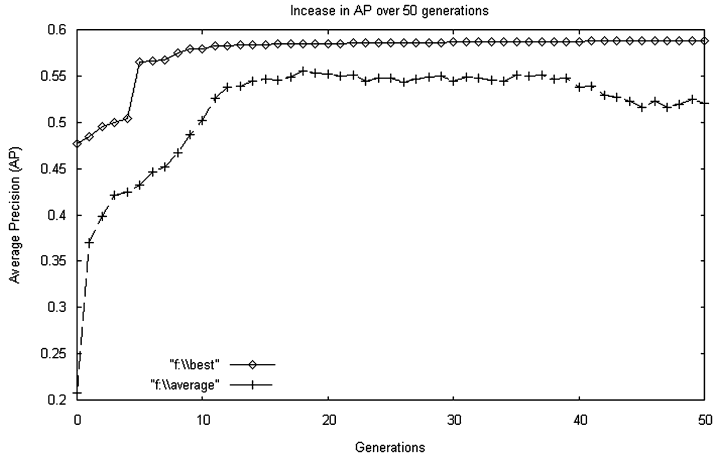

Fig. 4 Fitness of Best and Average solutions over 50 generations

in a reasonable time. However, for larger TREC style collections smaller population sizes will have to used in order to evolve a solution in an reasonable time.

4.2. Empirically measuring the benefit of sources of evidence

The aim of this experiment is to empirically measure the benefit, in terms of average precision, of each of the terminals (measures) in Table 2. A measure may have a beneficial but complex interaction with other measures which other techniques may not be able to uncover. GP can often find interactions and relationships that are not apparent. The experiment starts with a limited terminal set and evolves a solution. Terminals are added one by one and for each terminal set, solutions are evolved. As the number of terminals increase we examine the change in average precision of the best solution. Terminals are added in the order they appear in table II. Three runs of the GP for each terminal set is conducted and the best solution from the three runs is chosen and its average precision is plotted. The Medline collection is used as the training set in this experiment as we have seen a marked increase in average precision over other schemes on this collection.

Figure 5 shows the average precision for the best solution from the three runs of the GP for each terminal set. As the terminal set increases in size, the average precision also increases as more measures of the document and collection become available. Each bar is labelled with the terminal that was added at that point. Thus, the bar on the far right is the best solution found for the full terminal set. The bar on the far left can be used as a benchmark because it represents a simple binary weighting scheme. This binary weighting scheme shows quite a high average precision for this collection. However, on the larger OHSU90-91 collection this binary weighting is significantly lower at roughly 11.5%.

We can see in Figure 5 that when the cf measure is added to the terminal set the average precision increases by roughly 4%. We can also see that there is a marked increase in average precision for each terminal added up until the max_freq terminal is added. At this point it is unclear whether the cf measure alone is responsible for the increase in precision or if it combines favourably with previously introduced measures. The experiment is run again with the cf measure as the sixth terminal added to the terminal set. All the terminals used in this further test have shown a marked improvement when included in the terminal set (see

