(1+1)-Evolutionary Gradient Strategy to Evolve Global Term Weights in Information Retrieval

Osman Ali Sadek Ibrahim, Dario Landa-Silva

Abstract In many contexts of *Information Retrieval (IR)*, term weights play an important role in retrieving the relevant documents responding to users' queries. The term weight measures the importance or the information content of a keyword existing in the documents in the IR system. The term weight can be divided into two parts, the *Global Term Weight (GTW)* and the *Local Term Weight (LTW)*. The GTW is a value assigned to each index term to indicate the topic of the documents. It has the discrimination value of the term to discriminate between documents in the same collection. The LTW is a value that measures the contribution of the index term in the document. This paper proposes an approach, based on an evolutionary gradient strategy, for evolving the Global Term Weights (GTWs) of the collection and using *Term Frequency-Average Term Occurrence (TF-ATO)* as the Local Term Weights (LTWs). This approach reduces the problem size for the term weights evolution which reduces the computational time helping to achieve an improved IR effectiveness compared to other Evolutionary Computation (EC) approaches in the literature. The paper also investigates the limitation that the relevance judgment can have in this approach by conducting two sets of experiments, for partially and fully evolved GTWs. The proposed approach outperformed the Okapi BM25 and TF-ATO with DA weighting schemes methods in terms of Mean Average Precision (MAP), Average Precision (AP) and Normalized Discounted Cumulative Gain (NDCG).

Osman Ali Sadek Ibrahim
ASAP Research Group, School of Computer Science
The University of Nottingham.
CS Dept., Minia University, Al-Minya, Egypt.
e-mail: psxoi@nottingham.ac.uk

Dario Landa-Silva
ASAP Research Group, School of Computer Science
The University of Nottingham
e-mail: dario.landasilva@nottingham.ac.uk

1 Introduction

The effectiveness of an Information Retrieval (IR) system is measured by the quality of retrieving relevant documents responding to user information needs (queries). One of the common models used in IR is *Vector Space Model (VSM)*. Documents are represented in VSM as vectors of term weights. The term weight has a significant impact on the IR system effectiveness to retrieve relevant documents responding to user information needs. An IR system contains the document weight representations of the document collection in the form of an IR index file [32]. For every index term in an IR index file, a term weight measures the information content or the importance of the term in the document. This term weight has two parts: the local and the global weights. The *Local Term Weight (LTW)* measures the contribution of the term within a given document. The *Global Term Weight (GTW)* measures the discrimination value of the term to represent the topic of the documents in the collection. GTW also indicates the importance of the term as a good discriminator between documents. Figure 1 shows the term weights structure in the *Index File* in an IR system.

Term weights can be improved for achieving better IR effectiveness if the users can identify examples of the relevant documents that they require for their current search. These examples of relevant documents and their corresponding user queries are stored into the relevance judgment file of the document collection. The relevance judgment of the IR document collection contains the group of relevant documents identified by users and their corresponding user information needs (queries). Evolutionary Computation (EC) techniques have been used extensively to improve IR effectiveness using the relevance judgment feedback from IR systems [7, 8]. Some of that previous research does not consider the problem size and the computational time that are required in order to achieve an improvement in IR effectiveness. Another issue is the dynamic variation in the document collection in real IR systems that happens when documents are added to or removed from the collection.

The related work on the *Term-Weighting Problem* can be divided into two categories: 1) evolving collection-based Term-Weighting Schemes (TWS) and 2) evolving term weights. These approaches have limited success to be used in real IR systems due to several reasons as explained below, which gives the motivation for the work presented in this paper.

1. The TWS evolved by Genetic Programming (GP) rely on the relevance judgment [9, 7, 23] to check the quality of the proposed weighting function. These approaches have the following limitations:
   - The problem size of creating better collection-based weighting function using GP is large [9, 23, 12]. This is because the whole documents space

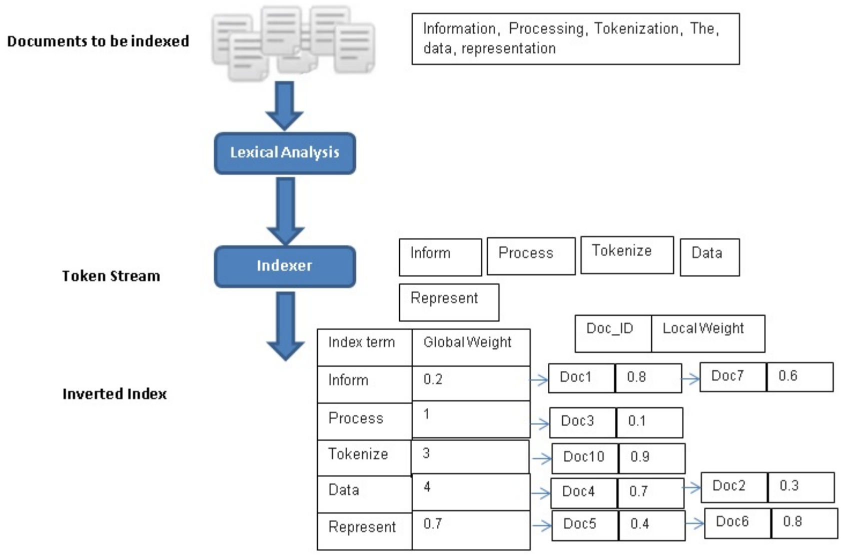

Fig. 1 The Construction of the Index File (also called post file) which serves as an index for the IR system. It contains the global and local term weights for every term in each document and the document and term identifiers with the local term weight for each term.

in the collection is considered in the evolving procedure of the global and local term-weighting functions.

