Artif Intell Rev (2006) 26:35–47
DOI 10.1007/s10462-007-9034-5

Evolved term-weighting schemes in Information Retrieval: an analysis of the solution space

Ronan Cummins · Colm O'Riordan

Received: 1 October 2006 / Revised: 30 October 2006 / Published online: 12 September 2007

Abstract Evolutionary computation techniques are increasingly being applied to problems within Information Retrieval (IR). Genetic programming (GP) has previously been used with some success to evolve term-weighting schemes in IR. However, one fundamental problem with the solutions generated by this stochastic, non-deterministic process, is that they are often difficult to analyse. In this paper, we introduce two different distance measures between the phenotypes (ranked lists) of the solutions (term-weighting schemes) returned by a GP process. Using these distance measures, we develop trees which show how different solutions are clustered in the solution space. We show, using this framework, that our evolved solutions lie in a different part of the solution space than two of the best benchmark term-weighting schemes available.

Keywords Genetic programming · Information Retrieval · Term-weighting schemes

1 Introduction

A term-weighting scheme is essentially the document ranking function in an Information Retrieval (IR) system. As such, they are a very crucial part of any IR system (Salton and Buckley 1988) and improving upon them is a vibrant area of research within IR. These term-weighting schemes assign values to search terms based on how useful they are likely to be in determining the relevance of a document. Documents are scored in relation to a query using one of these term-weighting schemes and are returned in a ranked list format.

Genetic Programming (GP) (Koza 1992) is a biologically-inspired search algorithm useful for searching large complex spaces. As GP is a non-deterministic algorithm it cannot be expected to produce the same solution each time. Restart theory in GP suggests that it is

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

C. O'Riordan
e-mail: colm.oriordan@nuigalway.ie

necessary to restart the GP a number of times in order to achieve good solutions (Luke 2001). As a result, an important question regarding the solutions generated by the GP process is: do all the good solutions behave similarly or is the GP bringing us to a different area in the solution space each time?

This paper presents a framework for evaluating the distance between the ranked lists produced from different term-weighting schemes in order to understand their relative closeness. These different term-weighting schemes are produced using a GP process. We introduce two different distance measures and show that they are useful in determining how term-weighting schemes are expected to perform in a general environment. We use the distance measures to show where the evolved term-weighting schemes lie in the solution space. Indicating where the solutions lie in the solution space is only useful when a corresponding increase in performance is seen. Thus, results for all of the term-weighting schemes resultant from the GP are presented. This paper is an extension of previous work (Cummins and O'Riordan 2006a). Further results and discussion are presented here for short and long queries.

Section 2 of this paper introduces some GP terminology and some relevant research in the area is discussed. Section 3 introduces the two distance measures developed. Our experimental setup is outlined in Sect. 4, while Sect. 5 presents and discusses our results. Finally, our conclusions are summarised in Sect. 6.

2 Genetic programming for term-weighting

Inspired by the theory of natural selection, the GP process usually starts with a population of randomly-created solutions (although some approaches seed the initial population with certain known solutions). These solutions, encoded as trees, undergo generations of selection, reproduction and mutation until suitable solutions are found.

Each tree (genotype) contains nodes which are either functions or terminals. The phenotype of the individual is often described as its behaviour and is essentially the solution in its environment. Selection occurs based on the fitness only. Fitness is determined by the phenotype, which is in turn determined by the genotype. As one can imagine, different genotypes can produce the same phenotype, and different phenotypes can have the same fitness. For many problems in GP in an unchanging environment, the same genotype will produce the same phenotype which will have the same fitness. Bloat is a another common phenomenon in GP; where solutions grow in size without a corresponding increase in fitness.

Figure 1 shows how terminology within the GP paradigm is mapped to that within IR. Mean average precision (MAP) is used as the fitness function as it is a commonly used metric to evaluate the performance of IR systems and is known to be a stable measure (Buckley and Voorhees 2000). Furthermore, it has been used with success in previous research evolving term-weighting schemes in IR (Fan et al. 2004; Trotman 2005; Cummins and O'Riordan 2006b).

