A Framework for the study of Evolved Term-Weighting Schemes in Information Retrieval

Ronan Cummins and **Colm O'Riordan** [1]

Abstract. Evolutionary algorithms and, in particular, Genetic Programming (GP) are increasingly being applied to the problem of evolving term-weighting schemes in Information Retrieval (IR). One fundamental problem with the solutions generated by these stochastic processes is that they are often difficult to analyse. A number of questions regarding these evolved term-weighting schemes remain unanswered. One interesting question is; do different runs of the GP process bring us to similar points in the solution space?

This paper deals with determining a number of measures of the distance between the ranked lists (phenotype) returned by different term-weighting schemes. Using these distance measures, we develop trees that show the phenotypic distance between these term-weighting schemes. This framework gives us a representation of where these evolved solutions lie in the solution space.

Finally, we evolve several global term-weighting schemes and show that this framework is indeed useful for determining the relative closeness of these schemes and for determining the expected performance on general test data.

1 INTRODUCTION

Information retrieval (IR) is concerned with the return of relevant documents from a collection of unstructured documents given a user need. It has been recognized that the effectiveness of vector space approaches to IR depend crucially on the term weighting applied to the terms of the document vectors [15]. These term-weights are typically calculated using term-weighting schemes that assign values to 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) is a biologically inspired search algorithm useful for searching large complex spaces. Inspired by the theory of natural selection, the GP process creates a random population of solutions. These solutions, encoded as trees, undergo generations of selection, reproduction and mutation until suitable solutions are found. As GP is a non-deterministic algorithm it cannot be expected to produce a similar solution each time. Restart theory in GP suggests that it is necessary to restart the GP a number of times in order to achieve good solutions [9]. 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?

Recently, IR fusion techniques, that use the rankings from several retrieval systems to determine the final document ranking, have been shown to increase the performance of IR systems [16]. These techniques only work when the ranked lists from the different retrieval systems return different ranked lists. Thus, when new term-weighting schemes are developed it is important, in many respects, to determine if these new schemes are similar to existing ones in terms of the ranked lists produced, or if indeed they belong to a new family of weighting scheme.

This paper presents a framework for evaluating the distance between the ranked lists produced from different term-weighting schemes in order to understand the relative closeness of these schemes. We develop two different distance measures and show that they are useful in determining how the term-weighting schemes are expected to perform in a general environment. We use these distance measures to create trees visualizing the distances between the weighting schemes.

Section 2 of this paper introduces term-weighting schemes useful for determining the discrimination value of a term. Section 3 introduces the GP process and existing approaches using GP to evolve term-weighting schemes are also discussed. Section 4 introduces our framework and outlines two distance measures. Our experimental setup is outlined in section 5 while section 6 discusses our results. Finally, our conclusions and future work are summarised in section 7.

2 INFORMATION RETRIEVAL

2.1 Term-Weighting for vector models

Term-weighting schemes assign values to terms based on measures of the term in both a global (collection-wide) and local (document-specific) context. Yu and Salton [19] suggest that the best distinguishing terms are those which 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 a term weighting function should vary directly with term frequency and inversely with document frequency. The *idf* scheme, first introduced by Sparck Jones [17], gives a higher weight to terms that occur in fewer documents. The original *idf* measure is often calculated as follows:

$$idf = log(\frac{N+1}{df_t}) \quad (1)$$

where N is the number of documents in the collection and df_t is the number of documents containing term t. A modern weighting scheme developed by Robertson et al. [13] is the BM25 weighting scheme. The global part of this weighting scheme is a variation of the traditional *idf* measure and is calculated as follows:

$$idf_{rsj} = log(\frac{N - df_t + 0.5}{df_t + 0.5}) \quad (2)$$

[1] University of Ireland, Galway. email: ronan.cummins@nuigalway.ie, colmor@it.nuigalway.ie

The idf measure forms the basis of many modern term-weighting schemes as it determines what initial weight a search term should receive [12]. It is worth noting that documents are typically not retrieved by idf only, and are usually used in conjunction with local measures to aid retrieval performance. However, if we can firstly find out what initial weight a search term should be given, we can then improve upon this by looking at the within-document characteristics to further improve retrieval performance. Developing global weighting schemes separately has been shown to benefit the performance of IR systems [11, 4, 14] and is an important goal in developing full weighting schemes which include local characteristics, like term-frequency and document normalisation. These idf type schemes are also used in many other domains within IR to weight features (e.g. document classification).

