Evolving Co-occurrence Based Query Expansion Schemes in Information Retrieval Using Genetic Programming

Ronan Cummins and Colm O'Riordan

Dept. of Information Technology,
National University of Ireland,
Galway, Ireland.
ronan.cummin@nuigalway.ie,colmor@it.nuigalway.ie

Abstract. Global query expansion techniques have long been proposed as a solution to overcome the problem of term mismatch between a query and its relevant documents. This paper describes a method which automatically tackles the problems of how to find the best terms for the expansion of a particular query and secondly, how to weight these terms for use with the original query. Genetic Programming is used to evolve schemes for term selection using global (collection-wide) co-occurrence measures. The schemes evolved are also used to weight the term in the expanded query as they are a measure of the term's importance in relation to the query. As a result, the genetic program has to learn a suitable scheme for identifying the best correlates for the query concept and also a scheme that correctly weights these in relation to each other. These schemes are tested on standard test collections and show a significant increase in performance on the training data but only modest improvement on the collections that are not included in training.

1 Introduction

Information Retrieval (IR) is concerned with the automatic retrieval of all relevant documents given a user need (query). However, vocabulary differences between the user and the supplier of information have often lead to a difficulty in retrieving many documents. Query expansion techniques have long been proposed as a means of overcoming term mismatch between the user's vocabulary and the vocabulary of the documents in the collection. Query expansion techniques typically add a number of non-query terms to the original query based on some heuristics in order to improve the performance of the original query. Typically, there are two types of query expansion methods; Local (pseudo-relevance or blind feedback) and Global (automatic thesaurus construction) query expansion techniques. This paper is concerned with the latter. In automatic thesaurus construction, terms are added to the original query based on their co-occurrence frequencies with query terms throughout the entire collection.

Recently there have been more and more attempts applying machine learning techniques to the domain of IR. Genetic Programming (GP) has been adopted by some researchers as it has certain advantages over other machine learning techniques. In particular, GP outputs a symbolic representation of a solution which can be used in further analysis. As a result, GP solutions are often quite general and are particularly suited to such problems. Developed in the early 1990's, the GP area [1] has grown and helped to solve problems in a variety of domains. GP is inspired by Darwinian theory of natural selection, where individuals that have a higher fitness value will survive and produce offspring. GP can be viewed as an artificial way of selective breeding.

This paper presents a Genetic Programming framework that artificially breeds query expansion selection schemes for use in a standard vector space framework. The next section introduces some background material in both query expansion and GP. Section Three describes the system and experimental design. Results and analysis are discussed in detail in section Four. Finally, our conclusions are presented in section Five.

2 Background

2.1 Global Query Expansion approaches

Global query expansion approaches analyse the entire document collection and use co-occurrence relationships between terms to build a matrix of term-term relationships. Usually, term-term matrices of this type contain weights which are a measure of how synonymous one term is with another. These matrices are large and computationally expensive to compute. The matrices are used to cluster terms based on their co-occurrence data in the hope that terms that are closer together in this term-space are synonymous. Conceptually, the role of documents and terms are interchanged in the retrieval model. In essence, documents become the features of the term. Thus, two terms that appear in the same document are indexed by a similar feature and are deemed to have some type of synonymous relationship. Many formulas have been proposed to measure the association between two terms using co-occurrence data. The similarity between two terms t_i and t_j can be determined by evaluating the difference between the two-vectors $\overrightarrow{t_i} = (d_{i1}, d_{i2}, ..., d_{in})$ and $\overrightarrow{t_j} = (d_{j1}, d_{j2}, ..., d_{jn})$ in the document vector space. A simple binary weighting on these document weights would lead to the following cosine formulation of similarity between two terms:

$$cos(t_i, t_j) = \frac{df(t_i, t_j)}{\sqrt{df(t_i) df(t_j)}} \tag{1}$$

where $df(t_i, t_j)$ is the number of documents in which both t_i and t_j occur and $df(t_i)$ is the number of documents in which t_i occurs. There are many variations of such formulas which aim to accurately find the best synonyms for a term. Many approaches have attempted to add the best synonyms for each individual term in the query to the original query. Many of these approaches have seen relatively little or no improvement in the retrieval of relevant information over the original query [2, 3].