2.1 Previous research

The search for term-weighting schemes that outperform traditional methods has been a vibrant research area since the inception of IR. Many successful attempts have been made to formulate schemes using theoretical methods such as the vector space model (Salton et al. 1975), the probabilistic model (Jones et al. 2000) and more recently various language models (Ponte and Croft 1998). Other attempts have focused on combining existing parts of term-weighting

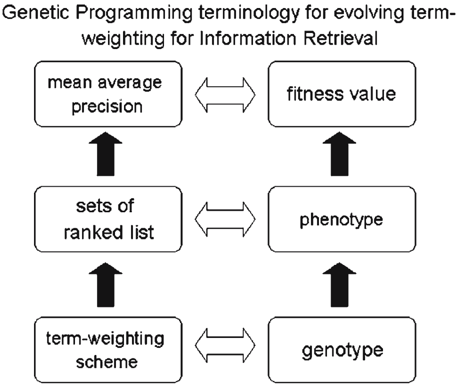

Fig. 1 GP for Information Retrieval

schemes in an unguided sense (Zobel and Moffat 1998). While more recently a number of attempts have focused on determining a set of constraints for which all *good* term weighting schemes should satisfy (Fang and Zhai 2005). This approach has the benefit of reducing the search space of weighting schemes to those that are known to have beneficial properties.

GP techniques have previously been adopted to evolve weighting functions and are shown to outperform standard weighting schemes in an ad hoc framework (Fan et al. 2004; Oren 2002; Trotman 2005; Cummins and O'Riordan 2006b). However, in many of these approaches a critical analysis of the solutions evolved is not presented. In previous work (Cummins and O'Riordan 2006b), some analysis of the genotypes of term-weighting schemes and the terminal set was conducted mainly for small collections. It was also concluded that local (within-document) schemes should be evolved on larger TREC style collection. More importantly, an analysis of the phenotypic space was not conducted. It is important to gain an understanding of where in the solution space the best solutions lie as different term-weighting schemes can lead to a better understanding of term-weighting in general. These can also lead to improved retrieval in systems that use fusion techniques which use different ranked lists from different IR systems.

Distance measure between ranked lists are currently used in IR. Spearman's rank correlation and Kendall tau correlation are two common correlations that measure the difference between ranked sets of data. However, both Spearman's rank correlation and Kendall's tau use all of the ranked data in a pair of ranked lists.

2.2 Parts of a term-weighting function

We separate the weighting scheme into three different parts and search each problem space in turn. As a result, we evolve weighting schemes, which score a document d in relation to a query Q, in the following structure:

$$score(d, Q) = \sum_{t \in Q \cap d} (gw_t \times ntf \times qtf) \tag{1}$$

where gw_t is a global weighting, ntf is a normalised term-frequency and qtf is the frequency of the term in the query (which remains constant throughout *all* weighting schemes in this paper). The global part of the scheme weights terms on their ability to discriminate between documents. The normalised term-frequency consists of a term-frequency influence factor which promotes documents with more occurrences of a particular term. The aim of the normalisation part of the term-frequency is to avoid over-weighting longer documents simply because they have more matching terms. It is worth noting that this function framework may restrict the weighting scheme to certain forms. However, it should be noted that most studies into term weighting assume, or can be reduced to, a similar structure. Indeed, both benchmarks used in this research fit this model.

3 Phenotype distance measures

In our framework we measure the phenotype of our solutions by examining the sets of ranked lists returned by a term-weighting solution for a set of topics on a document collection (its environment). Two measures which were previously developed (Cummins and O'Riordan 2006c) and used for exploring the global space of term-weighting schemes are included here for completeness.

The distance measures developed only measure the parts of the ranked lists which affect the MAP (fitness) of a solution. This is important as the rank of relevant documents is the only direct contributing factor to the fitness of individuals within the GP process. The first metric measures the average difference between the ranks of relevant documents in two sets of ranked lists. This measure will tell us if the same relevant documents are being retrieved at, or close to, the same ranks and will tell us if the weighting schemes are evolving toward solutions that produce similar phenotypes. Thus, the distance measure $dist(a, b)$, where a and b are two weighting schemes, is defined follows:

$$dist(a,b) = \frac{1}{|R|}\sum_{i \in R}\begin{cases} |lim - r_i(b)| & \text{if } r_i(a) > lim \\ |r_i(a) - lim| & \text{if } r_i(b) > lim \\ |r_i(a) - r_i(b)| & \text{otherwise} \end{cases}$$

where R is the set of relevant documents for all queries used and $r_i(a)$ is the rank position of relevant document i under weighting scheme a. The maximum rank position available from a list is denoted by lim and is usually 1000 (as this is the usually the maximum rank for official TREC runs). Thus, when comparing two schemes this measure will tell us how many rank positions, on average, a relevant document is expected to change from scheme a to scheme b. Although different parts of the phenotype will impact on the fitness in different amounts (i.e. changes of rank at positions close to 1,000 will not change MAP significantly, while changes of rank in the top 10 may change MAP considerably) they are an important part in distinguishing the behaviour of the phenotype. The change in position at high ranks can tell us about certain features of the weighting scheme and the behaviour at these ranks.