3 GENETIC PROGRAMMING

Genetic Programming [8] is a stochastic searching algorithm, inspired by natural selection. In the GP process, a population of solutions is created randomly (although some approaches seed the initial population with certain known solutions). The 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). 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. Crossover is the main reproductive mechanism in GP. When two solutions are selected from the selection process, their genotypes are combined to create a new individual. One point crossover is the norm for genetic programming. This is where a single point is located in both parents and the sub-trees are swapped at these points to create two new solutions. Mutation (asexual reproduction) is the random change of the value of a gene (or the change of a subtree) to create a new 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 after a predefined number of generations. Bloat is a common phenomenon in GP. Bloat is where solutions grow in size without a corresponding increase in fitness.

3.1 Phenotype

The phenotype of the individual is often described as its behaviour. 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 map to the same phenotype, and different phenotypes can have the same fitness. For most problems in GP in an unchanging environment, identical genotypes will map to identical phenotypes which will have the same fitness.

3.2 Previous Research

GP techniques have previously been adopted to evolve weighting functions and are shown to outperform standard weighting schemes in an adhoc framework [6, 10, 18, 4]. However, in many of these approaches a critical analysis of the solutions evolved is not presented. It is important to gain an understanding of the solutions obtained from these evolutionary processes and have a means of rating the differences between the schemes.

In [7], differences in retrieval systems are analysed using the ranked lists returned from the various systems. The distance between two ranked lists is measured using the number of out-of-order pairs. Using the measure it can then be determined if two systems are in essence the same (i.e. if they return the same ranked lists for a set of queries). Spearman's rank correlation and Kendall's tau are two common correlations that measure the difference between ranked sets of data. Both Spearman's rank correlation and Kendall's tau use all of the ranked data in a pair of ranked lists.

4 FRAMEWORK

4.1 Phenotypic Distance Measures

Figure 1 shows how the GP paradigm is adopted to evolve term-weighting schemes in IR. We use mean average precision (MAP) as our fitness function as it is a commonly used metric to evaluate the performance of IR systems and is known to be a stable measure [1]. Furthermore, it has been used with success in previous research evolving term-weighting schemes in IR [6, 18].

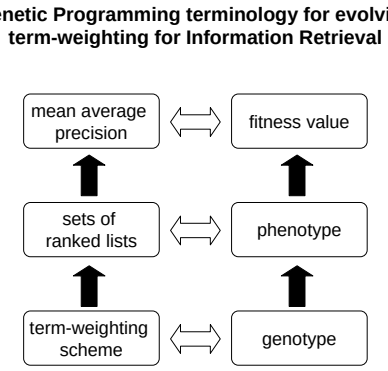


Figure 1. GP for Information Retrieval

For our framework, we measure the phenotype of our solutions by examining the sets of ranked lists returned by the term-weighting solution for a set of topics on a document collection (its environment). Spearman's rank correlation uses all available document ranks from two ranked lists and not just the ranks of relevant documents. We wish to develop distance measures for 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.

