Department of Information Technology

technical report NUIG-IT-201205

Evolving Term-Selection Schemes for Pseudo-Relevance Feedback in Information Retrieval

R. Cummins (NUI, Galway)
C. O'Riordan (NUI, Galway)

Evolving Term-Selection Schemes for Pseudo-Relevance Feedback in Information Retrieval

Ronan Cummins and Colm O'Riordan
Dept. of Information Technology,
National University of Ireland,
Galway, Ireland.
ronan.cummins@nuigalway.ie
colmor@it.nuigalway.ie

January 24, 2006

Abstract

Automatic query expansion in Information Retrieval aims to improve retrieval performance by overcoming the problem of term mismatch between a query and its relevant documents. Pseudo-relevance (blind) feedback techniques have been shown to be of benefit on large TREC collections in recent years. This technique analyses terms in the top few documents deemed relevant by the system, reformulates the query and runs the newly formulated query through the system again.

This paper describes a method which uses Genetic Programming to evolve a scheme for selecting and weighting terms from the top ranked documents in order to expand the initial query and increase the mean average precision achieved by the system. The scheme is also used to weight the terms in the reformulated query. As a result, the genetic program has to, not only learn a scheme for identifying the best terms for expansion, but also learn a scheme which correctly weights these in relation to each other. The resulting schemes are tested on standard test collections and are shown to increase mean average precision over existing benchmark term selection schemes.

1 Introduction

Information Retrieval (IR) is concerned with the automatic retrieval of relevant documents given a user need (query). However, vocabulary differences between the user and supplier of information have often led to a difficulty in retrieving many relevant documents. Query expansion techniques have long been proposed as a means of overcoming term mismatch between the user's vocabulary and the vocabulary of the documents in the collection. Query expansion techniques typically add a number of terms to the original query based on some heuristics in order to improve the performance of the original query. Typically, there are two types of query expansion methods; global (automatic thesaurus construction) and local (pseudo-relevance or blind feedback) query expansion techniques [1, 2]. This paper is concerned with the latter. In pseudo-relevance feedback, the top P documents from an initial retrieval run for the query are deemed relevant. Then, terms from this set of P documents are added to the original query. The evolution of schemes to correctly select terms and their associated weight is the focus of this paper.

Recently there have been more and more attempts applying machine learning techniques to the domain of IR. Genetic Programming (GP) has been adopted by some researchers as it has advantages over other machine learning techniques. In particular, GP outputs a symbolic representation of a solution which can be further analysed. It is also well suited to problems where generalisation is needed. Developed in the early 1990s, the GP area [3] has grown and helped to solve problems in a variety of areas. GP is inspired by the Darwinian theory

of natural selection, where individuals that have a higher fitness value will survive and produce offspring. GP can be viewed as an artificial means of selective breeding.

This paper presents a Genetic Programming framework that artificially breeds query expansion selection schemes for use in a standard adhoc retrieval framework. The next section introduces some background material in query expansion. The GP process is outlined in section three. Some past approaches of evolutionary computation techniques applied to IR are also reviewed in this section. Section four describes the system and experimental design. Results and analysis are discussed in detail in section five. Finally, our conclusions are discussed in section six.

2 Background

This section presents background material for existing term-weighting, query expansion techniques and terminology used for the remainder of this paper.

2.1 Local Query Expansion Approaches

As previously mentioned, pseudo-feedback techniques use the top few documents from an initial retrieval run from which to extract terms and add to the original query. The terms to add to the query are chosen based on their frequency characteristics. As the top P documents are deemed relevant, the Robertson/Sparck-Jones weight [4] developed for the probabilistic model of IR is often used to determine the weight for a term. This weight (w_{rsj}) is calculated as follows:

$$log(\frac{(pdf_t + 0.5)/(P - pdf_t + 0.5)}{(df_t - pdf_t + 0.5)/(N - df_t - P + pdf_t + 0.5)}) \quad (1)$$

where P is the number of pseudo-relevant documents, N is the number of documents in the collection, df_t is the document frequency of term t and pdf_t is number of pseudo-relevant documents that term t occurs in. A simple but effective term-selection scheme used in some TREC runs [4] is as follows:

$$TSV_t = pdf_t \times w_{rsj} \quad (2)$$

where pdf_t is the number of pseudo-relevant documents (P) term t occurs in. A number of terms (E) is then chosen based on the TSV_t and these are added to the query. The weight applied to these expanded terms is the w_{rsj} from (1). The number of terms (E) and number of top ranked documents (P) deemed relevant are usually fixed although there has been attempts applying thresholds to the TSV_t so that only good quality expansion terms are added to the original query [4].