To measure the difference a change in rank *could* make in terms of MAP, we modify the $dist(a, b)$ measure so that the change in rank of a relevant document is weighted similarly to how MAP weights relevant documents in a ranked list. This weighted distance measure, $w\_dist(a, b)$, is similar to the measure described in (Carterette and Allan 2005) and is calculated as follows:

$$w\_dist(a,b) = \frac{1}{|Q|}\sum_{q \in Q}\frac{1}{|R_q|}\sum_{i \in R_q}\begin{cases} \left|\frac{1}{lim} - \frac{1}{r_i(b)}\right| & \text{if } r_i(a) > lim \\ \left|\frac{1}{r_i(a)} - \frac{1}{lim}\right| & \text{if } r_i(b) > lim \\ \left|\frac{1}{r_i(a)} - \frac{1}{r_i(b)}\right| & \text{otherwise} \end{cases}$$

Table 1 Document collections

Collection	#Docs	#Words/doc	σ Doc length	#Topics
G-TRAIN	35,412	72.7	59.24	1–63
L-TRAIN	32,059	251.7	259.79	301–350
LATIMES	131,896	251.7	251.90	301–350
FBIS	130,471	249.9	554.42	351–400
FT91-93	138,668	221.8	196.44	401–450
OH90-91	148,162	81.4	64.02	1–63

where Q is the set of queries and R_q is the set of relevant documents for a query q. This measure tells us how the change in rank of a relevant document might affect MAP. It is entirely possible that two ranked lists could be considerably different yet have a similar MAP. This distance measure is thus more important if we wish to determine how the difference in the phenotype might affect fitness.

We use these distance measures to develop trees which show the distance between term-weighting schemes in the solution space. These trees are constructed from a distance matrix and a clustering algorithm (i.e. in our case the neighbour-joining method). For example, if we have N entities (solutions), we can create an $N \times N$ distance matrix using one of our distance measures. Then, using this distance matrix, we can then create a tree using a suitable drawing package (Choi et al. 2000) which represents the data and provides a visualisation of where our solutions lie in relation to each other.

Both Spearman's rank correlation and Kendall's tau are not technically suitable for measuring the parts of the phenotype that contribute exclusively to fitness in a training environment. For example, consider two ranked lists in which all the relevant documents for a query are positioned at the same ranks. If some non-relevant documents are positioned at different ranks, both of the aforementioned measures would indicate there is some difference in these solutions in the training environment. However, this would not be the case as they have actually placed the same relevant documents in the exact same positions and the GP process has actually identified the same solution.

4 Experimental setup

4.1 Document collections

The collections used in this research (Table 1) are subsets of the TREC collections used in standard IR evaluations. We use three different query types (short, medium and long) with these test collections. Short queries (s) use only the title field from the topics. Medium queries (m) use title and description fields, while long queries (l) use the title, description and narrative fields of the topics.

The G-TRAIN collection is used to evolve the global weighting scheme and the L-TRAIN collection is used to evolve the term-frequency factor and the normalisation schemes. The L-TRAIN collection has longer documents and the standard deviation (σ) of the document lengths is also greater, which provides a more varied environment in which to evolve the term-frequency and normalisation parts of the weighting scheme. The G-TRAIN collection consists of documents and topics (queries) from the OHSUMED collection while the L-TRAIN collection consists of documents and topics from the LATIMES collection.

Table 2 Terminals for each problem domain

Global terminals	Description
N	No. of documents in the collection
df	Document frequency of a term
cf	Collection frequency of a term
V	Vocabulary of collection (no. of unique terms)
C	Size of collection (total number of terms)
0.5, 1, 10	*The constants*, 0.5, 1 and 10
Parameters	7 Runs of a population of size 100 for 50 generations
Term-frequency Terminals	
tf	Raw term-frequency of a term
0.5, 1, 10	*The constants*, 0.5, 1 and 10
Parameters	7 Runs of a population of size 100 for 50 generations
Normalisation terminals	
l	Document vector length (unique terms)
l_{avg}	Average document vector length (unique terms)
l_{dev}	Standard deviation of document vector lengths (unique terms)
tl	Total document length (all terms)
tl_{avg}	Average total document length (all terms)
tl_{dev}	Standard deviation of total document lengths (all terms)
ql	Query vector length (unique terms)
qtl	Query total length (all terms)
0.5, 1, 10	*The constants*, 0.5, 1 and 10
Parameters	7 Runs of a population of size 200 for 25 generations

We previously used a subset of the OHSUMED collection to evolve global weighting schemes with success. However, when attempting to evolve local parts of the term-weighting scheme using this collection, we found the resultant schemes to be very specific to this collection as the document lengths and term-frequencies were not representative of most other TREC collections. It is also worth noting that for the normalisation problem we used 12 short, 13 medium and 12 long topics as it has been suggested that query length may have an impact on normalisation (HE and Ounis 2003). For the other weighting problems medium length queries were used. We conduct two-tailed t-tests on our test data to see if our evolved schemes are indeed different from the standard benchmarks at each stage.