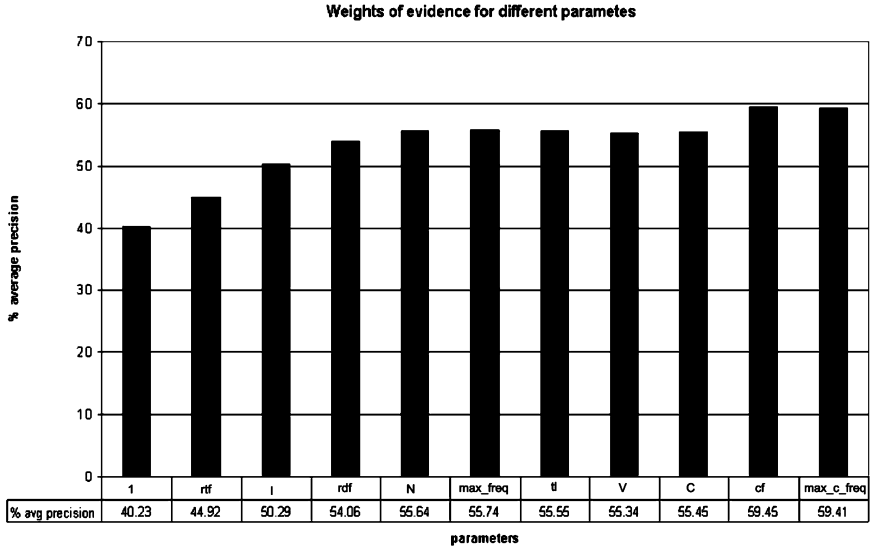

Fig. 5 Empirical benefit in terms of AP for measures

Figure 5). This is done to rule out the other terminals as being advantageous in terms of average precision. Figure 6 shows the cf measure added as the sixth terminal. From this experiment it can be seen that the cf measure is an important measure in determining the relevence of documents in this collection. We can also see that these weighting schemes can be reduced to a combination of the terminals, 1, rtf, l, df, N and cf while still maintaining high average precision values. In the solution presented in the first experiment, the cf/df combination appears in the best solution a number of times. This combination determines the average number of times a term appears in the documents which contain that term. This

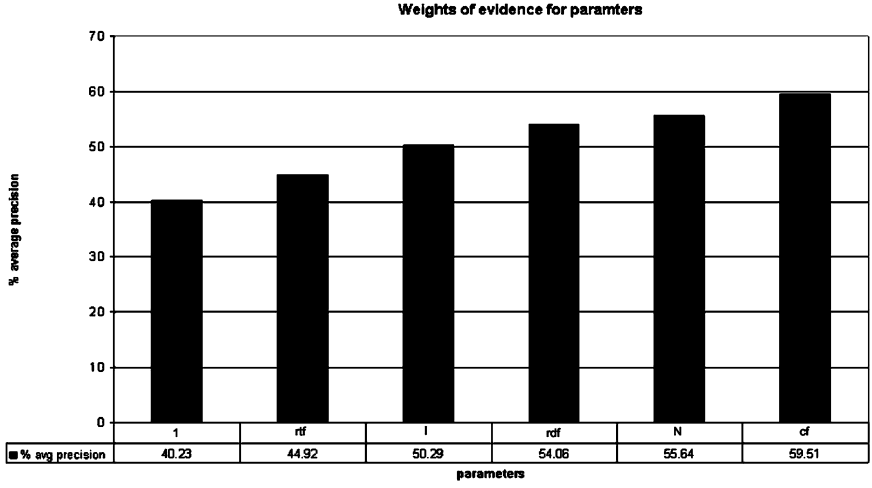

Fig. 6 Empirical benefit in terms of AP for 6 terminals

Table 5 AP for *idf*, *BM25-idf* and gw_t

Collection	Docs	Qrys	*idf*	*BM25-idf*	gw_t
CISI	1460	76	18.70%	18.76%	22.25%
Medline	1033	30	46.78%	47.00%	54.09%
Cranfield	1400	225	33.63%	33.64%	37.06%
NPL	11,429	93	25.67%	25.66%	28.89%
OHSU88	70,825	61	25.74%	25.75%	27.83%
OHSU89	74,869	63	26.03%	26.06%	27.70%
OHSU90-91	148,162	63	21.13%	21.18%	25.13%

measure is investigated in more detail in the global weighting schemes evolved in the next experiment.

4.3. Evolving global schemes

The aim of this experiment is to evolve solutions in the global domain with a binary weighting for the local part of the scheme (lw_t). The properties of the cf measure will also be investigated as the terminals used in this experiment are limited to cf, df, 1 and N. The solutions are limited to a depth of 6. The training collection used is the CISI collection as the query set is significantly larger (see Table 1) and thus has more terms. It is important that the weighting scheme work for a large sample of terms. We also test these global schemes on the larger OHSUMED collection. The following is one of the best solutions evolved on the CISI collection using only the global measures previously mentioned:

$$gw_t = \frac{\log(N/df)}{\sqrt{df}} \times \log\left(\frac{cf}{df}\right) \times \log(df) \tag{9}$$

Table 5 shows the average precision for this solution on the CISI training collection and the five collections that were not included in training. We can see that the global scheme evolved shows a significant increase in average precision over *idf* type solutions on all seven collections regardless of their size.

When the terms in the collection are placed in rank order, the gw_t weight of these terms is similar to that which Luhn predicted would lead to identifying terms with a high resolving power. More recently, Greiff (1998) also concluded that a flattening of the *idf* measure at both high and low frequencies would result in increased precision. The reason for the flattening of the curve at low frequency levels in our evolved solution is the presence of the cf measure. Figures 7 and 8 show the terms placed in rank order and the gw_t weight assigned to each term for the CISI training collection and the larger OHSU88 collection.