2.2 Initial Ranking

The *tf-idf* family of weighting schemes [5] are the most widely used weighting schemes for the vector space model. The BM25 weighting scheme, developed by Robertson et al. [6], is a weighting scheme based on the probabilistic model. Okapi-*tf* is calculated as follows:

$$Okapi\text{-}tf = \frac{rtf}{rtf + k_1((1-b) + b\frac{dl}{dl_{avg}})} \quad (3)$$

where rtf is the raw term frequency and dl and dl_{avg} are the length and average length of the documents respectively. k_1 and b are tuning parameters set to 1.2 and 0.75 respectively. The idf_t of a term as determined in the BM25 formula is simply the w_{rsj} (1) when no relevance information is available. Thus, the weight assigned to a query term t in the document d is as follows:

$$w(t,d) = Okapi\text{-}tf \times log(\frac{N - df_t + 0.5}{df_t + 0.5}) \quad (4)$$

The score for a document d can then be calculated as follows:

$$BM25(Q,d) = \sum_{t \in Q \cap d_i} (w(t,d) \times qtf_t) \quad (5)$$

where qtf_t is the term-frequency of t in the query Q. Thus, $BM25(Q,d)$ is the score of document d in relation to the query Q.

3 Genetic Programming

This section introduces and summarises the Genetic Programming process.

3.1 Basic Algorithm

Genetic Programming [3] is a heuristic stochastic searching algorithm, inspired by natural selection, and is efficient for navigating large complex search spaces. In the GP process, a population of solutions is created randomly. These solutions are encoded as trees and can be thought of as the genotypes of the individuals. Each tree (genotype) contains nodes which are either functions (operators) or terminals (operands). The values on the nodes of each tree are referred to as alleles. 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. Individuals are selected for reproduction based on their fitness value. Fitter solutions will be selected more often.

Once selection has occurred, 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 allele of a gene to create a new individual or the change of a subtree of an individual. Selection and recombination occurs until the population is replaced by newly created individuals. 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 following a predefined number of generations, or until convergence of the population is achieved or after an individual is found with an acceptable fitness.

Tournament selection is one of the most common selection methods used. In tournament selection, a number of solutions are chosen at random from the population 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 called the tournament size and this can be increased or decreased to increase or decrease the speed of convergence.

Crossover is the main reproductive mechanism in GP. When two solutions are selected from the selection process, their genomes are combined to create a new individual.

3.2 Existing Evolutionary Approaches

There have been several attempts using Genetic Programming to evolve term-weighting schemes in an adhoc or vector based framework [7, 8, 9, 10]. Some of these attempts have been successful in increasing the performance of the initial ranking function over the benchmark used. As the performance of pseudo-relevance feedback techniques crucially relies on the number of actual relevant documents that lie in the top ranks of the initial search, Fan et al. [11] has also used a Genetic Programming technique to increase the initial ranking and in particular, has shown it to aid the performance of pseudo-relevance feedback. Work which evolves Boolean type queries using expansion terms has also been conducted [12, 13].

Genetic Algorithms (GA) have also been used in many areas of Information Retrieval. Many approaches attempt to evolve a query representation so that the optimal document set is returned. Horng and Yeh [14] use an GA approach to extract keywords from a subset of relevant documents to construct a query and then adapt the weights to best suit the relevant documents. Query representations have also been evolved by GA using relevance feedback techniques [15].