- The computational run-time required to create better collection-based weighting functions using GP is high [9, 23, 12]. In [12], the computational run-time for evolving TWS in a subset of 20-Newsgroup collection [1] using GP was 18 hours. Moreover, other GP approaches [9, 23] applied on very small collections used a cluster of computers or took long computational run-time.

2. Evolving term weights of the document representations and evolving TWS using EC have resulted in better IR effectiveness regarding *Mean Average Precision (MAP)* and *Average Precision (AP)* [8, 7].

The main aim of this work is to propose a method to increase IR effectiveness by evolving better representations of documents in the collection for the trained queries with less computer memory usage. This is accomplished by evolving the Global Term Weights (GTWs) of the collection rather than evolving representations for the whole collection as is typically done with previous EC approaches in the literature. Hence, the main contribution of this paper is the development of a *(1+1)-Evolutionary Gradient Strategy ((1+1)-EGS)* with *Adaptive Ziggurat Random Gaussian Mutation* [18, 11, 21] to evolve GTWs. The proposed methodology reduces the problem size, from evolving $(N \times M)$ document representation vectors to evolving $(1 \times M)$ vector, where N is the number of documents in the collection and M is the number of index terms in the collection. This paper also also examines a new meta-heuristic method ((1+1)-EGS) in IR with a new methodology for evolving document representation. This method considers the limitation of the relevance judgment of the document collections in EC [16].

In order to evaluate the performance of the proposed method, experimental results are presented and discussed. The study compares results from using classical, fully evolved and partially evolved IR experiments. The proposed approach obtained improved MAP and improved AP compared to the Okapi BM25 and TF-ATO weighting schemes [25, 24, 16]. In addition, the ratio of AP improvement obtained is larger than the one from evolving global term weighting function approaches in some related work [9, 13, 7].

Section 2 reviews the related work, particularly on using EC for improving automatic indexing problem domain. Section 3 describes the proposed approach. Section 4 presents experimental setting and results while also discussing the key observations from the study. Finally, Section 5 presents some conclusions and outlines proposed future work.

2 Related Works

EC techniques have been widely used to improve effectiveness on various IR problems [7, 22, 20]. The objective functions used in such EC techniques usually rely on the relevance judgment to determine the quality of the candidate solutions being evolved. The following subsections outline the research that has been carried out on the document indexing problem [32] using EC techniques. The document indexing problem refers to assigning weights to each term in every document in the collection. This type of problem can be divided into: 1) evolving term-weighting schemes, and 2) evolving term weights.

2.1 Evolving term-weighting schemes

In this category, researchers have tried to evolve the best TWS for improving IR effectiveness using Genetic Programming (GP). However, these TWS can be considered as collection-based functions since each document collection has different characteristics. Furthermore, all the document collections are partially judged to simulate real document collections. Consequently, most of the index terms in a test collection do not exist in the training queries and their relevant documents [16].

The first approach to evolving a weighting function using GP was developed by Fan et. al. [13] using two document collections. One was the Cranfield collection containing 1,400 documents and 225 queries. The other one was the Federal Register (FR) text collection from TREC 4 containing a huge number of documents (55,554 documents) compared to its queries (50 queries). Fan et. al. argued that few documents are relevant for these queries so they choose a number of documents larger (2,200 documents) than the number of relevant documents as a training set. Fan et. al. used the precision based on collections relevance judgment with a threshold as a fitness function in their application. The evolved TWS with their GP approach was tested for the same trained queries on the whole document collections. Their results outperformed TF-IDF [17]. No results for the Cranfield collection have been produced with their approach [13]. Furthermore, the Federal Register (FR) text collection is a pooled collection and there is a question of how reliable it is to use the average precision and other effectiveness measurements on this collection [5, 29]. These limitations do not exist for evaluating the sampled judged document collections [5, 29].

Oren [23] proposed GP to evolve the term-weighting function using a terminal set similar to the one used by Fan et. al [13] but with an additional function operator (square root). Oren used the Cystic Fibrosis database [27] which consists of 1239 documents and 100 queries and compared his approach to the TF-IDF term-weighting scheme. Oren's method outperformed TF-IDF in respect of recall-precision values. In this case, a cluster of computers were used due to the problem size. Thus, the computational cost of Oren's approach even in the small collection used, was very high.

Cummins and O'Riordan [9] proposed a methodology for evolving local and global term-weighting schemes from small document collections. They showed that their global weighting function evolved on small collections also increased the average precision on larger document collections. However, their local weighting function evolved on small collections did not perform well on large collections. They conducted experiments on five document collections: Medline, Cranfield, CISI, NPL and Ohsumed. The computational run time required by their approach on the smallest training set from Medline collection was significant, 6 hours on a standard PC. Then, the main limitations of their approach are: 1) long computational time on medium and large document collections, 2) the issue of document collections being partially judged, and hence 3) poor performance on collections other than the training set.