To compare two sets of ranked lists, we introduce a measure which essentially measures the average difference between the ranks of relevant documents in two sets of ranked lists. In this measure, we ignore the ranks of non-relevant documents as they do not contribute to the fitness although they do technically contribute to the phenotype of the individual. 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 towards solutions

that promote similar features of relevant documents. Thus, one of the phenotypic distance measures ($dist(a,b)$), where a and b are two weighting schemes, is defined as follows:

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

where R is the set of relevant documents in the collection for all of the queries used and $r_i(a)$ is the rank position of relevant document i under weighting scheme a. lim is the maximum rank position available from a list and is usually 1000 (as this is the usually the maximum rank for official TREC runs). As a result, relevant documents that are ranked outside the top 1000 are treated as being at rank 1000. 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 for relevant documents at positions near 1000 do not significantly effect the MAP) 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 weighting scheme and the behaviour at these ranks.

We also develop a second measure of the distance between two ranked lists which takes into account the effect a change in rank has on MAP. To measure the actual 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 on how it effects MAP. This weighted distance measure ($w\_dist(a,b)$) is similar to the measure described in [2] and is calculated as follows:

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

where Q is the number of queries and R_q is the relevant documents for a query q. This measure tells us how a change in rank of a relevant document will affect the MAP (i.e. changes of rank at positions close to 1000 will not change the MAP significantly, while changes of rank in the top 10 may change MAP considerably). Of course, it is entirely possible that two ranked lists could be considerably different yet have a similar MAP, as they may be promoting different relevant documents.

4.2 Neighbour-joining trees

Neighbour-joining is a bottom-up clustering method often used for the creation of phylogenetic trees. However, we use the method to produce trees that represent solutions that are from *different* runs of our GP. The algorithm requires knowledge of the distance between entities that are to be represented in the tree. A distance matrix is created for the set of entities using a distance measure and the tree can then be produced from the resulting data. We use this clustering technique to visualize the phenotypic distance between the best solutions output by our GP. For example, if we have N entities or 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 [3] which represents the data and can provide a visualisation into where our solutions lie in relation to each other. This model is also well suited to our evolutionary paradigm. We use this technique simply to visualise the distance between our term-weighting solutions which are developed using GP.

5 EXPERIMENTAL SETUP

5.1 Approach Adopted

We evolve global term-weighting schemes in the following framework:

$$score(d,q) = \sum(gw_t \times qtf) \quad (3)$$

where $score(d,q)$ is the score a document d recieves in relation to a query q, gw_t is the global weighting and qtf is the frequency of the term in the query. All documents in the collection are scored in relation to the query and ranked accordingly. We are only evolving the global (term-discrimination) part of the weighting scheme as an example of our framework. However, the entirety of the term-weighting scheme can be evolved and analysed in a similar manner.

5.2 Training and Test Collections

We use collections from TREC disks 4 and 5 as our test collections. A different set of 50 TREC topics is used for each of the collections (apart from the Federal Register collection (FR) for which we use 100 TREC topics). For each set of topics we create a medium length query set (m), consisting of the title and description fields, and a long query set (l) consisting of the title, description and narrative fields. We also use documents from the OHSUMED collection as a test collection for medium length queries (OH90-91). We only use the topics in these sets that have relevant documents in the collection.

The TRAIN collection (used in training) consists of 35,412 documents from the OHSUMED collection and the 63 topics. The lengths of these topics range from 2 to 9 terms. Standard stop-words from the Brown Corpus[2] are removed and remaining words are stemmed using Porter's algorithm. No additional words are removed from the narrative fields as is the case in some approaches. Table 1 shows some characteristics of the document collections used in this research.

Table 1. Document Collections

Collection	#Docs	#words/doc	#Topics	medium	long
TRAIN	35,412	72.7	0-63	4.96	None
LATIMES	131,896	251.7	301-350	9.9	29.9
FBIS	130,471	249.9	351-400	7.9	21.9
FT91-93	138,668	221.8	401-450	6.5	18.7
FR	55,630	387.1	301-400	8.9	25.9
OH90-91	148,162	81.4	0-63	4.96	None

5.3 Terminal and Function Set

Tables 2 and 3 show the functions and terminals that are used in all runs of the GP.