4 Design and Experimental Setup

4.1 Term-Selection

The GP approach adopted evolves the scheme used to select and weight terms for use in the expanded query in order to improve the retrieval of the system. For each query expansion scheme, each term in the top P documents from the initial retrieval run is rated on how useful it is. Firstly, it is necessary to choose a value for P and decide how many terms to add to the original query (E). Previous research has indicated that values for P should be between 8 and 16 and values for E should lie between of 7 and 42. We use $P = 10$ and $E = 16$ as these lie within the best parameter ranges [16]. As we use the term selection scheme evolved to weight the expansion terms, the top 16 should be the most important and terms lower down the rank should

not modify the query as much. We also aim to show that this is the case and the solution learned is adequate for any value of E. We choose E to be 16 so that solutions can be evolved in a reasonable time as longer queries take longer to process.

4.2 Term Re-Weighting

The problem of what weight to give the expanded queries is solved by assuming that the weight of an expanded term is a function of the usefulness of the expansion term. It is also logical to assume that the weight of the expansion term is also related to the weighting scheme applied to the original query terms (i.e. a *tf-idf* type scheme). Thus, the following formula is used to score the complete expanded query (EQ) in relation to a document d:

$$sim(EQ, d) = BM25(Q, d) + \sum_{t \in E} ESV_t \times w(t, d) \quad (6)$$

where EQ is the expanded query, Q is the original query, E is the set of expansion terms, ESV_t is our evolved selection value and $w(t, d)$ is as (4). Thus, a weighting of 1 for ESV_t would indicate that the expansion term is as important as if it had occurred in the original query. In this way the GP can also learn the correct weighting for the expansion terms which is a function of the usefulness of the expansion term. It is worth noting that this is slightly different from the way in which terms are re-weighted using the Robertson/Sparck-Jones weight (1).

4.3 Document collections and pre-processing

The document collections used in this research are the Medline, NPL and OHSUMED collections[1]. We divide the OHSUMED collections into three collections. Two collections which are of similar size and contain documents from the years 1988 and 1989 respectively are constructed. The third OHSUMED collection contains documents from both 1990 and 1991. To test the evolved selection schemes on various query lengths we use half of the LATIMES and FBIS collections from TREC disks 4 and 5. We use a different set of 50 TREC topics for each of these collections. For each set of topics we create a short query set (s) which consists of the title fields of the topics, a medium length query set (m) which consists of the title and description fields, and a long query set (l) which consists of the title, description and narrative fields.

The documents and queries are pre-processed by removing standard stop-words from the Brown Corpus[2] and are stemmed using Porter's stemming algorithm [17]. All queries with no relevant documents are ignored by the system. Table 1 shows the characteristics of the document collections used once preprocessing is completed:

Table 1: Characteristics of document collections

Collection	# docs	words/doc	# Topics	words/topic
Medline	1,033	56.8	30	11
NPL	11,429	18.8	93	6.78
OHSU88	70,825	75.3	63	4.97
OHSU89	74,869	76.9	63	4.97
OHSU90-91	148,162	81.4	63	4.97
LATIMES (s)	65,138	250.8	301-350	2.42
LATIMES (m)				9.92
LATIMES (l)				29.86
FBIS (s)	61,578	257.2	351-400	2.42
FBIS (m)				7.88
FBIS (l)				21.98

4.4 Terminal and Function set

To determine the terminal and function set, it is neccessary to consider the characteristics of the terms in the set of pseudo-relevant documents and the characteristics of these terms in the entire collection. It is important to try to keep the terminals as primitive (atomic) as possible so that there are fewer assumptions as to how the relevance of terms, documents and pseudo-relevant documents are related. The GP should be able to discover the best way to combine these to improve the performance for the given training data. Table 2 and Table 3 show the terminal and function set used in the experiments respectively.

4.5 Fitness Function

The mean average precision (MAP), used as the fitness function, is calculated for each scheme by comparing the ranked list returned by the system

[1] http://www.dcs.gla.ac.uk/idom/ir_resources/

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

Table 2: Terminal Set

Terminal	Description
N	no. of documents in the collection
P	no. of documents in pseudo-relevance set (10)
cf_t	frequency of term t in the collection
df_t	document frequency of term t in the collection
pcf_t	frequency of term t in pseudo-relevant documents
pdf_t	number of pseudo-relevant documents containing t
V	vocabulary of collection (no. of unique terms)
C	size of collection (no. of words)
U	vocabulary of pseudo-relevant document set
S	size of pseudo-relevant document set in words

Table 3: Function Set

Function	Description
+, ×, /, -	standard arithmetic functions
log	the natural log
sqrt	square-root function
sq	square

for each query expansion scheme against the human determined relevant documents for each query. MAP is calculated over all points of recall and is frequently used as a performance measure in IR systems as it provides a measure of both the accuracy and recall of the retrieval system.