2.2 Evolving term weights

Genetic Algorithms (GA) have been used for evolving term weights to produce better document representations for the whole document collections. These approaches are also based on the relevance judgment. The same drawbacks noted before also arise in these approaches: the reliance on partial relevance judgment for the collection and the need to run the GA again when changes happen in the collection (dynamic collections).

Gordon [14] proposed the first approach applying a GA to IR for adapting term weights for every document in the corpus. He demonstrated the value of using a GA for adapting terms weights instead of using probabilistic models. He also pointed at some issues when using probabilistic models: dependencies among index terms, dependency on the estimation of probabilities, relevance judgment based on a small set of queries and high computational cost of automated probabilistic models. The GA used a probability of crossover equal to 1 with no mutation and relevance feedback adaptation for the fitness function. He showed that the GA improved documents representation to distinguish between relevant and non-relevant queries. The problem size was very large, more than the document space as it consisted of multiple representations for each document.

Vrajitoru [31] also applied a GA to adapt term weights. The approach used a new dissociated crossover and tested different ways to generate the initial documents descriptions. Vrajitoru ran experiments using two document collections (CISI and CACM collections) larger than Gordon's collection [14]. This approach also has the limitation related to the relevance judgment due to the document collections nature.

3 The Proposed Approach

This section presents the proposed approach to evolve Global Term Weights (GTWs) in information retrieval from document collections. The method uses Term Frequency-Average Term Occurrence (TF-ATO) [16] and a (1+1)-Evolutionary Gradient Strategy (EGS) for this purpose. The general Evolutionary Gradient Algorithms was described in [3, 18]. To the best of our knowledge, this approach is the first one that focuses on evolving the GTWs vector instead of evolving term-weighting functions or evolving term weights for the whole document collection, as discussed in the introduction. Experiments conducted here show that this approach achieves better *MAP* and *AP* compared to the other methods in the literature [9, 7, 23, 13].

Table 1 The Notations Used in Algorithm 1.

Notation	Definition
RelDocSet	is the relevant document vector set of TF-ATO as the form of local term weight vectors.
IRRDocSet	is the irrelevant document vector set of TF-ATO as the local term weight representations.
QSet	is the query set of vectors in TF-ATO form.
ParentChromosomeGTW	is the current parent proposed of the evolved GTW vector chromosome for the index terms.
OffspringChromosomeGTW	is the current offspring of the evolved GTW vector of the index terms. This is the mutated (evolved) GTW parent chromosome (PG) of the current iteration.
ZGaussian(0,1)	is the Ziggurat random Gaussian number with 0 mean and 1 standard deviation and the value is between 0 and 1 [11].
MutatPos	is the position of the gene that will undergo mutation.
MutatPosGood	is the array that saved the previous position of the gene that had mutations in the previous iteration.
NoMutations	is the number that indicates the number of genes (GTWs) that will be mutated.
NoMutationsGood	is the saved number from the previous generation that indicates the number of genes (GTWs) that had mutations.
MaxGTW	is the maximum GTW which is 1 in our case with using TF-ATO as a local weighting scheme.
Random(t1,t2)	is a function used to generate random number between t1 and t2

An outline of the main steps in the method is given next. The first step is to obtain the corresponding vectors of local term weights for three sets of documents: the relevant document set, the irrelevant document set and their query set. These vectors contain TF-ATO values [16], of the index terms for every document in the three sets. Next, a (1+1)-EGS and Ziggurat random sampling [11] is used to mutate the gradient steps. This method was selected because it has been shown that compared to other evolutionary strategies methods, Ziggurat random sampling has lower cost in terms of memory space or computational run-time [21]. The aim of the (1+1)-EGS is to optimize the cosine similarity [4] between the relevant document vectors and the query vectors. At the same time, it aims to minimize the cosine similarity between the irrelevant document vectors and the query vectors. The evolved GTWs will then be assigned to index-terms in the document collection. These GTWs are multiplied by TF-ATO to produce term weight vectors for each document in the collection.

The pseudo-code of the (1+1)-EGS is shown in Algorithm 1 and Table 1 lists the notations used in the pseudo-code. Steps 1 to 6 include two methods to initialize the parent GTW chromosome. The first method gives higher