5.4 GP parameters

We use MAP as our fitness function. All tests are run for 50 generations with an initial random population of 100 solutions on the training collection (TRAIN) detailed in Table 1. The tournament size is set to 3. We restrict all trees to a depth of 6. As a result this has the effect of reducing bloat, improving generalisation, reducing the search space and increasing the speed of the GP. As our highest order operator is binary, the longest individual we can have can contains 63

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

Table 2. Function Set

Function	Description
+, ×, /, -	arithmetic functions
log	the natural log
$\sqrt{}$	square-root function
sq	square

Table 3. Terminal Set

Terminal	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	*the constant* 0.5
1	*the constant* 1
10	*the constant* 10

genes $(2^6 - 1)$. We believe that this is a large enough space in which to find suitable term-weighting schemes. The creation type used is the standard ramped half and half creation method used by Koza [8]. We use an elitist strategy where the best individual is automatically transfered to the next generation. 4% mutation is used in our experiments. Due to the stochastic nature of GP a number of runs is often needed to allow the GP converge to a suitably good solution. We run the GP seven times and choose the best solution from each of those runs. This gives us seven evolved solutions and two benchmark solutions (1) (2) to use with our document collections.

6 RESULTS

Table 4 shows the MAP of seven global evolved weighting schemes (gw) on our training data in no particular order. We can see that all the evolved schemes are better than our benchmarks (idf and idf_{rsj}) in terms of MAP.

Table 4. % MAP for all global weightings

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

Figure 2 shows the best and average of the population from the two best runs of the GP (i.e. gw_1 and gw_2). It is worth noting that the best individual from the seven randomly created populations (i.e. generation 0) is not better than the best solution produced after the 50^{th} generation from the worst of the seven runs.

Tables 5, 6 and 7 shows the distance matrices for all the global weighting schemes for the training data using Spearman's rank correlation, $dist(a, b)$ and $w\_dist(a, b)$ measures respectively. Spearman's rank correlation gives us values in the range of -1 to $+1$ and uses all of the documents in the ranked list. As the Spearman correlations of the ranked lists produced by the global weighting scheme are all positively correlated, we simply use $1-$ Spearman's rank correlation as a distance measure. This will give us 1 if the lists are randomly correlated and 0 if they are identical. We use this correlation

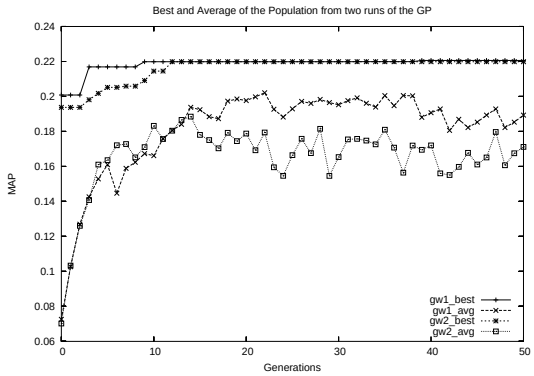


Figure 2. Best and Average Fitness for best two global runs

as a comparison to our distance measures that only look at distances of relevant documents.

The values in Table 6 indicate the average number of rank positions a relevant document changes. While the values in Table 7 indicate the maximum possible percentage MAP difference between two schemes. By looking at the difference between the ranked lists of each global weighting we can get an idea of the landscape of the solution space in the global domain.