4.6 GP Parameters

All experiments are run for 50 generations with an initial random population of 1000. The solutions are trained on the entire Medline collection and query set. They are then tested for generality on the collections that were not included in training. Trees are limited to a depth of 8 to promote generality as shorter solutions are usually more general [18]. We use an elitist strategy where the best performing individual is copied into the next generation. The tournament size is set to 4. The aim is to discover general natural language characteristics for query expansion that will aid retrieval performance. As we wish to select terms based on some normalised level (for example from 0 to 1, indicating how much of the original *tf-idf* weight we should assign to the expanded term) we do not add terms that occur in the original query to the expanded query during training. We do not want the expansion terms to overwhelm the original query. We re-evaluate the evolved schemes after training allowing the orignal terms to be selected and re-weighted like the other standard benchmark schemes.

4.7 Benchmark Selection schemes

The benchmark selection scheme used is as defined in (2). For this benchmark which we will call TSV, terms are selected based on formula (2) which basically chooses the most frequently occuring discriminating terms in the top P documents. The expanded terms are then weighted using the w_{rsj} weight (1) and *okapi-tf* instead of the traditional *idf* and *okapi-tf* as relevance information is assumed to be available. We also test another re-weighting scheme which is simply $w_{rsj}/3$ so that the expanded query is not dominated by expanded terms. This is similar to a re-weighting scheme used previously [19] and found to be effective. The terms are selected as TSV but are simply weighted as one third of the w_{rsj} weight. This benchmark will be called $TSV\frac{1}{3}$. We have also conducted prelimary tests to determine whether original query terms should be added to the expanded query and in effect re-weighted. We have found that it is beneficial to allow the original query terms to be selected for re-weighting under the same conditions as non-query terms. It is worth noting that on the Medline collection the expanded queries improve significantly over the original query because of its initially high MAP. As a result many relevant documents will be contained in the top P documents from the inital ranking.

Table 4: % MAP for Benchmarks

		Original	Expanded	
Collection	Qrys	$BM25$	$\boldsymbol{TSV}$	$\boldsymbol{TSV\frac{1}{3}}$
Medline	30	53.43	62.69	60.78
NPL	96	28.75	27.11	28.62
OSHU88	63	32.78	34.29	36.61
OSHU89	63	30.69	30.15	31.30
OSHU90-91	63	28.08	31.41	31.69

The best benchmark seems to be somewhat dependent on what collection is used but we can see that on all the larger collections (over a few thousand documents) the $TSV\frac{1}{3}$ is the best performing scheme on the collections tested. The reduction of the weight of expanded terms seems to reinforce claims made in previous work [19] as it increases

MAP. This is the benchmark that will be used for the remainder of this paper and is shown in Table 4.

4.8 Efficiency of Approach

The approach adopted is quite feasible even for larger document collections. All measures (terminals) in Table 2 are gathered for each query by an initial retrieval run using the BM25 matching function. Then, the time taken to evaluate a query of length L in a collection of N documents is $O(N \times L)$. For the entire GP process this grows to $O(N \times (L+E) \times G \times M)$, where G in the number of generations, M is the size of the population and E is the number of terms added to the original query. This is only slightly more computationally intensive than previous approaches that evolve term-weighting schemes in IR [20]. It involves one extra inital retrieval run and longer queries. We use the smallest Medline collection and quite a large population of 1000 for 50 generations.

5 Results and Analysis

5.1 An Evolved Selection scheme

The following solution is the best evolved selection scheme (ESV) on the Medline collection after 4 runs of the GP:

$$\sqrt{\frac{(\frac{pcf}{V} \times log(pdf) \times pcf^2) + (\frac{P}{\sqrt{df}} \times log(pdf) \times log(pcf))}{log(\frac{P}{\sqrt{df}} \times log(log(pcf)) \times V)}} \quad (7)$$

Figure 1 shows the evolved selection values of two terms of different document frequencies (10 and 1000) for varying values of pcf and pdf. We can see that if a term of document frequency 10 occurs in all pseudo-relevant documents it will get a weighting of close to 2 (or double its $BM25$ value).