Algorithm 1: (1+1)-Evolutionary Gradient Strategy for Evolving GTWs

```
Data:
{RelDocSet:} is the Relevant Document Vector Set of TF-ATO weights.
{IRRDocSet:} is the Irrelevant Document Vector Set of TF-ATO weights.
{QSet:} is the Query Vector Set of TF-ATO weights.
{MaxGTW:} is equal 1 in case of using TF-ATO as a weighting scheme.
{M:} is equal to the number of index terms used to evolve their GTWs.
{Good:} has FALSE as an initialization value.
Result: Evolved GTWs of the Index Terms based on the relevance judgment values
Initialization for (IndexTerm $Term_i \in M$) do
    if ($Term_i$ is a good discriminator) then
        ParentChromosomeGTW[i] = MaxGTW + ZGaussian(0,1);
    else
        ParentChromosomeGTW[i] = ZGaussian(0,1);
    end
    OffspringChromosomeGTW[i] = ParentChromosomeGTW[i];
end
while CosineSimilarity(RelDocSet,QSet,ParentChromosomeGTW) $\leq$ Maximum do
    if (Good==TRUE) then
        NoMutations=NoMutationsGood;
    else
        NoMutations = Random(0,M);
        NoMutationsGood = NoMutations;
    end
    for i=1 $\rightarrow$ NoMutations do
        if (Good==TRUE) then
            MutatPos=MutatPosGood[i];
        else
            MutatPos = Random(0,M);
            MutatPosGood[i]=MutatPos;
        end
        OffspringChromosomeGTW[MutatPos]=OffspringChromosomeGTW[MutatPos]+
        (ParentChromosomeGTW[MutatPos] -
        OffspringChromosomeGTW[MutatPos]) * ZGaussian(0,1);
    end
    /* Keep the fitter evolved chromosome                                  */
    if (CosineSimilarity(RelDocSet,QSet,ParentChromosomeGTW)
    <CosineSimilarity(RelDocSet,QSet,OffspringChromosomeGTW)) AND
    (CosineSimilarity(IRRDocSet,QSet, ParentChromosomeGTW) >
    CosineSimilarity(IRRDocSet,QSet,OffspringChromosomeGTW)) then
        for i=1 $\rightarrow$ M do
            ParentChromosomeGTW[i] = OffspringChromosomeGTW[i];
            Good=TRUE;
        end
    else
        for i=1 $\rightarrow$ M do
            OffspringChromosomeGTW[i] = ParentChromosomeGTW[i];
            Good=FALSE ;
        end
    end
end
```

initialization values and is applied to index terms that are good discriminators. An index term is a good discriminator when: 1) it exists in irrelevant documents only or 2) it exists with higher TF-ATO value in relevant documents than in irrelevant document and this index term exists in the queries. The second method gives lower initialization values and is applied to index terms that are not good discriminators. Adding MaxGTW (a value of 1) to the initialization for good discriminators, instead of only a Ziggurat random number, reduces the convergence run-time. The initialized parent chromosome is then copied as the offspring chromosome in step 7. Then, the main evolution cycle of the (1+1)-EGS is described in steps 9-36. The stopping criterion of the algorithm (step 9) indicates that the evolution will stop when the maximum similarity (a value of 1 as given by the cosine function) between relevant documents and user queries is achieved. Steps 10 to 24 show the procedure to control the mutation within the (1+1)-EGS. As shown in step 23, the actual mutation operator uses the genes gradient multiplied by Ziggurat random Gaussian number with mean equal to 0 and standard deviation equal to 1 as the step-size. Steps 10 to 22 show the strategy to control the number of gradient mutations and the position in the chromosome to mutate. Note that this strategy repeats the mutation settings when the mutated offspring chromosome improves upon the parent chromosome (this is indicated by the Boolean variable *Good*). The objective function that examines the quality of the offspring solution is shown in step 25. This objective function contains two conditions. The first condition is to increase the cosine similarity value between the relevant document vector set and the query vector set. The second condition is to reduce the cosine similarity between the irrelevant document vector set and the query vector set. That is, the offspring GTW chromosome is selected as the parent chromosome (line 27) for the next iteration if it increases the discrimination between the relevant and irrelevant document vector sets with the query vector set. In this case, the variable *Good* is set to TRUE so that the mutation settings are repeated in the next iteration. Otherwise, the offspring GTW chromosome is replaced by the parent GTW chromosome (line 32), and the variable *Good* is set to FALSE.

As explained above, the initialization step in the above (1+1)-EGS distinguishes between index terms that are good discriminators and those that are not. This gives the proposed approach the ability to tackle *Polysemy*, one of the challenges in natural language. Polysemy happens when the same terms exists in both the relevant and the irrelevant document sets and the term has multiple different meanings in different contexts. Hence, Polysemy words are not good discriminators because they have high TF-ATO values (LTWs) in relevant and irrelevant documents. However, with the proposed approach Polysemy words get lower GTWs than the good discriminator terms, which emphasizes their non-discriminating nature.

4 Experimental Study and Evaluation

4.1 Document Collections

Nine document collections were used in these experiments [15, 10, 28, 30]. Table 2 shows their main characteristics. In these experiments, four combination groups from the document collections were used to produce four test collections. Each test collection combination contains three textual materials: a set of documents, a set of queries, and relevance judgments between documents and queries. For each query, a list of relevant documents is associated with it. The first test collection consists of Ohsumed, CISI and CACM document collections [15, 10], containing 353226 documents and 233 queries. The second test collection consists of Cranfield, Medline and NPL document collections [10], containing 13862 documents and 348 queries. These two test collections were formed from sampled collections and they have been widely used for research such as in [26, 7]. The third and fourth collection combinations are from three document collections in the TREC Disks 4 & 5 with two different query sets and their relevance judgments. Crowdsourced and robust relevance evaluation were used with the queries and relevance judgments [28, 30]. These third and fourth combinations contain FBIS, LA and FT document collections. The third test collection contains 472525 documents and 230 queries, while the fourth collection contains 18260 documents and 10 queries.

Table 2 Characteristics of the Document Collections Used in the Experiments.