However, independently analysing each query term ignores the concept of the query. Thus, terms that are selected for expansion based on this type of method typically have no context related to them (i.e. a term maybe closely related to one of the query terms but may not be related to the concept of the entire query). A concept based approach to query expansion has previously been attempted by promoting terms similar to the entire query by summing the individual associations for each term in the query [2]. Thus, terms that would be chosen for expansion would be similar to many terms in the query and thus have a concept associated with them. In this way the problem of term independence of query terms is somewhat overcome. This approach was somewhat successful on certain collections although the baseline weighting scheme used to weight the original terms in the query was poor as shown by the results on the NPL collection [2, 4]. Other attempts at creating collection dependent automatic thesauri by limiting the co-occurrence of terms to certain parts of text (e.g. paragraphs, sentences and phrases) have shown to be somewhat effective [5, 4].

Once terms have been chosen to be added to the original query by some expansion algorithm, the weight of the terms to be added must be determined. Term selection and re-weighting are the two main challenges that face global thesaurus techniques.

2.2 Standard term-weighting approaches

The BM25 weighting scheme, developed by Robertson et al. ([6]), is a weighting scheme based on the probabilistic model. The weight assigned to a term in the BM25 scheme is a product of Okapi-*tf* and *idf*. Okapi-*tf* is calculated as follows:

$$Okapi\text{-}tf = \frac{rtf}{rtf + k_1((1-b) + b\frac{dl}{dl_{avg}})} \tag{2}$$

where rtf is the raw term frequency and dl and dl_{avg} are the length and average length of the documents respectively. k_1 and b are tuning parameters. The *idf* of a term as determined in the BM25 formula is as follows:

$$idf_t = log(\frac{N - df_t + 0.5}{df_t + 0.5}) \tag{3}$$

where N is the number of documents in the document set and df_t is the document frequency of term t. The score for the document d can be calculated as followed:

$$BM25(Q,d) = \sum_{t \in Q \cap d} (Okapi\text{-}tf \times idf_t \times qrtf_t) \tag{4}$$

where $qrtf_t$ is the raw term frequency of t in the query Q. Thus, $BM25(Q,d)$ is a measure of the similarity between the document d and the query Q.

2.3 Genetic Programming

GP is a heuristic stochastic searching method that is efficient for navigating large, complex search spaces. The advantage of this evolutionary approach is that it can help to solve problems in which the roles of variables are not correctly understood. GP is often used to automatically derive functions whose variables combine and react in complex ways.

Initially, a random population of solutions is created. The solutions are modelled as tree-like structures with operators as internal nodes (functions) and operands as leaf nodes (terminals). These nodes are often referred to as genes and their values as alleles. Each solution is rated based on how it performs in its environment. This is achieved using a fitness function. Once this is done, reproduction can occur. Solutions with a higher fitness will produce more offspring. Goldberg uses the roulette wheel example where each solution is represented by a segment on a roulette wheel proportionately equal to the fitness of the solution [7]. Reproduction (recombination) can occur in variety of ways. The most common form is sexual reproduction where two different individuals (parents) are selected and two separate children are created by combining the genotypes of both parents. The coded version of a solution is called its genotype, as it can be thought of as the genome of the individual, while the solution in its environment is called its phenotype. The fitness is evaluated on the phenotype of a candidate solution while reproduction and crossover is performed on the genotype. Once the recombination process is complete each individual's fitness in the new generation is evaluated and the selection process starts again. The algorithm usually ends when a certain number of generations have been completed, when convergence of the population has been detected or when an individual is found with an acceptable fitness.

3 Design and Experimental Setup

3.1 Term-Selection

The GP approach adopted evolves the scheme used to select and weight terms for use in the expanded query in order to improve the retrieval performance of the system. For each query expansion scheme, each term in the corpus is rated based on how close it is to the query concept. The following formula shows the similarity (or Term Selection Value) between the term t and the entire query Q:

$$TSV(Q,t) = \sum_{q \in Q} (correlation_{qt} \times qrtf_q) \tag{5}$$

where Q is the query, q is a query term, t is a non-query term in the corpus, $qrtf_q$ is the raw term-frequency of term q in the query and $correlation_{qt}$ is the query expansion scheme to be evolved. As a result, $correlation_{qt}$ is a measure of the degree to which term t and the query term q are related by co-occurrence measures. By extension, $TSV(Q,t)$ represents the similarity between the entire query Q and a non-query term t. For each query Q, a number of top terms are chosen and added to the query vector. The number of terms added to the query can easily be increased without any change to the formula as terms further down the ranked list should have less significance in the expanded query as they are weighted as a function of their $TSV(Q,t)$ value.