From Figure 1 we can see that the selection value is directly proportional to the frequency of the term in the pseudo-relevant document set (pcf) and the number of pseudo-relevant documents the term occurs in (pdf). Also, the scheme will also tend to promote terms that have a lower document frequency. We can also see that the ESV weight tends to zero as the pdf tends to 1. Characteristics of this evolved weight will be discussed later.

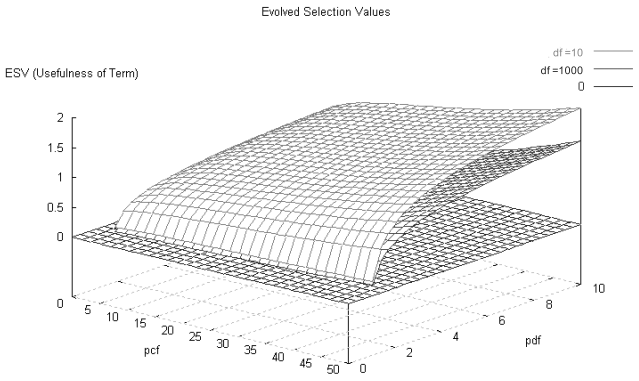

Figure 1: ESV for two different terms on OHSU88

Table 5 shows the mean average precision (MAP) of the expanded queries using the best benchmark and best evolved selection scheme (ESV). It is encouraging that the MAP increases on many unseen document collections. Although the training set used is quite small we can see that the term-selection properties for collections of various sizes are quite similar to each other as the best scheme found on the training set increases the MAP on larger collections.

Table 5: MAP for expanded queries using $TSV\frac{1}{3}$ and ESV_t

Collection	Qrys	$BM25$	$TSV\frac{1}{3}$	ESV_t
Medline	30	53.43	60.78	64.20
NPL	93	28.75	28.62	28.77
OHSU88	63	32.78	36.61	37.00
OHSU89	63	30.69	31.30	33.98
OHSU90-91	63	28.08	31.69	32.71
LATIMES (s)	50	28.15	29.57	31.94
LATIMES (m)	50	30.85	31.26	34.18
LATIMES (l)	50	31.41	31.72	34.00
FBIS (s)	50	18.98	18.89	19.82
FBIS (m)	50	22.09	22.67	25.10
FBIS (l)	50	21.83	23.04	25.73

Table 6 shows the top 16 terms added to the medium topic 301 for the LATIMES collection. Topic 301 is about "International Organized Crime" and the description is "Identify organizations that participate in international criminal activity, the activity, and, if possible, collaborating organizations and the countries involved". This is

stemmed to the following: **intern organ crime identifi organ particip intern crimin activ activ collabor organ countri involv**. It is interesting to see that the GP evolves a scheme (ESV) which weights terms on a suitable scale for expansion. The first stem selected (**'anti-terror'**) is a non-query term and gets added to the query with about 0.44 of its BM25 weight. The second stem selected (**'organ'**) appears in the original query and is also added with a weight of about 0.44 indicating that its weight in the re-formulated query will be 1.44 times its BM25 weight. It is interesting to note that although most terms are common to the top 16 terms for both selection schemes, the order in which they appear is different. However there are certain terms that are unique to the top 16 terms for a selection scheme. It is also worth noting that for the best benchmark scheme, terms are not weighted in the same way there are selected for expansion. Terms are selected by their TSV value (2) and weighted using the w_{rsj} weight (1). When many more terms are added to a query, its performance should not change significantly as the weight given to the expansion terms should be weighted correctly to reflect the usefulness of the term. This property is investigated in the next section.

