An Analysis of the Solution Space for Genetically Programmed Term-Weighting Schemes 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

Abstract. Evolutionary algorithms and Genetic Programming (GP) in particular are increasingly being applied to the problem of evolving term-weighting schemes in Information Retrieval (IR). One fundamental problem with the solutions generated by this stochastic, non-deterministic process is that they are often difficult to analyse.
We develop a number of 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. Using this framework we show that our evolved solutions lie in a different part of the solution space than two of the best benchmark term-weighting schemes available.

1 Introduction

Information retrieval (IR) is becoming increasingly more important as more and more information is becoming available on-line in unstructured formats. Term-weighting schemes are a crucial part of IR systems as they assign values to search terms based on how useful they are likely to be in determining the relevance of a document. The effectiveness of many approaches to IR depends crucially on the term weighting applied to the terms of the document vectors [11]. 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) [8] 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 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?

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 develop two different distance measures and show that they are useful in determining how term-weighting schemes are expected to perform in a general environment.

Section 2 of this paper introduces some GP terminology and existing approaches using GP to evolve term-weighting schemes in IR. Section 3 introduces the two distance measures developed. Our experimental setup is outlined in section 4, while section 5 discusses our results. Finally, our conclusions are summarised in section 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 the GP paradigm is used to evolve term-weighting schemes in 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 [1]. Furthermore, it has been used with success in previous research evolving term-weighting schemes in IR [6, 13, 5].

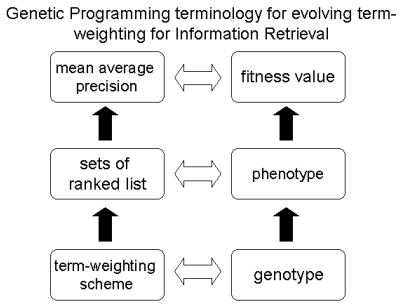

Fig. 1. GP for Information Retrieval

2.1 Previous Research

GP techniques have previously been adopted to evolve weighting functions and are shown to outperform standard weighting schemes in an ad hoc framework [6, 10, 13, 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 where in the solution space the best solutions lie.

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 this measure, it can 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. 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. The global part 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 terms. Both benchmarks used in this research fit this model as do most term-weighting schemes in IR.

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). We develop distance measures for only 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.

We introduce a measure which 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:

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

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 1000 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 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 the change in rank of a relevant document will affect MAP. It is entirely possible that two ranked lists could be considerably different yet have a similar MAP.

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 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 provides a visualisation of where our solutions lie in relation to each other.

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. 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.

4.2 GP Terminal and Function Set

The set of functions used for *all* experiments is $F = \{\times, +, -, /, log, square, square\text{-}root\}$. We ran the GP seven times for each of the three problems (global weighting, term-

Table 1. Document Collections

Collection	#Docs	#words/doc	σ doc length	#Topics	#words/topic
G-TRAIN	35,412	72.7	59.24	1-63	4.96
L-TRAIN	32,059	251.7	259.79	301-350	9.9
LATIMES	131,896	251.7	251.90	301-350	9.9
FBIS	130,471	249.9	554.42	351-400	7.9
FT91-93	138,668	221.8	196.44	401-450	6.5
OH90-91	148,162	81.4	64.02	1-63	4.96

frequency and normalisation). Along with the two benchmarks, this gives us nine solutions in total for each of the three problems. Tables 2, 3 and 4 show the terminal set and some GP parameters for each of the three weighting problems.

Table 2. Global Weighting Terminals

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, 1, 10	*the constants*, 0.5, 1 and 10
Parameters	7 runs of a Population of size 100 for 50 generations

Table 3. Term-Frequency Weighting Terminals

Terminal	Description
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

Table 4. Normalisation Weighting Terminals

Terminal	Description
l	document length (unique terms)
l_{avg}	average document length (unique terms)
l_{dev}	standard deviation of document 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 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

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 [12]. The second scheme is the idf_{rsj} as found in the BM25 scheme [12].

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

Table 5 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.

Table 5. % 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

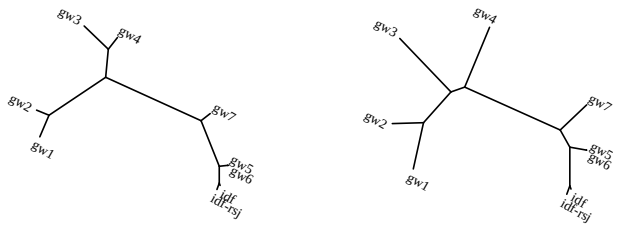

Fig. 2. Trees for global weightings using *dist* and *w_dist* respectively

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 means that the difference in terms of MAP could be as high as 4.15% MAP. Figure 2 seems to indicate that the solutions are evolving towards ranked lists 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 (i.e. ranked list). On the test collections (Table 6), we can see that the differences in MAP between the evolved weightings and both types of *idf* (which return very similar ranked lists) are all statistically significant ($p < 0.05$) using a two-tailed t-test.

Table 6. % MAP for *idf* and global weightings on test data

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

On the test data, we can see that gw_1 to gw_4 perform similarly. gw_5 and gw_6 still perform slightly better than idf_{rsj}, while gw_7 still performs slightly better than these again. As gw_2 is very close to the our best solution (gw_1) and is elegant in form, we choose it to form part of the entire weighting scheme.

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

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).

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

Table 7 shows the MAP of the seven term-influence weighting function evolved on the training collection. It is important to note that all our evolved functions are used in conjunction with gw_2.