ID	Description	No. of Docs	No. of Queries
Cranfield	Aeronautical engineering abstracts	1400	225
Ohsumed	Clinically-Oriented MEDLINE subset	348566	105
NPL	Electrical Engineering abstracts	11429	93
CACM	Computer Science ACM abstracts	3200	52
CISI	Information Science abstracts	1460	76
Medline	Biomedicine abstracts	1033	30
TREC Disks 4&5 (Robust 2004)	News and Broadcast WebPages	472525	230
TREC Disks 4&5 (Crowdsource 2012)	News and Broadcast WebPages	18260	10

4.2 IR System Evaluation

In this experimental study, *Mean Average Precision (MAP)*, *Average Precision (AP)* and *Normalized Discounted Cumulative Gain (NDCG)* were used [4, 19, 6]. Let $d_1, d_2, ..., d_{|D|}$ denote the sorted documents by decreasing order of their similarity measure function value, where $|D|$ represents the number of testing documents. The function $r(d_i)$ gives the relevance value of a document d_i. It returns 1 if d_i is relevant, and 0 otherwise. The AP per query or q ($AvgP(q)$) is defined as follows:

$$AvgP(q) \ = \ \frac{1}{|D|} \ \Sigma_{i=1}^{|D|} \ r(d_i) \ . \ \Sigma_{j=1}^{|D|} \ \frac{1}{j} \tag{1}$$

The MAP for a set of queries is the mean of the average precision values over all queries. This can be given by the following equation:

$$MAP \ = \ \frac{\Sigma_{q=1}^{Q} \ AvgP(q)}{Q} \tag{2}$$

where Q is the number of queries. The Normalized Discounted Cumulative Gain of top-k documents retrieved (NDCG@k) can be calculated by the following equation:

$$NDCG@k = \frac{1}{IDCG@k} * \Sigma_{i=1}^{k} \frac{2^{r(d_i)} - 1}{log_2(i+1)} \tag{3}$$

The *Discounted Cumulative Gain of top-k documents retrieved (NDCG (k))* can be calculated by the following equation:

$$DCG@k = \Sigma_{i=1}^{k} \frac{2^{r(d_i)} - 1}{log_2(i+1)} \tag{4}$$

where $IDCG@k$ is the ideal (maximum) discounted cumulative gain of top-k documents retrieved and $r(d_i)$ returns 1 if the document retrieved in position i is relevant and has 0 otherwise. If all top-k documents retrieved are relevant, the $DCG@k$ will be equal to $IDCG@k$.

4.3 Experimental Results

In this paper, two term-weighting schemes were used. The first weighting scheme was the BM25 Okapi probabilistic weighting scheme [25, 24]. This weighting scheme has a good capability for estimating the term weights based on probability theory [2]. The second weighting scheme was TF-ATO with the Discriminative Approach (DA) [17, 16], which is the only existing non-evolved approach that gives a good performance by discriminating documents without requiring any prior knowledge of the collection's relevance judgment. The number of index terms that were used in evolving their GTWs in the *Partially Evolved Experiment* in the test collections were 31658, 14679, 63091 and 6230 respectively. These terms are the keywords that exist in the relevant documents, the top-30 irrelevant documents using TF-ATO weighting scheme and their corresponding queries in the relevance judgment. In this experiment, the remaining non-evolved index terms in the document collections had values of 1s as GTWs. The number of index terms used in the *Fully Evolved Experiment* were 241450, 21600, 476850 and 18429 terms respectively. These terms constitute all the index terms in the collections.

Table 3 The NDCG@30 in the Four Collection Combinations of Using Okapi BM25, TF-ATO with DA and the Proposed Approach.

Normalized Discounted Cumulative Gain for top-30 Documents Retrieved				
DocID	BM25 Okapi	TF-ATO with DA	Fully Evolved	Partially Evolved
1st Collection Combination	0.451	0.525	0.663	0.695
2nd Collection Combination	0.515	0.57	0.733	0.754
3rd Collection Combination	0.558	0.608	0.768	0.778
4th Collection Combination	0.519	0.569	0.729	0.739

Table 4 The Mean Average Precision in the Four Collection Combinations of Using Okapi BM25, TF-ATO with DA and the Proposed Approach.

Mean Average Precision (MAP)				
DocID	BM25 Okapi	TF-ATO with DA	Fully Evolved	Partially Evolved
1st Collection Combination	0.29	0.364	0.4272	0.4779
2nd Collection Combination	0.345	0.4	0.4884	0.5157
3rd Collection Combination	0.3767	0.4243	0.5007	0.5245
4th Collection Combination	0.399	0.4512	0.5144	0.522