Table 6: Scores for Expansion terms for Topic 301 (m) on LATIMES

Topic 301					Topic 301				
Terms	ESV	df	pcf	pdf	Terms	TSV	df	pcf	pdf
anti-terror	0.44	12	3	2	activ	55.5	4916	29	10
organ	0.44	7035	47	10	organ	51.5	7035	47	10
hoodlum	0.38	23	3	2	crimin	22.4	2170	23	6
racket	0.38	341	18	3	repres	20.6	6615	10	7
fbi	0.37	812	25	4	includ	17.4	1811	19	8
rico	0.36	231	35	2	white-collar	17.4	94	3	3
crime	0.35	2681	36	5	justic	17.1	2053	15	5
activ	0.35	4916	29	10	cooper	16.9	2134	7	5
white-collar	0.34	94	3	3	terror	16.7	673	5	4
crimin	0.33	2170	23	6	civil	16.2	2427	16	5
mobster	0.33	67	5	2	act	16.1	5840	14	6
indict	0.30	779	15	3	fbi	16.0	812	25	4
law	0.30	6995	32	6	crime	15.7	2681	36	5
mafia	0.30	110	5	2	anti-terror	15.0	12	3	2
justic	0.30	2053	15	5	law	14.9	6995	32	6
mob	0.30	244	11	2	identifi	14.7	3259	5	5

5.2 Expanding Queries by More Terms

To test whether the evolved expansion scheme (ESV) correctly weights expansion terms, we tested the scheme by adding various number of terms to the original query. The evolved scheme was originally evolved by adding the top 16 terms to each query on the Medline collection. Figure 2 shows both the evolved selection scheme and the best benchmark scheme ($TSV\frac{1}{3}$) for varing numbers of expansion terms on both of the OHSU88 and OHSU89 collections. We evaluated queries with up to 48 expanded terms in multiples of 8 terms.

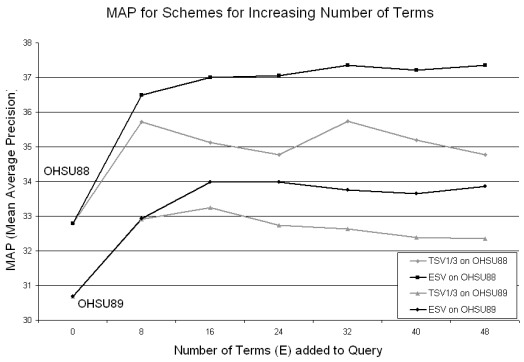

Figure 2: MAP for varying number of Terms (E)

We can see from Figure 2 that the evolved selection scheme is quite stable as more terms are added to the query. On both collections tested the MAP of the evolved scheme is more stable as more terms are added to the original query. The benchmark scheme is quite erratic for various numbers of terms and seems to find good expansion terms for various values of E. This would seem to indicate that the weighting values are not correct for the benchmark expansion scheme (i.e. bad expansion terms are getting an incorrectly high weighting or good expansion terms are not getting a sufficiently high selection value). We can see for the evolved selection value ((ESV) (7)), a term occurring in only one pseudo-relevant document will get a zero weighting because of the $log(pdf)$ part of the numerator and in effect is not added to the query. Thus, the number of terms available for selection is limited to terms that occur in at least two pseudo-relevant

documents. It is also interesting to note that the ESV scheme will only select terms that occur more than three times through-out the pseudo-relevant document set. The $log(log(pcf))$ part of the denominator in equation (7) is only positive when the pcf value is three or above and is negative for all other values. This leads the entire denominator, and as a result the scheme, being undefined when pcf is below three. This leads to a very selective type of expansion.

For example, the short 53^{rd} OHSUMED query evaluated on the OHSU89 collection has only ever 3 terms added to it using the ESV scheme as all other potential expansion terms do not meet the minimum requirement for expansion. However, the benchmark selection scheme does not have such thresholds and all terms within the pseudo-relevant document set can be added and will be given some weighting. This 53^{rd} OHSUMED query is an extreme case and usually more terms are available for selection. For example, when adding 48 terms to each of the 63 queries on the OHSU89 collection, 22 queries actually add the full 48 terms. The remaining 41 queries have already added all terms that would receive a positive weighting and thus have exhausted their supply of possible expansion terms before the 48^{th} term. This is why the MAP only changes slightly as more terms are added to each query for the ESV scheme in Figure 2.