Table 7. % MAP for benchmarks and $gw_2.tf_i$ 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

Firstly, we can see that $Piv_{s=0}$ performs poorly compared to $BM25_{b=0}$. Also, we can see that 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. When we look at the differences of the ranked lists produced, 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 best evolved weighting scheme is quite large at this stage. Taking the best term-frequency factor from the training set (tf_6), we can see that it can be simplified to show that it is basically a flattened form of the term-frequency part of the $BM25_{b=0}$ weighting. As an example of the distances between solutions in the trees in Figure 3, $dist(BM25_{b=0}, tf_6)$ is 84.41 and $w_dist(BM25_{b=0}, tf_6)$ is 5.58%.

$$tf_6 = log(\frac{10}{\sqrt{(0.5/tf) + 0.5}}) = log(\sqrt{\frac{200.tf}{1 + tf}}) \tag{3}$$

As all of the best term-frequency schemes are very similar in behaviour and MAP on both the training and test data (Table 8), we choose the best performing scheme on the training data (tf_6). Our term-weighting formula now consists of $gw_2.tf_6$.

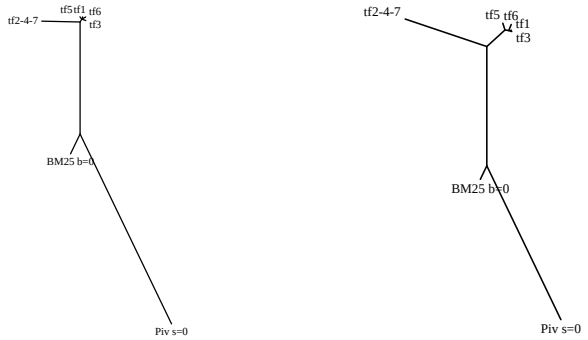

Fig. 3. Trees for term-frequency weightings using *dist* and *w_dist*

Table 8. % MAP for benchmarks and $gw_2.tf_i$ on test data

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 $\approx$	213 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. We compare our full evolved weighting scheme against the full Pivoted Document Normalisation and BM25 schemes [12].

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

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. As the term-frequency is normalised similar to the BM25 model, we evolve the normalisation factor n_i in the following formula:

$$gw_2.n_i tf_6 = gw_2.log(\sqrt{\frac{200.\frac{tf}{n_i}}{1 + \frac{tf}{n_i}}}) \tag{4}$$

As a naming convention, we will call the complete scheme $gw_2.n_i tf_6$ where n_i is the normalisation factor chosen. Table 9 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 Figure 4 we can see that n_6 is one of its closest neighbours.

From the test data (Table 10) we can see that n_4 and n_6 perform similarly well and are both significantly better than the best benchmark ($BM25$). Although, n_7 has a similar MAP on the training set we can see that it lies nearer some of the poorer forming solutions. n_7 performs only slightly better than n_4 on two of the collections as it has different features. We can see from both trees that n_2 is phenotypically close

Table 9. % MAP for benchmarks and $gw_2.n_itf6$ 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

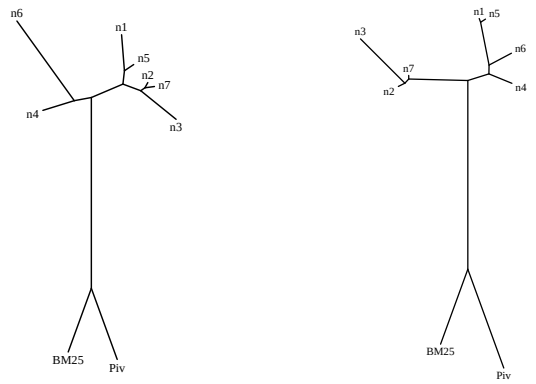

Fig. 4. Trees for normalisation weightings using $dist$ and w_dist

to n_7. These two schemes are also comparable in terms of MAP on the test data. On the test data, we can see that n_7's increase over BM25 is not statistically significant ($p < 0.05$). As an example of the distances between solutions in the trees in Figure 4 $dist(BM25, n_4)$ is 71.42 and $w_dist(BM25, n_4)$ is 9.47%. The $p-$values for the t-tests also confirm much of these findings.

$$n_4 = \frac{l}{l_{avg}} \times \frac{qtl}{10} \quad n_6 = \sqrt{log(qtl)} \times log(qtl) \times \frac{l}{l_{avg}} \quad n_7 = \frac{tl}{tl_{dev} + \frac{l}{qtl}} \tag{5}$$

Table 10. % MAP for benchmarks and $gw_2.n_itf6$ on test data

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-$value \approx$	213 Topics	0.006	-	0.613	0.090	0.206	0.043	0.455	0.011	0.070

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. 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 an area of the solution space that is different current benchmarks schemes.

7 Acknowledgments

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

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*, pages 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*, pages 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*, pages 305–306, 2005.
5. Ronan Cummins and Colm O'Riordan. Evolving local and global weighting schemes in information retrieval. *Information Retrieval*, 9(3):311–330, June 2006.
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. Technical report, Computer Systems Laboratory, National Institute of Standards and Technology. Gaithersbury, M.D, 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)*, pages 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. Gerard Salton and Chris Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.
12. A. Singhal. Modern information retrieval: A brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 24(4):35–43, 2001.
13. Andrew Trotman. Learning to rank. *Information Retrieval*, 8:359 – 381, 2005.