Tables 3 and 4 show the average results of 10 runs of the proposed approach. These results are focused in the *MAP* and the *Normalized Discounted Cumulative Gain (NDCG@30)* for the experimental study. The *Partially Evolved Experiment* and the *Fully Evolved Experiment* in general outperformed the Okapi BM25 and TF-ATO with DA approaches in terms of effectiveness. From table 3, the *(NDCG@30)* values of the *Partially Evolved Experiment* were 0.695, 0.754, 0.778 and 0.739 for the test collection combinations, while the *NDCG@30* values of the *Fully Evolved Experiment* were 0.663, 0.733, 0.768 and 0.729. The ratios of improvement in *NDCG@30* regarding Okapi BM25 in *Partially and Fully Evolved Experiments* were better than the improvement gained in evolving term-weighting functions in the literature [9, 13]. The ratios of improvement using the *Partially Evolved Experiments* with respect to Okapi BM25 were 54.1%, 46.41%, 39.43% and 42.39% respectively in the four collections, while the improvement ratios in the *Fully Evolved Experiments* were 47.01%, 42.33%, 37.63% and 40.46%. From table 4, the improvement ratios in the *MAP* values in the *Partially Evolved Experiments* were 64.8%, 49.5%, 39.24% and 30.83%, while the improvement ratios in the *MAP* values in the *Fully Evolved Experiments* were 47.31%, 41.57%, 32.92% and 28.92% respectively. Tables 6, 7, 8 and 9 in the Appendix, show the detailed results of the AP and MAP.

Table 5 The Average Computational Run-time per a Document in the Four Collection Combinations of Using Okapi BM25, TF-ATO with DA and the proposed Approach.

Average computational run-time in seconds per a document				
DocID	BM25 Okapi	TF-ATO with DA	Fully Evolved	Partially Evolved
1st Collection Combination	17	15	300	180
2nd Collection Combination	19	17	430	120
3rd Collection Combination	18	15	600	230
4th Collection Combination	17	15	260	75

From Table 5, the average computational run time for the *Partially Evolved Experiment* was less than for the *Fully Evolved Experiment* by 120 to 370 seconds depending on the number of evolved index terms in the GTWs vector. Thus, the *Partially Evolved Experiment* outperformed the *Fully Evolved Experiment* on computation time and also system effectiveness. The average running time of the *Partially Evolved Experiment* was between 75 seconds and 230 seconds in the smallest and largest collection combination, while the average computational time for the *Fully Evolved Experiment* was between 260 seconds and 600 seconds. In general, the TF-ATO with DA outperformed the other approaches in terms of computation time. However, the TF-ATO with DA weighting scheme had lower effectiveness values than the proposed approach. Thus, the next step in future research will be to reduce the computational time using a combined machine learning technique with (1+1)-EGS. These experiments were conducted on a 3.60 GHz Intel (R) core(TM) i7-3820 CPU and the implementation was in Java NetBeans under Windows 7 Enterprise Edition.

5 Conclusion and Future Work

This paper proposes an approach based on a (1+1)-Evolutionary Gradient Strategy and on Term Frequency-Average Term Occurrence (TF-ATO), for evolving the Global Term Weights (GTWs) of the document collection in Information Retrieval (IR). By using (1+1)-chromosomes of M genes, the proposed method is less demanding in terms of computer memory, compared to other evolutionary computation approaches for IR used in the literature. Other approaches in the literature use non-adaptive evolutionary computation techniques and have large search spaces for evolving document vectors. In contrast, the technique described here optimized the document vectors through a GTW vector using the local weight vectors of the collection. This approach also has positive impacts on improving IR effectiveness. In addition, the *Partially Evolved Experiment* considers the limitations of the relevance judgment and dynamic variation of the collection. The index terms that did not exist in the *Partially Evolved Experiment* had values of 1 for GTWs and TF-ATO for LTWs. The *Partially Evolved Experiment* was used to evolve the GTWs of the index terms existing in the relevant document set and top-30 irrelevant document set rather than all the index terms existing in the collection. The remaining documents that did not have relevance judgment values only had TF-ATO representations. The *Partially Evolved Experiment* outperformed the *Fully Evolved Experiment* in IR system effectiveness. In addition, the two experimental methods had better effectiveness than the Okapi and TF-ATO weighting schemes. On the other hand, the *Fully Evolved Experiment* consumed more computational time than the *Partially Evolved Experiment* for evolving GTWs for the queries existing in the collection's relevance judgment. The extension of this work will involve examining the proposed approach using city block function, distance function, MAP and NDCG@30 as objective functions. This extended work will investigate these objective functions for better performance. Moreover, combining machine learning techniques with (1+1)-EGS for evolving the GTWs is also an interesting future research direction towards achieving better IR effectiveness with less computational run-time than when using an (1+1)-Evolutionary Gradient Strategy.

References

1. 20 Newsgroups Document Collection: (Accessed (2015)), `http://qwone.com/~jason/20Newsgroups/`
2. Amati, G., Van Rijsbergen, C.J.: Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Transaction Information System 20(4), 357–389 (Oct 2002)
3. Arnold, D.V., Salomon, R.: Evolutionary gradient search revisited. IEEE Transactions on Evolutionary Computation 11(4), 480–495 (2007)