4.2 GP terminal and function set

Table 2 shows the terminal set and some GP parameters for each of the three weighting problems. The set of functions used for *all* experiments is $F = \{\times, +, -, /, log, x^2, \sqrt{x}\}$, where x is some function or terminal. We ran the GP seven times for each of the three problems (global weighting, term-frequency and normalisation). Together with the two benchmarks, this gives us nine solutions in total for each of the three problems. In preliminary tests, we determined that the normalisation problem needs a larger population in order to evolve good solutions. This is most likely due to the size of the terminal set and problem domain. It is well known that document normalisation is a difficult problem and there has been much research into automatic tuning for document normalisation.

Table 3 % MAP for all global weightings on training data

Collection	idf	idf_{rsj}	gw_1	gw_2	gw_3	gw_4	gw_5	gw_6	gw_7
G-TRAIN	19.83	19.98	22.05	21.98	21.60	21.69	20.11	20.11	20.75

5 Results

5.1 Global term-weighting

We use two benchmarks against which to evaluate our evolved global schemes. The first scheme is the idf as found in the Pivoted Normalisation scheme, while the second scheme is the idf_{rsj} as found in the BM25 scheme (Singhal 2001).

$$idf = log\left(\frac{N+1}{df_t}\right) \qquad idf_{rsj} = log\left(\frac{N - df_t + 0.5}{df_t + 0.5}\right) \tag{2}$$

Table 3 shows the seven global weighting schemes (gw_i) evolved on our training data. We can see that all the evolved schemes are better than our benchmarks in terms of MAP on our training set. Figure 2 shows the trees derived from the distances between all the weighting schemes using both distance measures.

The better evolved schemes (gw_1 to gw_4) are clustered closer together. For example, $dist(idf_{rsj}, gw_1)$ is 34.32 which means that a relevant document moves on average 34.32 rank positions between the schemes. $w\_dist(idf_{rsj}, gw_1)$ is 0.0415 which is related to the possible difference in MAP. Figure 2 seems to indicate that the solutions are evolving to-wards ranked lists produced by gw_1 as there is an increase in performance as we get closer to the best solution. Obviously, phenotypically close solutions will have a similar fitness but it is not necessarily true that solutions with a similar fitness will have a similar phenotype (i.e. ranked list). On the test collections for all medium and long queries (Tables 5 and 6), we can see that the differences in MAP between the evolved weightings and idf_{rsj} are all statistically significant ($p < 0.05$) using a paired two-tailed t-test. It is worth noting that short queries (Table 4) are less affected by a global weighting (which is redundant when the query length is one) and thus perform similarly. However, an interesting point for short queries is that idf_{rsj} is significantly different to the idf of the pivoted normalisation scheme, although the actual difference in MAP is very small. This indicates differences between our distance measures and statistical tests.

On the test data, we can see that gw_1 to gw_4 perform somewhat similarly. gw_5 and gw_6 still perform slightly better than idf_{rsj}, while gw_7 still performs slightly better than these

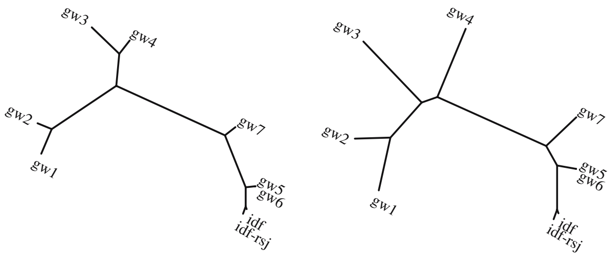

Fig. 2 Trees for global weightings using $dist$ and w_dist respectively

Table 4 % MAP for idf and global weightings on unseen data for short queries

Collection	Topics	idf	idf_{rsj}	gw_1	gw_2	gw_3	gw_4	gw_5	gw_6	gw_7
LATIMES	301–350 (s)	17.83	17.91	17.60	18.12	18.90	18.85	18.68	18.68	18.82
FBIS	351–400 (s)	11.19	11.24	11.65	11.72	11.76	11.68	11.41	11.41	11.32
FT91-93	401–450 (s)	21.69	21.69	21.80	21.79	22.42	22.89	21.57	21.57	21.74
p-value ≈	150 Topics	0.001	–	0.822	0.909	0.316	0.177	0.342	0.342	0.296