An interesting point to note is that the global weighting scheme identified (9) completely ignores terms that occur once or twice in a collection. This weighting scheme completely eliminates terms that are totally concentrated in one document. It also eliminates terms that occur exactly once in every document in which they appear (i.e. where $cf = df$) and thus whose concentration is very low. This has the effect of considerably reducing the size of the vocabulary of the collection. Of the 8,342 terms in the CISI collection only 1,782 (around 21%) are assigned a non-zero weight under the evolved scheme. The remaining 6,560 are assigned a weight of zero and are effectively removed from the collection. For the larger OHSU88 collection only 32,774 of the 175,021 terms recieve a non-zero weight. Thus,

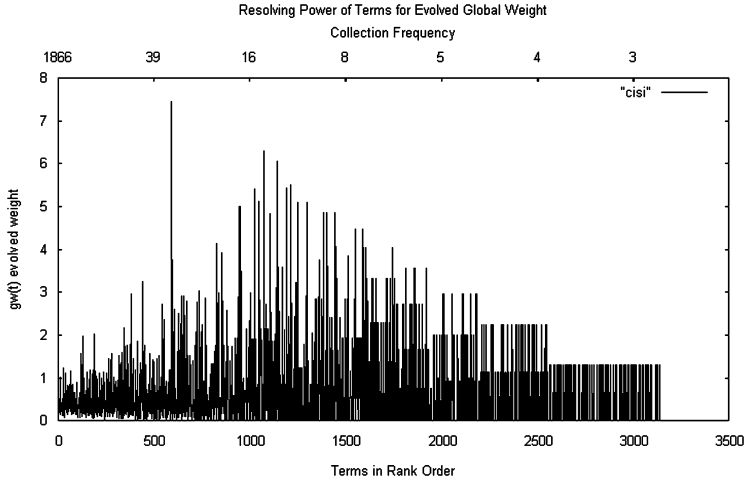

Fig. 7 gw_t for terms placed in rank order for the CISI collection

effectively we use less than 20% of the terms in the corpus after preprocessing and still see an increase in average precision. This efficient use of term indexes may be of benefit in certain feature extraction techniques that need a reduced feature space. It is interesting that these characteristics are identified by evolutionary techniques to be advantageous, as they are used in some feature extraction techniques like document-frequency thresholding (Lewis, 1992). The fact that many low frequency terms are completely eliminated by our evolved global scheme is not advocated in particular and is rather an observation of the evolved scheme. However, it is again confirmed, by evolutionary techniques, that the usefulness of such terms is low for general queries. It is also interesting that global schemes evolved on a small collection can be directly applicable to larger collections. We believe this is because the

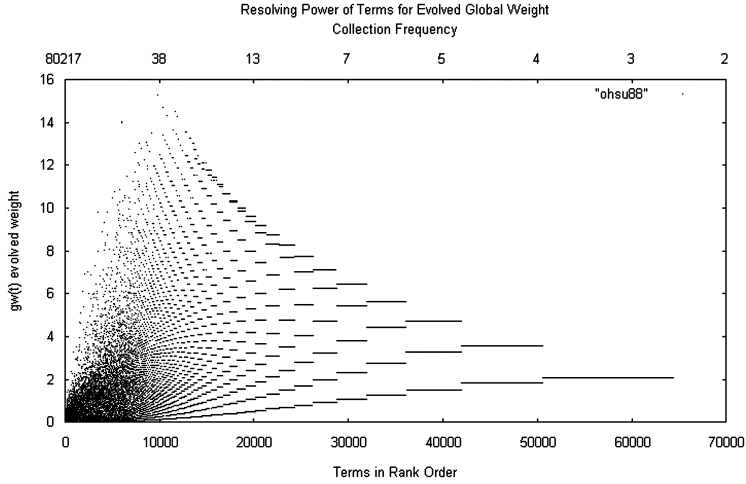

Fig. 8 gw_t for terms placed in rank order for the OHSU88 collection

small collections represent a sufficiently large sample of text in a global context to accurately represent general characteristic for document and collection frequencies.