It is also important to point out that some of the document collections in this research are somewhat shorter compared to other TREC documents and this leads to a smaller pool of potential expansion terms being available during the term-selection phase of the process. These characteristic have been learned when the number of pseudo-relevant documents is set to 10 and may not be generalisable for all values of P. However, we can see that by selecting 16 terms and also using the selection value to weight the term in the expanded query a very general stable selection scheme has been learned. Furthermore, the GP has evolved a type of automatic thresholding into the weighting scheme that is different for each query and is dependent on the quality of expansion terms available to it.

6 Conclusion and Future Work

We have shown that GP is a viable means of finding term selection schemes in Information Retrieval. We have also shown that the automatic process can evolve general selection schemes that increase mean average precision over other standard benchmarks. By selecting a set number of terms and using the selection value to weight the term in the expanded query a very general stable reweighting scheme can been learned. The scheme identified has also evolved a type of automatic thresholding into the weighting scheme that is different for each query and is dependent on the quality of expansion terms available to it. Future work includes further analysis of the evolved term-selection schemes to explain certain poor performing queries.

References

[1] Qiu, Y., Frei, H.P.: Concept-based query expansion. In: Proceedings of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval, Pittsburgh, US (1993) 160–169

[2] Xu, J., Croft, W.B.: Query expansion using local and global document analysis. In: SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (1996) 4–11

[3] Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, MA, USA (1992)

[4] Robertson, S.E., Walker, S.: Okapi/keenbow at trec-8. In: TREC. (1999)

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

[6] Robertson, S.E., Walker, S., Hancock-Beaulieu, M., Gull, A., Lau, M.: Okapi at TREC-3. In: In D. K. Harman, editor, The

Third Text REtrieval Conference (TREC-3) NIST. (1995)

[7] Oren, N.: Re-examining tf.idf based information retrieval with genetic programming. Proceedings of SAICSIT (2002)

[8] Fan, W., Gordon, M.D., Pathak, P., Xi, W., Fox, E.A.: Ranking function optimization for effective web search by genetic programming: An empirical study. In: Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 4, IEEE Computer Society (2004) 40105

[9] Trotman, A.: Learning to rank. Information Retrieval **8** (2005) 359 – 381

[10] Cummins, R., O'Riordan, C.: Evolving general term-weighting schemes for information retrieval: Tests on larger collections. Artificial Intelligence Review **24** (2005) 277–299

[11] Fan, W., Luo, M., Wang, L., Xi, W., Fox, E.A.: Tuning before feedback: combining ranking discovery and blind feedback for robust retrieval. In: SIGIR '04: Proceedings of the 27th annual international conference on Research and development in information retrieval, New York, NY, USA, ACM Press (2004) 138–145

[12] Steele, R., Powers, D.: Evolution and evaluation of document retrieval queries. In Powers, D.M.W., ed.: NeMLaP3/CoNLL98: New Methods in Language Processing and Computational Natural Language Learning, Flinders University, Adelaide, Australia, ACL Association for Computational Linguistics (1998) 163–164

[13] Smith, M.P., Smith, M.: The use of genetic programming to build boolean queries for text retrieval through relevance feedback. Journal of Information Science **23** (1997) 423–431

[14] Horng, J., Yeh, C.: Applying genetic algorithms to query optimization in document retrieval. Information Processing & Management **36** (2000) 737–759

[15] Lopez-Pujalte, C., Guerrero-Bote, V.P., de Moya-Anegon, F.: Genetic algorithms in relevance feedback: a second test and new contributions. Inf. Process. Manage. **39** (2003) 669–687

[16] Billerbeck, B., Zobel, J.: When query expansion fails. In: SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, New York, NY, USA, ACM Press (2003) 387–388

[17] Porter, M.: An algorithm for suffix stripping. Program **14** (1980) 130–137

[18] Kuscu, I.: Generalisation and domain specific functions in genetic programming. In: Proceedings of the 2000 Congress on Evolutionary Computation CEC00, IEEE Press (2000) 1393–1400

[19] Billerbeck, B., Scholer, F., Williams, H.E., Zobel, J.: Query expansion using associated queries. In: CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management, New York, NY, USA, ACM Press (2003) 2–9

[20] Fan, W., Gordon, M.D., Pathak, P.: A generic ranking function discovery framework by genetic programming for information retrieval. Information Processing & Management (2004)