Table 5 % MAP for idf and global weightings on unseen data for medium queries

Collection	Topics	idf	idf_{rsj}	gw_1	gw_2	gw_3	gw_4	gw_5	gw_6	gw_7
LATIMES	301–350 (m)	19.11	19.16	21.80	22.49	23.48	22.98	20.92	20.92	21.12
FBIS	351–400 (m)	10.30	10.41	15.16	15.68	14.55	14.33	11.61	11.61	11.72
FT91-93	401–450 (m)	27.38	28.15	27.52	27.86	27.56	27.92	27.04	27.04	27.10
OH90-91	1–63 (m)	21.68	21.72	24.96	25.69	25.02	25.28	22.96	22.96	23.68
p-value ≈	213 Topics	0.272	–	0.004	0.0001	0.0001	0.0001	0.018	0.018	0.021

Table 6 % MAP for idf and global weightings on unseen data for long queries

Collection	Topics	idf	idf_{rsj}	gw_1	gw_2	gw_3	gw_4	gw_5	gw_6	gw_7
LATIMES	301–350 (l)	13.57	13.79	21.60	24.27	24.78	24.30	16.37	16.37	16.63
FBIS	351–400 (l)	06.76	06.97	12.30	13.32	14.07	13.84	08.34	08.34	09.01
FT91-93	401–450 (l)	23.11	23.13	27.17	28.28	28.31	29.13	24.95	24.95	25.80
p-value ≈	150 Topics	0.300	–	0.001	0.001	0.001	0.001	0.001	0.001	0.001

again. It should be noted that gw_1 seems to have over-trained slightly as it does not perform as well on some collections for longer queries, although it still performs much better than the benchmarks.

As gw_2 is very close to the best solution on the training set (gw_1) and is elegant in form (rewritten in (3)), we choose it to form part of the entire weighting scheme. Of course we could choose any one of the better performing weighting schemes upon which to evolve the remaining parts of the term-weighting scheme. It may also worth noting that we have since determined that gw_4 may be slightly better choice for weighting terms on certain collections (Cummins and O'Riordan 2006c).

$$gw_2 = \frac{cf^2\sqrt{cf}}{df^3} \tag{3}$$

5.2 Term-frequency factor

To evolve the term-frequency influence factor, we assume an average length document (i.e. no normalisation). We evolve the term-frequency factor while keeping the global weighting constant (i.e. gw_2). We compare our evolved weighting scheme against the Pivoted Document Normalisation and the BM25 scheme assuming average length documents (i.e. $s = 0$ and $b = 0$ respectively).

$$Piv_{s=0} = log(1 + log(1 + tf)) \cdot idf \qquad BM25_{b=0} = \frac{tf}{1.2 + tf} \cdot idf_{rsj} \tag{4}$$

Table 7 % MAP for benchmarks and $gw_2 \cdot tf_i$ influences on training data

Collection	$Piv_{s=0}$	$BM25_{b=0}$	tf_1	tf_2	tf_3	tf_4	tf_5	tf_6	tf_7
L-TRAIN	14.10	21.74	26.91	24.37	26.92	24.37	26.82	27.16	24.37

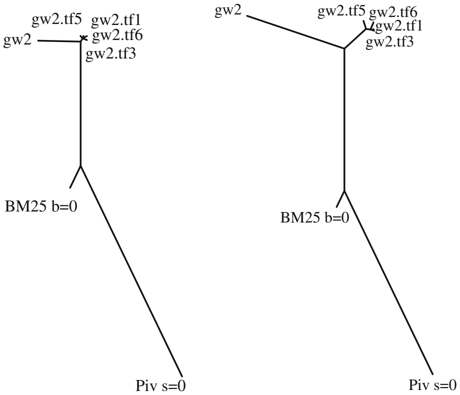

Fig. 3 Trees for term-frequency weightings using $dist$ and w_dist

Table 7 shows the MAP of the seven term-influence weighting functions evolved on the training collection. It is important to note that all our evolved term-frequency factors (tf_i) are used in conjunction with gw_2. We can see that $Piv_{s=0}$ performs poorly compared to $BM25_{b=0}$ indicating that the term-frequency factor is poor for the pivoted normalisation scheme. We can see that schemes tf_1, tf_3, tf_5 and tf_6 have a similar MAP. tf_2, tf_4 and tf_7 have the same MAP but actually produce a constant weighting for the term-frequency part (i.e. no tf terminal was present in the solution) and thus perform as a binary weighting (these three schemes are represented in Fig. 3 as gw_2). When we look at the differences in the phenotypes, we see that the best term-influence schemes (tf_1, tf_3, tf_5 and tf_6) behave similarly on the training collection. We can also see that the difference between the benchmarks and our weighting schemes is quite large. Taking the best term-frequency factor from the training set (tf_6), we can see that it can be simplified to show that it has characteristics similar to in form to that of the term-frequency part of the $BM25_{b=0}$ weighting. Examples of the distances between solutions in the trees in Fig. 3 are: $dist(BM25_{b=0}, gw_2 \cdot tf_6)$ is 84.41 and $w\_dist(BM25_{b=0}, gw_2 \cdot tf_6)$ is 0.0558.