4. Baeza-Yates, R.A., Ribeiro-Neto, B.A.: Modern Information Retrieval - the concepts and technology behind search. Pearson Education Ltd., Harlow, England, 2nd edition edn. (2011)
5. Buckley, C., Dimmick, D., Soboroff, I., Voorhees, E.: Bias and the limits of pooling for large collections. Information Retrieval 10(6), 491–508 (2007)
6. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to rank using gradient descent. In: Proceedings of the 22Nd International Conference on Machine Learning. pp. 89–96. ICML '05, ACM, New York, NY, USA (2005)
7. Cordon, O., Herrera-Viedma, E., Lopez-Pujalte, C., Luque, M., Zarco, C.: A review on the application of evolutionary computation to information retrieval. International Journal of Approximate Reasoning 34, 241 – 264 (2003), soft Computing Applications to Intelligent Information Retrieval on the Internet
8. Cummins, R.: The Evolution and Analysis of Term-Weighting Schemes in Information Retrieval. Ph.D. thesis, National University of Ireland, Galway (May 2008)
9. Cummins, R., O'Riordan, C.: Evolving local and global weighting schemes in information retrieval. Information Retrieval 9(3), 311–330 (2006)
10. Document Collections From University Of Glasgow: (Accessed (2015)), `http://ir.dcs.gla.ac.uk/resources/test_collections/`
11. Doornik, J.A.: An improved ziggurat method to generate normal random samples (2005)
12. Escalante, H.J., Garcia-Limon, M.A., Morales-Reyes, A., Graff, M., y Gomez, M.M., Morales, E.F., Martinez-Carranza, J.: Term-weighting learning via genetic programming for text classification. Knowledge-Based Systems 83, 176 – 189 (2015)
13. Fan, W., Gordon, M.D., Pathak, P.: Personalization of search engine services for effective retrieval and knowledge management. In: Proceedings of the Twenty First International Conference on Information Systems. pp. 20–34. ICIS '00, Association for Information Systems, Atlanta, GA, USA (2000)
14. Gordon, M.: Probabilistic and genetic algorithms in document retrieval. Commun. ACM 31(10), 1208–1218 (Oct 1988)
15. Hersh, W., Buckley, C., Leone, T.J., Hickam, D.: Ohsumed: An interactive retrieval evaluation and new large test collection for research. In: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 192–201. SIGIR '94, Springer-Verlag New York, Inc., New York, NY, USA (1994)
16. Ibrahim, O., Landa-Silva, D.: Term frequency with average term occurrences for textual information retrieval. Soft Computing 20(8), 3045–3061 (2016)
17. Ibrahim, O.A.S., Landa-Silva, D.: A new weighting scheme and discriminative approach for information retrieval in static and dynamic document collections. In: Computational Intelligence (UKCI), 2014 14th UK Workshop on. pp. 1–8 (Sept 2014)
18. Kuo, R., Zulvia, F.E.: The gradient evolution algorithm. Information Sciences 316(C), 246–265 (Sep 2015)
19. Kwok, K.L.: Comparing representations in Chinese information retrieval. In: SIGIR '97 Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 34–41. ACM, New York, NY, USA (1997)
20. Liu, T.Y.: Learning to rank for information retrieval. Foundations and Trends in Information Retrieval 3(3), 225–331 (Mar 2009)
21. Loshchilov, I.: A computationally efficient limited memory cma-es for large scale optimization. In: Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation. pp. 397–404. GECCO '14, ACM, New York, NY, USA (2014)
22. MacFarlane, A., Tuson, A.: Local search: A guide for the information retrieval practitioner. Information Processing & Management 45(1), 159 – 174 (2009)
23. Oren, N.: Reexamining tf.idf based information retrieval with genetic programming. In: Proceedings of the 2002 Annual Research Conference of the South African Institute

of Computer Scientists and Information Technologists on Enablement Through Technology. pp. 224–234. SAICSIT '02, South African Institute for Computer Scientists and Information Technologists, Republic of South Africa (2002)

24. Pérez-Iglesias, J., Pérez-Agüera, J.R., Fresno, V., Feinstein, Y.Z.: Integrating the probabilistic models BM25/BM25F into lucene. CoRR abs/0911.5046 (2009)
25. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: Bm25 and beyond. Foundation Trends Information Retrieval 3(4), 333–389 (Apr 2009)
26. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. 34(1), 1–47 (Mar 2002)
27. Shaw, W.M., Wood, J.B., Wood, R.E., Tibbo, T.R.: The Cystic Fibrosis Database: Content and Research Opportunities. Library and Information Science Research 13(4), 347–366 (1991)
28. Smucker, M.D., Kazai, G., Lease, M.: Overview of the trec 2012 crowdsourcing track. Tech. rep., DTIC Document (2012)
29. Tonon, A., Demartini, G., Cudr-Mauroux, P.: Pooling-based continuous evaluation of information retrieval systems. Information Retrieval Journal 18(5), 445–472 (2015)
30. Voorhees, E.M.: Overview of the trec 2004 robust retrieval track (2004)
31. Vrajitoru, D.: Crossover improvement for the genetic algorithm in information retrieval. Information Processing & Management 34(4), 405 – 415 (1998)
32. Zobel, J., Moffat, A.: Inverted files for text search engines. ACM Comput. Surv. 38(2) (Jul 2006)

Appendix – Detailed results of the experimental study

Tables 6, 7, 8 and 9 show the *Average Precision (AP), Mean Average Precision (MAP)* of Okapi BM25, TF-ATO with DA and the proposed approach in fully and partially experiments.

Table 6 The improvement in MAP and AP on Okapi BM25, TF-ATO with DA, Partially Evolved and Fully Evolved Experiments of the first collection.