Table 5. $1-$ spearman's rank correlation between all global weightings on TRAIN

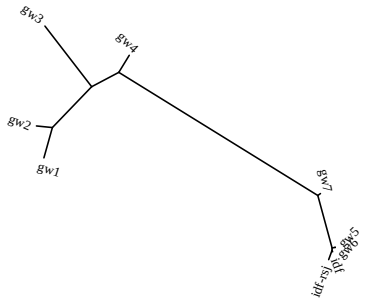
Scheme	idf	idf_{rsj}	gw_1	gw_2	gw_3	gw_4	gw_5	gw_6	gw_7
idf	00.00	0.014	0.524	0.501	0.503	0.396	0.007	0.007	0.078
idf_{rsj}		00.00	0.531	0.507	0.507	0.400	0.020	0.020	0.089
gw_1			00.00	0.059	0.186	0.174	0.523	0.523	0.451
gw_2				00.00	0.210	0.136	0.499	0.499	0.425
gw_3					00.00	0.170	0.501	0.501	0.435
gw_4						00.00	0.393	0.393	0.324
gw_5							00.00	00.00	0.076
gw_6								00.00	0.076
gw_7									00.00

Table 6. $dist$ measure between all global weightings on TRAIN

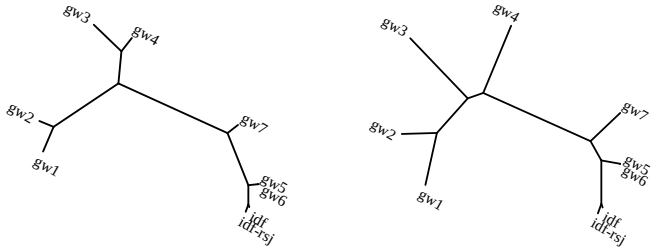
Scheme	idf	idf_{rsj}	gw_1	gw_2	gw_3	gw_4	gw_5	gw_6	gw_7
idf	00.00	01.27	36.50	35.07	32.81	30.31	03.65	03.65	11.15
idf_{rsj}		00.00	35.94	34.32	31.90	29.42	04.52	04.52	12.01
gw_1			00.00	05.07	21.26	18.46	35.34	35.34	28.13
gw_2				00.00	20.60	15.96	34.15	34.15	27.08
gw_3					00.00	06.62	30.48	30.48	24.66
gw_4						00.00	28.37	28.37	22.15
gw_5							00.00	00.00	09.16
gw_6								00.00	09.16
gw_7									00.00

Table 7. w_dist % measure between all global weightings on TRAIN

Scheme	idf	idf_{rsj}	gw_1	gw_2	gw_3	gw_4	gw_5	gw_6	gw_7
idf	00.00	00.21	04.30	04.17	04.10	03.91	00.99	00.99	01.78
idf_{rsj}		00.00	04.20	04.15	04.26	04.01	01.10	01.10	01.86
gw_1			00.00	01.33	02.73	03.16	04.23	04.23	04.25
gw_2				00.00	02.50	02.16	04.09	04.09	04.03
gw_3					00.00	02.64	03.96	03.96	03.95
gw_4						00.00	03.48	03.48	03.38
gw_5							00.00	00.00	01.18
gw_6								00.00	01.18
gw_7									00.00



1−Spearman's rank correlation



dist w_dist

Figure 3. Neighbour-Joining trees for global weightings

Firstly, from Figure 3 we can see that the phenotypic distance measures produce trees of a similar structure. The only difference in form is that gw_3 and gw_4 are clustered together directly using the unweighted $dist$ measure. It is important to note that the trees visualize different aspects of the ranked lists. For example, the distance between the top four performing schemes (gw_1 to gw_4) and the remaining schemes is greater in the tree created from Spearman's rank correlation than for the other two trees. This is because Spearman's rank correlation uses the ranks of non-relavant documents. Looking at the tree produced by the $dist(a, b)$ measure, we can see that gw_3 and gw_4 are quite similar in terms of the actual ranks of relevant documents. However, when looking at the tree produced by $w\_dist(a, b)$ for these two schemes, we can see that some of these differences are at low ranks as the possible difference in MAP is quite large.