$$tf_6 = log\left(\frac{10}{\sqrt{(0.5/tf) + 0.5}}\right) = log\left(\sqrt{\frac{200 \cdot tf}{1 + tf}}\right) \tag{5}$$

Our evolved schemes thus far (especially for medium and long queries) are considerably better than the benchmarks at this point. As all of the best term-frequency schemes are very similar in behaviour and MAP on both the training and test data (Tables 8–10), we choose the best performing scheme on the training data (tf_6) as a suitable term-frequency factor. Again, any of the suitable term-frequency factor could have been used. Our term-weighting formula now consists of $gw_2 \cdot tf_6$.

Table 8 % MAP for benchmarks and $gw_2 \cdot tf_i$ on unseen data for short queries

Collection	Topics	$Piv_{s=0}$	$BM25_{b=0}$	tf_1	tf_3	tf_5	tf_6
LATIMES	301–350 (s)	20.95	24.75	24.58	24.47	24.49	24.89
FBIS	351–400 (s)	16.30	19.98	20.30	20.15	20.40	20.27
FT91-93	401–450 (s)	22.50	31.38	31.10	31.14	31.11	31.35
p-value ≈	150 Topics	0.001	–	0.944	0.999	0.947	0.797

Table 9 % MAP for benchmarks and $gw_2 \cdot tf_i$ on unseen data for medium queries

Collection	Topics	$Piv_{s=0}$	$BM25_{b=0}$	tf_1	tf_3	tf_5	tf_6
LATIMES	301–350 (m)	13.80	20.55	24.31	24.35	24.30	24.38
FBIS	351–400 (m)	13.40	13.47	19.44	18.96	19.50	19.06
FT91-93	401–450 (m)	23.62	33.03	32.35	33.36	32.97	32.37
OH90-91	1–63 (m)	18.40	25.36	28.79	28.66	28.64	28.80
p-value ≈	213 Topics	0.001	–	0.001	0.001	0.001	0.001

Table 10 % MAP for benchmarks and $gw_2 \cdot tf_i$ on unseen data for long queries

Collection	Topics	$Piv_{s=0}$	$BM25_{b=0}$	tf_1	tf_3	tf_5	tf_6
LATIMES	301–350 (l)	10.94	13.98	25.86	26.02	25.92	25.87
FBIS	351–400 (l)	08.45	08.35	16.32	16.62	16.52	16.25
FT91-93	401–450 (l)	19.36	26.59	30.65	30.82	30.50	30.72
p-value ≈	150 Topics	0.001	–	0.001	0.001	0.001	0.001

5.3 Normalisation

To evolve the normalisation scheme, we assume a normalised term-frequency similar to the BM25 model. We compare our full evolved weighting scheme against the full Pivoted Document Normalisation and BM25 schemes (Singhal 2001).

$$Piv = \frac{log(1 + log(1 + tf)) \cdot idf}{(1 - s) + s \cdot \frac{tl}{tl_{avg}}} \quad BM25 = \frac{tf \cdot idf_{rsj}}{tf + (k_1.((1 - b) + b.\frac{tl}{tl_{avg}}))} \tag{6}$$

We set $s = 0.2$ for the pivoted normalisation scheme and set $b = 0.75$ and $k_1 = 1.2$ for the BM25 scheme as these are the default values (Jones et al. 2000). We evolve the normalisation factor n_i in the following formula:

$$gw_2 \cdot n_i tf_6 = gw_2 \cdot log\left(\sqrt{\frac{200 \cdot \frac{tf}{n_i}}{1 + \frac{tf}{n_i}}}\right) \tag{7}$$

As a naming convention, we will call the complete scheme $gw_2 \cdot n_i tf_6$ when n_i is the normalisation factor chosen. Table 11 shows the MAP of the seven evolved normalisation functions on the training data. We can see that n_4 is the best normalisation scheme on the training data and from Fig. 4 we can see that n_6 is one of its closest neighbours. n_7 performs quite well on the training data but seems to lie in a different part of the solution space to n_4 and n_6. Another point to notice is that, on the training collection and the unseen test collections,

Table 11 % MAP for benchmarks and $gw_2 \cdot n_i tf_6$ on training data

Collection	Piv	$BM25$	n_1	n_2	n_3	n_4	n_5	n_6	n_7
LOCAL-T	26.12	28.02	29.97	30.59	29.44	31.31	30.08	31.01	31.15