3.2 Term Re-Weighting

We assume that the weight of an expanded term is a function of $TSV(Q,t)$ (i.e. the similarity of that term to the query). It is also logical to assume that the weight of the expansion term is also related to the weighting scheme applied to the original query terms (i.e. a *tf-idf* type scheme). Thus, the following formula is how our system scores the complete expanded query (EQ) in relation to a document d:

$$sim(EQ,d) = BM25(Q,d) + \sum_{t \in E} TSV(Q,t) \times Okapi\text{-}tf \times idf_t \qquad (6)$$

where EQ is the expanded query, Q is the original query, E is the set of expansion terms. Thus, a weighting of 1 for $TSV(Q,t)$ would indicate that the expansion term t is as important as if it had occurred in the original query. In this way the GP can also learn the correct weighting for expansion terms.

3.3 Document collections and preprocessing

The document collections used in this research are the Medline, CISI, Cranfield, NPL, LISA and OHSUMED collections[1]. Only the first 30 queries for the CISI and Cranfield collections are used as efficiency is of prime concern. The largest collection (OHSU88) is a subset of document of the full OHSUMED collection. It consists of half of the documents from the 1988 collection. All documents and queries are pre-processed by removing standard stop-words and stemmed using Porter's stemming algorithm [8]. The weighting scheme applied to the query terms is a relative term frequency weighting scheme. All queries with no relevant documents are ignored by the system.

Global query expansion techniques are computationally intensive. We reduce the number of terms in the collection by using a feature selection technique which reduces the number of the terms in each corpus to roughly 25%. We eliminate all dilute terms (i.e. terms whose document frequency equals its collection frequency). This has been shown to be a characteristic of evolved weighting schemes on both small and large collections [9]. Typically, this eliminates terms of a low frequency and variations have been used in other feature extraction techniques like document frequency thresholding. Table 1 shows the characteristics of the document collections after preprocessing and feature selection.

3.4 Terminal and Function set

To determine the terminal and function set, it is neccessary to consider the characteristics of the documents in which the query terms and possible expansion terms co-occur in the entire corpus. It is also important to consider the characteristics of each query term and each possible expansion term independently in the entire corpus. Table 2 shows the terminal set chosen. This set is divided into

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

Table 1. Characteristics of document collections

Collection	Docs	Terms	Reduced Terms	Avg Len	Qrys	Avg len
Medline	1,033	10,975	3,614	56.8	30	11
Cranfield	1,400	9,014	2,518	59.6	30	8.3
CISI	1,460	8,342	2,110	47.8	30	7.56
LISA	6,004	16,168	4,411	36.3	35	6.78
NPL	11,429	7,759	2,468	18.78	93	6.78
OSHU88	35,412	113,145	31,496	48.03	61	5.05

two parts in order to draw attention to the source of information for the chosen terminals. The top half of the table shows collection-wide statistics for both the query term q and the possible expansion term t independently of each other. The bottom part of the terminal set shows measures of the set of documents in which both the query term q and possible expansion term t co-occur. We will define the set of documents in which both t and q occur as C_{qt}.

Table 2. Terminal Set

Terminal	Description
1	*the constant* 1
0.5	*the constant* 0.5
cf_q	frequency of a query term (q) in the collection
cf_t	frequency of a non-query term (t) in the collection
df_q	no. of documents a query term (q) appears in
df_t	no. of documents a non-query term (t) appears in
N	no. of documents in a collection
S	no. of words in the collection
$\|Q\|$	no. of terms in the query Q
bin_{qt}	no. of documents in C_{qt}
$prod_{qt}$	sum of the product of the term-frequencies in C_{qt}
min_{qt}	sum of the minimum of the term-frequencies in C_{qt}
sum_{qt}	sum of the sum of the term-frequencies in C_{qt}
cof_q	sum of the term-frequencies for q in C_{qt}
cof_t	sum of the term-frequencies for t in C_{qt}
W_{qt}	total no. of words in C_{qt}

For the set of documents in which two terms co-occur, we combine the within-document (local) measures for those terms in an intiutive manner. For example the $prod_{qt}$ measure is often used to measure the correlation between two terms and is calculated as follows:

$$prod_{qt} = \sum_{d \in N} (tf_{dq} \times tf_{dt}) \tag{7}$$

where tf_{dt} is the term-frequency of t in document d. The bin_{qt} measure is calculated similary assuming a binary weighting on the within-document term-frequencies. The min_{qt} measure is the sum of the intersection (or minimum) of the term-frequencies.