The *idf* scheme can often incorrectly identify terms with a low frequency as having a high resolving power. In formulating a scheme for term weighting, terms within the collection with a high resolving power must firstly be identified. Then, documents containing these terms can be examined on a local level so as to distinguish these documents from each other as best as possible. The cf measure can weight low frequency terms lower in a global context. By combining the *idf* and cf measures, terms of a high resolving power can be better identified. These two measures combine together, acting somewhat like a filter which rejects both high and low frequency terms. This is in fact what the evolutionary technique has discovered as the cf/df combination presents us with a global scheme that closely adheres to Luhn's theory. The cf/df measure *consistently* appears in all the fitter solutions found on the training data used in this research. This cf/df measure was first introduced by Kwok (1996) and used to improve the performace of short queries. Pirkola and Jarvelin (2001) also use this measure to improve the resolution power of search keys.

Low frequency terms are poor discriminators because they cannot distinguish between as many documents as higher frequency terms. For example, consider a query with 3 terms ($t1$, $t2$ and $t3$). If each term in the query has a document frequency of 1 (i.e. only one document containing the term), the maximum number of different sets of documents that can be distinguished by the query is 4 (3 of the sets containing only one document), regardless of the frequency of the term within the document. However, if the document frequency of all 3 terms is greater than 4, the maximum number of different sets of documents that can possibly be distinguished increases to 8. Thus, if a query has k terms, the terms with the highest resolving power have a document frequency of $N - k \geq df_t \geq 2^{k-1}$ where df_t is the document frequency of a term and N is the number of documents in the collection. In Figure 9, C represents the document collection and the sub-sets represent document sets that contain terms with the same document frequency.

When the document frequency is as low as 1, a Boolean model for retrieval is approached, as if the term occurs it is deemed relevant, otherwise it is deemed irrelevant. This type of coarse granularity leads to poor performance, as there is a poor notion of partial matching. Terms that lead to a more granular scheme should be given a higher weight. However, if the document frequency approaches the number of documents in a collection (N), the number of document sets that can be distinguished also decreases. At an extreme case (i.e. for very frequent terms), the document frequency equals the number of documents in a collection. In this case, there is no distinguishing between any documents in the collection on a global level. This is worse than when the document frequency is at its lowest (i.e. 1). The evolved global

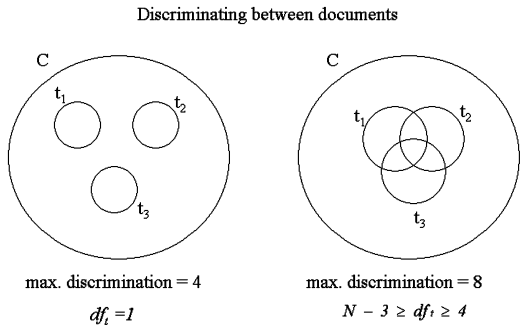

Fig. 9 Measuring the discriminating power of terms

scheme presented has many of the characteristics for identifing terms of a high resolving power. Van Rijsbergen (1979) summarizes these as follows "A term with high total frequency of occurrence is not very useful in retrieval irrespective of its distribution. Middle frequency terms are most useful particularly if the distribution is skewed. Rare terms with a skewed distribution are likely to be useful but less so than the middle frequency ones. Very rare terms are also quite useful but come bottom of the list except for the ones with a high total frequency".

4.4. Evolving local schemes

In this experiment, local weighting schemes are evolved dependent on the global weighting scheme (7) evolved in the previous experiment. It is important that the local and global parts of the schemes interact correctly. Thus, the terminal set for this experiment is rtf, l, max_freq, tl and 1. The depth of solution is again set to 6. The following solution is one of the best and simplest weighting schemes evolved on the CISI collection:

$$lw_t = \sqrt{\frac{1 + \log(rtf)}{\sqrt{tl}}} \tag{10}$$

The Okapi-*tf* weighting which was developed for large collections employs the idea that a single occurrence of a term in a document is more important than successive occurrences of that term. The local weighting evolved in this experiment has also found this to be advantageous. This effect is achieved using the square-root and log functions on the raw term frequency. The normalization factor used is a function of the total length of a document. When the local weight is combined with the global weighting the results are favourable only on the collections of similar size to that of the training set. The average precision for the full weighting compared to the BM25 scheme can be seen in Table 6. We see that on the Medline collection the difference between the evolved solutions and the BM25 scheme is 5.1%. When tested on the CISI collection, the full evolved solution sees an increase of about 2.2% in average precision over the BM25 scheme. When the schemes are tested on the Cranfield collection, we see that the evolved solution performs only slightly better that the BM25 solution. The full evolved solution performs poorly on the two largest collections. This full evolved solution performs worse than the gw_t evolved global weighting on its own. This would seem to suggest that the local weightings evolved on small collections cannot be succesfully applied directly to larger collections. The traditional *tf-idf* is the poorest