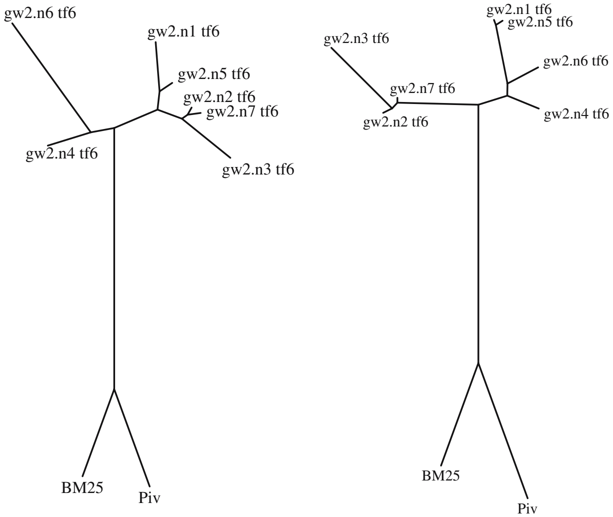

Fig. 4 Trees for normalisation weightings using $dist$ and w_dist

Table 12 % MAP for benchmarks and $gw_2 \cdot n_i tf_6$ on unseen data for short queries

Collection	Topics	Piv	$BM25$	n_1	n_2	n_3	n_4	n_5	n_6	n_7
LATIMES	301–350 (s)	24.26	24.17	22.76	23.38	23.32	23.86	22.80	23.87	24.44
FBIS	351–400 (s)	15.90	17.55	17.58	18.86	18.49	19.56	17.46	19.89	18.94
FT91-93	401–450 (s)	30.38	31.27	30.69	33.36	33.35	33.82	30.57	33.98	34.07
p-value ≈	150 Topics	0.195	–	0.310	0.503	0.557	0.167	0.290	0.127	0.106

normalisation considerably aids both benchmarks bringing them closer in terms of MAP to our weighting schemes.

On the unseen test data for short queries (Table 12), we can see that the three best schemes on the training data (n_6, n_7 and n_4) are still the best performing although none are significantly different to the BM25 scheme using a two-tailed t-test. For medium length queries (Table 13) we can see that n_4 and n_6 perform similarly and are both significantly different to the best benchmark ($BM25$) for medium length queries. On longer queries, n_4 is the only scheme that is significantly different than BM25. n_7 does not perform as well as n_4 or n_6 on medium and long queries (Table 14) and seems to have evolved to an area of the search space that contains useful features for shorter queries.

We can see from both trees that n_2 is phenotypically close to n_7. These two schemes are also comparable in terms of MAP on most of the unseen test data (especially medium and long queries). n_1 and n_5 are also phenotypically close and perform similarly on most of the unseen data for all query lengths. As an example of the distances between solutions in the trees in Fig. 4 $dist(BM25, gw_2 \cdot n_4 tf_6)$ is 71.42 and $w\_dist(BM25, gw_2 \cdot n_4 tf_6)$ is 0.0947.

Table 13 % MAP for benchmarks and $gw_2 \cdot n_i tf_6$ on unseen data for medium queries

Collection	Topics	Piv	$BM25$	n_1	n_2	n_3	n_4	n_5	n_6	n_7
LATIMES	301–350 (m)	25.48	25.61	27.21	28.03	26.89	28.42	26.99	28.64	27.69
FBIS	351–400 (m)	17.92	19.53	19.99	21.20	21.12	23.00	22.07	24.26	21.13
FT91-93	401–450 (m)	34.47	35.33	35.40	36.31	36.12	36.23	35.36	36.57	36.33
OH90-91	1–63 (m)	26.76	28.08	28.07	29.77	29.70	29.66	28.13	29.84	30.08
p-value ≈	213 Topics	0.006	–	0.613	0.090	0.206	0.043	0.455	0.011	0.070

Table 14 % MAP for benchmarks and $gw_2 \cdot n_i tf_6$ on unseen data for long queries

Collection	Topics	Piv	$BM25$	n_1	n_2	n_3	n_4	n_5	n_6	n_7
LATIMES	301–350 (s)	25.79	26.77	31.98	30.67	28.84	31.58	31.33	30.80	30.71
FBIS	351–400 (s)	17.59	20.03	21.48	21.83	18.99	24.60	21.51	24.21	21.63
FT91-93	401–450 (s)	34.49	35.35	35.73	36.32	35.20	37.08	35.80	36.86	36.62
p-value ≈	150 Topics	0.025	–	0.121	0.152	0.810	0.041	0.182	0.055	0.190

$$n_4 = \frac{l \times qtl}{10 \cdot l_{avg}} \quad n_6 = \frac{\sqrt{log(qtl)} \times log(qtl) \times l}{l_{avg}} \quad n_7 = \frac{tl}{tl_{dev} + \frac{l}{qtl}} \tag{8}$$

The results obtained from these normalisation schemes are somewhat inconclusive as only two of the better schemes lie in a similar area (n_4 and n_6). We can however indicate that schemes close to n_4 and n_6 seems to perform well on a wide variety of collections for varying query lengths. Although, on two collections using short queries and one collection using medium length queries n_7 is the best performing scheme.