In general, we can see that idf_{rsj}, idf, gw_5 and gw_6 are phenotypically close. Schemes gw_5 and gw_6 are actually phenotypically equivalent (i.e. return the same ranked lists) but not genotypically equivalent. The two versions of idf are very close. Schemes gw_1 and gw_2 are also phenotypically close while gw_3 and gw_4 are somewhat similar. An important point to note is that as we get phenotypically further from the best solution (gw_1) we see a relative drop in MAP on our training collection. This indicates that the solutions are evolving towards the ranked lists (on the training set) that are produced by gw_1. Obviously, phenotypically close solutions will have a similar fitness but it is not neccessarily true that solutions with a similar fitness will have a similar phenotype (e.g. as one can imagine that there exists many poor performing functions which return equally bad but different ranked lists). It is worth noting that these trees should be produced from the training data as this is the environment where the solutions were evolved. However, these trees can help us to predict the behaviour of the schemes on general data (if our training data is a representative sample).

Tables 8 and 9 show the MAP of all schemes for unseen test data on medium and long queries. Firstly, we can see that the differences in MAP between the evolved weightings and idf_{rsj} are all statistically significant ($p < 0.05$) using a two-tailed t-test. Both version of idf perform similarly as expected. We can see that gw_1 is no longer the best evolved weighting scheme, although it is still significantly better than idf. Schemes gw_2, gw_3 and gw_4 are now the best performing schemes on most of the collections. Schemes gw_5 and gw_6 still perform only slightly better than idf_{rsj}, while gw_7 still performs slightly better than these again. It would seem that gw_1 has overtrained slightly on the training collection. It is also worth pointing out that our training set seems to be quite general as most of the schemes perform similarly on test data. If we look at the genotypes of some of the schemes it leads us to a similar conclusion. We have re-written the following formulas in a more intuitive manner to provide transparency to the process. As a result, the re-written formulas may also be shorter (in depth) that those that were evolved originally.

$$gw_1 = \frac{V^2 cf^2 \sqrt{cf}}{C.df^3} + \sqrt{cf} \qquad gw_2 = \frac{cf^2\sqrt{cf}}{df^3}$$

$$gw_3 = \sqrt{(log(\frac{cf}{df}))^2 \times \frac{N}{df} \times (\frac{N^2}{df} + 1)}$$

$$gw_4 = \sqrt{\frac{cf^3 N}{df^4}} \qquad gw_5 = \sqrt{\sqrt{\frac{0.5}{df}}}$$

$$gw_6 = \sqrt{\frac{\sqrt{df}}{df}} \qquad gw_7 = \sqrt{\frac{\sqrt{cf/N}}{df^2}}$$

We can see that gw_1 is a more specific form of gw_2. Schemes gw_5 and gw_6 are an example of two different genotypes producing the same ranked lists. gw_6 will produce a score that is always double that of gw_5. We are evolving towards a ranked list on the training collection that is produced by the best two schemes (gw_1 and gw_2). The gw_2 scheme is a more general form of gw_1 and performs consistently better on our test data. The gw_3 scheme contains a problematic $log(cf/df)$ that will assign certain low frequency terms a zero weight [5] and makes it a poor choice for weighting in a retrieval context. This can be seen on the results for the FR collection when compared to one of its nearest neighbours gw_4. When looking at the individual queries for this collection (FR), we have determined that the difference between gw_4 and the other top schemes (gw_1 to gw_3) is only large for a very small number of queries. As a result it can be

Table 8. % MAP for idf and global weightings 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
FR	301-400 (m)	25.87	24.89	25.12	25.71	21.31	28.72	25.49	25.49	27.39
OH90-91	0-63 (m)	21.68	21.72	24.96	25.69	25.02	25.28	22.96	22.96	23.68
$\approx p\text{-}value$	241 Topics	0.272	-	0.004	0.0001	0.0001	0.0001	0.018	0.018	0.021

Table 9. % MAP for idf and global weightings 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
FR	301-400 (l)	16.23	16.95	22.78	22.75	20.86	27.83	19.84	19.84	19.92
$\approx p\text{-}value$	241 Topics	0.300	-	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001

concluded that gw_4 promotes certain useful features that are different than those of the rest of the schemes. These differences are noticeable on the FR collection because of its makeup. The gw_4 scheme seems to be a particularly robust global weighting scheme as shown on the test data. The difference between gw_4 and gw_3, for example, is not statistically significant. However, we know that gw_4 has advantagous retrieval features (as seen on the FR collection) for certain (albeit few) queries.