Table 6 AP for *tf-idf*, *Piv*, $BM25$ and $gw_t \times lw_t$

Collection	Docs	Qrys	*tf-idf*	Piv	$BM25$	evolved
CISI	1,460	76	20.87%	22.13%	22.67%	25.41%
Medline	1,033	30	48.96%	52.65%	53.47%	58.48%
Cranfield	1,400	225	37.75%	41.77%	42.08%	43.13%
NPL	11,429	93	20.59%	28.07%	28.02%	28.07%
OHSU88	70,825	61	19.19%	31.08%	32.49%	26.57%
OHSU89	74,869	63	17.80%	30.07%	30.51%	25.24%
OHSU90-91	148,162	63	16.73%	26.28%	27.67%	25.38%

performing solution. These results are consistent with research to date. In particular, the *idf* global measure is consistently used on small and large collections alike with little or no modification in form. *idf* continues to be used in modern weighting functions. We suggest that global measures which use information from the entire document collection can be used sucessfully on small and large collections alike. We also suggest that even in small collections as used in this research, the document and collection frequencies are suitably representative of any piece of text.

Much research on various weighting schemes has focused on within-document term frequencies and document length normalisation factors. However, within-document characteristics have different relevance properties for small sized collections than for larger TREC data. This infact is reinforced with results on these collections using the traditional *tf* and Okapi-*tf*.

5. Conclusion

We have shown that Genetic Programming is a viable alternative approach for developing term weighting schemes in Information Retrieval. In the first experiment we have shown that weighting schemes exist that outperform the BM25 weighting scheme on small collections. Futhermore, in the second experiment we have shown that GP can be used to test the benefit of measures of a document and collection. This experiment has also shown that the cf measure can be valuable in determining the relevance of certain documents. It is also worth noting that in previous research by Oren and Fan et al. this cf measure is not used in their terminal sets. The third experiment develops a new global scheme of term importance that has characteristics similar to that which Greiff (1998) predicted would lead to schemes that may achieve a higher precision. More specifically, the cf/df measure has been found independently by evolutionary techniques to be advantageous in terms of average precision on these document collections. The cf/df measure further strengthens Luhn's hypothesis that middle frequency terms contain a higher resolving power. The measure is shown to increase average precision on these test collections over the standard *idf* solution. The fact that all of the document sets see an increase in average precision over *idf* also suggests that the evolved global scheme works for general natural language document collections. It is also significant that the global schemes evolved on very small collections also aid retrieval on larger collections. The fourth experiment adds to the global weighting scheme developed by evolving a local weighting scheme dependent on the global scheme. We see that the complete weighting scheme can outperform the BM25 weighting scheme on collections similar in size to the training set. However, the full weightings evolved on small collections do not outperform BM25 on large collections.

Future work includes evolving both global and local weighting schemes on larger document collections. We also aim to tune Okapi-*tf* to interact correctly with the global scheme identified here. The techniques used here which evolve weighting schemes in their respective domains are important in the analysis and future development of such schemes.

References

Bergstrom A, Jaksetic P and Nordin P (2000) Enhancing information retrieval by automatic acquisition of textual relations using genetic programming. In: Proceedings of the 5th international conference on Intelligent user interfaces. pp. 29–32, ACM Press

Darwin C (1859) The Origin of the Species by means of Natural Selection, or The Preservation of Favoured Races in the Struggle for Life. First edition

Fan W, Fox EA, Pathak P and Wu H (2004a) The effects of fitness functions on genetic programming-based ranking discovery for web search. Journal of the American Society for Information Science and Technology 55(7):628–636

Fan W, Gordon MD and Pathak P (2004b) A generic ranking function discovery framework by genetic programming for information retrieval. Information Processing & Management