6 Conclusion

We have introduced two metrics that measure the distance between the ranked lists returned by different term-weighting schemes. These measures are useful for determining the closeness of term-weighting schemes and for analysing the solutions without the need to analyse the exact form (genotype) of a term-weighting scheme (although a number of schemes are shown to aid clarity). This framework can be used for all types of term-weighting schemes and also fits well with the GP paradigm.

The distance matrices produced from these distance measures can be used to produce trees. The two measures outlined are quite similar as the trees produced have a similar form indicating that they provide similar information about relative distances between phenotypes. The trees produced are also useful in determining the relative performance of the solutions on general test data. We have also shown that the best evolved weighting schemes lie in a different area of the solution space than current benchmarks schemes.

However, the normalisation schemes evolved are somewhat specific to certain query types and collections. Normalisation is known to be a difficult problem in IR and we intend to further our study into normalisation schemes using GP. The normalisation schemes in this research were evolved on one single training collection. It is known that normalisation can be tuned in a collection specific way (Chowdhury et al. 2002). We intend to use our GP framework to evolve normalisation schemes over multiple varied collections in order to find a more general normalisation scheme.

We also intend analysing our schemes in a more formal axiomatic framework. It is hoped that this work will motivate the use of these schemes. We hope to show that our weighting schemes satisfy a number of constraints to which all good term-weighting schemes should adhere.

Acknowledgements This work is being carried out with the support of IRCSET (the Irish Research Council for Science, Engineering and Technology) under the Embark Initiative. The authors would also like to thank the reviewers for their useful comments.

References

Buckley C, Voorhees EM (2000) Evaluating evaluation measure stability. In: SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval. New York, NY, USA, ACM Press, pp 33–40

Carterette B, Allan J (2005) Incremental test collections. In: CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management. New York, NY, USA, ACM Press, pp 680–687

Choi J-H, Jung H-Y, Kim H-S, Cho H-G (2000) PhyloDraw: a phylogenetic tree drawing system. Bioinformatics 16(11):1056–1058

Chowdhury A, McCabe MC, Grossman D, Frieder O (2002) Document normalization revisited. In: SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, USA, ACM Press, pp 381–382

Cummins R, O'Riordan C (2006a) An analysis of the solution space for genetically programmed term-weighting schemes in information retrieval. In: Bell PMD, Sage P (eds) 17th artificial intelligence and cognitive science conference (AICS 2006). Queen's University, Belfast, Northern Ireland

Cummins R, O'Riordan C (2006) Evolving local and global weighting schemes in information retrieval. Inform Retrieval 9(3):311–330

Cummins R, O'Riordan C (2006c) A framework for the study of evolved term-weighting schemes in information retrieval. In: Stein B, Kao O (eds) TIR-06 text based information retrieval, workshop. ECAI 2006. Riva del Garda, Italy

Fan W, Gordon MD, Pathak P (2004) A generic ranking function discovery framework by genetic programming for information retrieval. Inform Proces Manage 40(4):587–602

Fang H, Zhai C (2005) An exploration of axiomatic approaches to information retrieval. In: SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval. New York, NY, USA, ACM Press, pp 480–487

HE B, Ounis I (2003) A study of parameter tuning for term frequency normalization. In: CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management. New York, NY, USA, ACM Press, pp 10–16

Jones KS, Walker S, Robertson SE (2000) A probabilistic model of information retrieval: development and comparative experiments. Inf Process Manage 36(6):779–808

Koza JR (1992) Genetic programming: on the programming of computers by means of natural selection. MIT Press, Cambridge, MA, USA

Luke S (2001) When short runs beat long runs. In: Proceedings of the genetic and evolutionary computation conference (GECCO-2001). San Francisco, California, USA, Morgan Kaufmann, pp 74–80

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

Ponte JM, Croft WB (1998) A language modeling approach to information retrieval. In: SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval. New York, NY, USA, ACM Press, pp 275–281

Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval. Inform Process Manage 24(5):513–523

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

Singhal A (2001) Modern information retrieval: a brief overview. Bull IEEE Comput Soc Tech Comm Data Eng 24(4):35–43

Trotman A (2005) Learning to rank. Inform Retrieval8:359–381

Zobel J, Moffat A (1998) Exploring the similarity space. SIGIR Forum 32(1):18–34