7 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. This framework can be used for all types of term-weighting schemes and also fits well into the genetic programming paradigm.

The distance matrices produced from these distance measures can be used to produce trees that aid visualization of the solution space. The trees produced are also useful in determining the relative performance of the solutions on general test data. We have also shown that all the evolved global weighting schemes produced are evolving to a area of the solution space that is different from the types of idf currently being used to measure the discrimination value of a term. In future work, we intend to apply this framework to analyse entire term-weighting schemes which have been evolved.

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.

REFERENCES

[1] Chris Buckley and Ellen M. Voorhees, 'Evaluating evaluation measure stability', in *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 33–40, New York, NY, USA, (2000). ACM Press.

[2] Ben Carterette and James Allan, 'Incremental test collections', in *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pp. 680–687, New York, NY, USA, (2005). ACM Press.

[3] Jeong-Hyeon Choi, Ho-Youl Jung, Hye-Sun Kim, and Hwan-Gue Cho, 'Phylodraw: a phylogenetic tree drawing system.', *Bioinformatics*, **16**(11), 1056–1058, (2000).

[4] Ronan Cummins and Colm O'Riordan, 'An evaluation of evolved term-weighting schemes in information retrieval.', in *CIKM*, pp. 305–306, (2005).

[5] Ronan Cummins and Colm O'Riordan, 'Evolving general term-weighting schemes for information retrieval: Tests on larger collections.', *Artif. Intell. Rev.*, **24**(3-4), 277–299, (2005).

[6] Weiguo Fan, Michael D. Gordon, and Praveen Pathak, 'A generic ranking function discovery framework by genetic programming for information retrieval', *Information Processing & Management*, (2004).

[7] P. Kantor, K. Ng, and D. Hull. Comparison of system using pairs-out-of-order, 1998.

[8] John R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, USA, 1992.

[9] Sean Luke, 'When short runs beat long runs', in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pp. 74–80, San Francisco, California, USA, (7-11 2001). Morgan Kaufmann.

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

[11] A. Pirkola and K. Jarvelin, 'Employing the resolution power of search keys', *J. Am. Soc. Inf. Sci. Technol.*, **52**(7), 575–583, (2001).

[12] S. E. Robertson and S. Walker, 'On relevance weights with little relevance information', in *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 16–24. ACM Press, (1997).

[13] Stephen E. Robertson, Steve Walker, Micheline Hancock-Beaulieu, Aarron Gull, and Marianna Lau, 'Okapi at TREC-3', in *In D. K. Harman, editor, The Third Text REtrieval Conference (TREC-3) NIST*, (1995).

[14] Dmitri Roussinov, Weiguo Fan, and Fernando A. Das Neves, 'Discretization based learning approach to information retrieval.', in *CIKM*, pp. 321–322, (2005).

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

[16] Alan F. Smeaton, 'Independence of contributing retrieval strategies in data fusion for effective information retrieval.', in *BCS-IRSG Annual Colloquium on IR Research*, (1998).

[17] Karen Sparck Jones, 'A statistical interpretation of term specificity and its application in retrieval', *Journal of Documentation*, **28**, 11–21, (1972).

[18] Andrew Trotman, 'Learning to rank', *Information Retrieval*, **8**, 359 – 381, (2005).

[19] C. T. Yu and G. Salton, 'Precision weighting - an effective automatic indexing method', *Journal of the ACM*, **23**(1), 76–88, (1976).