Goldberg DE (1989) Genetic Algorithms in Search, Optimisation and Machine learning. Addison-Wesley

Gordon, M (1988) Probabilistic and genetic algorithms in document retrieval. Commun. ACM 31(10):1208–1218

Greiff W (1998) A theory of term weighting based on exploratory data analysis. In: Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98). Melbourne, Australia

Gustafson S (2004) An Analysis of Diversity in Genetic Programming. Ph.D. thesis, School of Computer Science and Information Technology, University of Nottingham, Nottingham, England

Hersh W, Buckley C, Leone TJ and Hickam D (1994) OHSUMED: an interactive retrieval evaluation and new large test collection for research. In: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 192–201, Springer-Verlag New York, Inc

Horng J and Yeh C (2000) Applying genetic algorithms to query optimization in document retrieval. Information Processing & Management 36(5):737–759

Kim S and Zhang B-T (2001) Evolutionary Learning of Web-Document Structure for Information Retrieval. In: Proceedings of the 2001 Congress on Evolutionary Computation CEC2001. pp. 1253–1260, IEEE Press

Koza JR (1992) Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, MA, USA

Kuscu I (2000) Generalisation and domain specific functions in Genetic Programming. In: Proceedings of the 2000 Congress on Evolutionary Computation CEC00. pp. 1393–1400, IEEE Press

Kwok KL (1996) A new method of weighting query terms for ad-hoc retrieval. In: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 187–195, ACM Press

Lewis D (1992) Feature Selection and Feature Extraction for Text Categorization. Proceedings of Speech and Natural Language Workshop pp. 212–217

Li L, Shang Y and Zhang W (2002) Improvement of HITS-based algorithms on web documents. In: Proceedings of the eleventh international conference on World Wide Web. pp. 527–535, ACM Press

Lucas JM, van Baronaigien DR and Ruskey F (1993) On rotations and the generation of binary trees. J. Algorithms 15(3):343–366

Luhn H (1958) The automatic creation of literature abstracts. IBM Journal of Research and Development pp. 159–165

Oren N (2002) Re-examining tf.idf based information retrieval with Genetic Programming. Proceedings of SAICSIT

Pirkola A and Jarvelin K (2001) Employing the resolution power of search keys. J. Am. Soc. Inf. Sci. Technol. 52(7):575–583

Porter M (1980) An algorithm for suffix stripping. Program 14(3):130–137

Robertson SE, Walker S, Hancock-Beaulieu M, Gull A and Lau M (1998) Okapi at TREC-7: Automatic ad hoc, filtering, VLC and interactive track. In: The Seventh Text REtrieval Conference (TREC-7) NIST

Salton G and C Buckley (1988) Term-weighting approaches in automatic text retrieval. Information Processing & Management 24(5):513–523

Salton G, Wong A and Yang CS (1975) A vector space model for automatic indexing. Commun. ACM 18(11):613–620

Salton G and Yang CS (1973) On the specification of term values in automatic indexing. Journal of Documentation 29, 351–372

Schultz C (1968) H.P. Luhn: Pioneer of Information Science - Selected Works. Macmillan, London

Singhal A (2001) Modern information retrieval: A brief overview. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 24(4):35–43

Singhal A, Buckley C and Mitra M (1996) Pivoted document length normalization. In: SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 21–29, ACM Press

Sparck Jones K (1972) A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation 28, 11–21

Van Rijsbergen, CJ (1979) Information Retrieval, 2nd edition. Dept. of Computer Science, University of Glasgow

Vrajitoru D (1998) Crossover improvement for the genetic algorithm in information retrieval. Inf. Process. Manage. 34(4):405–415

Vrajitoru D (2000) In F. Crestani, G. Pasi (eds.): Soft Computing in Information Retrieval. Techniques and Applications, pp. 199–222. Physica-Verlag

Yang J-J and Korfhage R (1993) Query Optimization in Information Retrieval Using Genetic Algorithms. In: Proceedings of the 5th International Conference on Genetic Algorithms. pp. 603–613, Morgan Kaufmann Publishers Inc

Yu, CT and Salton G (1976) Precision weighting - An effective automatic indexing method. Journal of the ACM 23(1):76–88

Zipf G (1949) Human Behaviour and the Principle of Least Effort. Addison-Wesley, Cambridge, Massachusetts