Recall	AP and MAP In The First Multi-topic Document Collection						
	BM25 Okapi	TF-ATO with DA	Fully Evolved Experiment	Partially Evolved Experiment	The ratio of improvement W.R.T. BM25 Okapi		
					DA Improvement (%)	Full Evolved Improvement (%)	Partially Evolved Improvement (%)
0.1	0.745	0.8161	0.8715	0.8908	9.54	16.98	19.57
0.2	0.504	0.6104	0.7360	0.8203	21.11	46.032	62.76
0.3	0.357	0.4724	0.5109	0.615	31.96	43.109	72.27
0.4	0.2358	0.3617	0.4204	0.4935	53.39	78.29	109.8
0.5	0.200	0.2883	0.3972	0.4095	44.15	98.6	104.75
0.6	0.155	0.2400	0.3082	0.3581	54.84	98.839	131.032
0.7	0.138	0.1989	0.2395	0.2975	44.13	73.55	115.58
0.8	0.135	0.1541	0.1904	0.2191	14.148	41.037	62.3
0.9	0.127	0.1301	0.171	0.197	-3.64	34.65	55.12
MAP	0.289	0.364	0.4272	0.4779	25.57	39.067	50.1

Table 7 The improvement in MAP and AP on Okapi BM25, TF-ATO with DA, Partially Evolved and Fully Evolved Experiments of the second collection.

Recall	AP and MAP In The Second Multi-topic Document Collection						
	Okapi BM25	TF-ATO with DA	Fully Evolved Experiment	Partially Evolved Experiment	The ratio of improvement W.R.T. Okapi BM25		
					DA Improvement (%)	Full Evolved Improvement (%)	Partially Evolved Improvement (%)
0.1	0.6195	0.765	0.857	0.8735	23.487	38.354	41.001
0.2	0.5087	0.6549	0.698	0.715	28.740	37.271	40.554
0.3	0.4785	0.5262	0.610	0.6569	9.969	27.544	37.283
0.4	0.3958	0.4084	0.575	0.5952	3.183	45.275	50.379
0.5	0.3475	0.3605	0.482	0.4957	3.741	38.705	42.647
0.6	0.2812	0.2925	0.3909	0.4283	4.018	39.011	52.312
0.7	0.2135	0.2255	0.3416	0.392	5.621	60.000	83.607
0.8	0.146	0.1923	0.2483	0.2805	31.712	70.068	92.123
0.9	0.1179	0.1727	0.1922	0.2046	46.480	63.020	73.537
MAP	0.3454	0.3998	0.4884	0.5157	17.44	46.583	57.049

Table 8 The improvement in MAP and AP on Okapi BM25, TF-ATO with DA, Partially Evolved and Fully Evolved Experiments of on TREC Disk 4&5 Robust 2004 relevance feedback [30].

Recall	AP and MAP In The Third Multi-topic Document Collection						
	Okapi BM25	TF-ATO with DA	Fully Evolved Experiment	Partially Evolved Experiment	The ratio of improvement W.R.T. Okapi BM25		
					DA Improvement (%)	Fully Evolved Improvement (%)	Partially Evolved Improvement (%)
0.1	0.61	0.729	0.898	0.907	19.51	47.21	48.69
0.2	0.59	0.66	0.82	0.85	11.86	38.98	44.07
0.3	0.53	0.55	0.58	0.6	3.77	9.43	13.21
0.4	0.43	0.49	0.53	0.54	13.95	23.26	25.58
0.5	0.38	0.41	0.43	0.46	7.89	13.16	21.05
0.6	0.32	0.33	0.4105	0.435	3.13	28.28	35.94
0.7	0.22	0.26	0.347	0.3912	18.18	57.73	77.82
0.8	0.17	0.2	0.273	0.296	17.65	60.59	74.12
0.9	0.14	0.19	0.218	0.2412	35.71	55.71	72.29
MAP	0.3767	0.4243	0.5007	0.5245	14.63	37.15	45.86

Table 9 The improvement in MAP and AP on Okapi BM25, TF-ATO with DA, Partially Evolved and Fully Evolved Experiments of on TREC Disk 4&5 crowdsource 2012 relevance feedback [28].

Recall	AP and MAP In The Fourth Multi-topic Document Collection						
	Okapi BM25	TF-ATO with DA	Fully Evolved Experiment	Partially Evolved Experiment	The ratio of improvement W.R.T. Okapi BM25		
					DA Improvement (%)	Fully Evolved Improvement (%)	Partially Evolved Improvement (%)
0.1	0.631	0.693	0.925	0.939	9.83	46.6	48.81
0.2	0.597	0.653	0.875	0.853	9.38	46.57	42.88
0.3	0.548	0.598	0.6203	0.638	9.12	13.19	16.42
0.4	0.463	0.569	0.5971	0.592	22.89	28.96	27.86
0.5	0.435	0.492	0.447	0.436	13.10	2.76	0.23
0.6	0.367	0.392	0.395	0.398	6.81	7.63	8.45
0.7	0.237	0.292	0.335	0.325	23.21	41.35	37.13
0.8	0.185	0.198	0.246	0.276	7.03	32.97	49.19
0.9	0.127	0.174	0.189	0.244	37.01	48.82	92.13
MAP	0.399	0.4512	0.5144	0.522	15.4	29.9	35.9