Table 3. Function Set

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

3.5 Fitness Function

The mean average precision (MAP), used as the fitness function, is calculated for each scheme by comparing the ranked list returned by the system for each query expansion scheme against the human determined relevant documents for each query. Mean average precision is calculated over all points of recall and is frequently used as a performance measure in IR systems as it provides a measure of both the accuracy and recall of the retrieval system.

3.6 GP Parameters

All experiments are run for 70 generations with an initial population of 2000. Populations of less than 500 for this problem converge prematurely as the terminal set is quite large. Experimental analysis shows us that the population converges before 70 generations when using the largest terminal and function set. The solutions are trained on an entire collection and query set. They are then tested for generality on the collections that were not included in training. Trees are limited to a depth of 10. The aim is to discover general natural language characteristics for query expansion that will aid retrieval performance. We evolve these term-selection schemes by adding the top 8 terms to the original query.

4 Results and Analysis

4.1 Evolved Term Selection Schemes

We evolved solutions on the three smaller collections. The best solution for each collection was chosen for evaluation on previously unseen data. These solution will be refered to as the Medline, CISI and Cranfield solutions for the remainer of this paper. The CISI solution (8) and Cranfield solution (9) evolved are shown as an example of the solutions found.

$$correlation_{qt} = \frac{min_{qt}}{(rdf_t \times log(cf_t)) + \sqrt{sum_{qt} \times cf_q}} \tag{8}$$

$$correlation_{qt} = \frac{scf_q + bin_{qt}}{(((((bin_{qt}/W_{qt}) \times (0.5 + rdf_t)) \times N) + scf_q)/(log(prod_{qt}))) + W_{qt}} \tag{9}$$

Table 4 shows the MAP for the original query and the expanded queries on all the collections included in this research. From an evaluation perspective the most important collections are those which are not included in training.

Table 4. MAP for expanded queries using best evolved solutions

				Solution		
Collection	Docs	Qrys	BM25	CISI	Medline	Cranfield
CISI	1,460	30	19.51%	23.37%	20.18%	20.40%
Medline	1,033	30	53.51%	55.63%	65.37%	56.50%
Cranfield	1,400	30	38.43%	36.60%	39.73%	41.61%
LISA	6,004	35	35.01%	36.43%	35.37%	34.25%
NPL	11,429	93	28.75%	29.01%	28.74%	28.77%
OHSU88	35,412	63	23.25%	24.33%	21.78%	23.43%

Firstly we can see that there is a significant increase in MAP on the Medline collection when using the solution specific to that collection. This confirms previous concept-based approaches [2] which also show a similar increase on this collection. However, this Medline solution seems to be specific to the collections as there is no substantial improvement on other collections. The CISI and Cranfield solutions do not achieve as high an average precision on the Medline collection as the Medline solution does on its own training data. The solution found on the CISI collection is the only solution that increases average precision on all three larger collections.

4.2 Increasing Terms added to Queries

To determine whether adding only 8 terms to each query is sufficient to learn a general term clustering algorithm, we calculated the MAP for the evolved formulas for expanded queries of various lengths. This is investigated as previous research recommends expanding the query by up to 100 terms [2]. This amount of expansion is computationally very expensive and rather unrealistic in a real IR setting. Table 5 shows the MAP of the evolved formulas tested on their training data adding a varying number of terms. We see that, in general, adding more terms does not significantly increase or decrease the MAP of the queries.

Table 5. MAP for different length expanded queries

				Terms Added to each query						
Collection	Docs	Qrys	BM25	Top 8	Top 16	Top 24	Top 32	Top 40	Top 48	Top 96
CISI	1,460	30	19.51%	23.37%	23.68%	23.77%	23.35%	23.43%	23.41%	22.40%
Medline	1,033	30	53.51%	65.37%	65.62%	66.36%	66.46%	66.38%	66.32%	66.30%
Cranfield	1,400	30	38.43%	41.61%	41.56%	41.17%	40.79%	40.62%	40.96%	38.23%

Table 5 indicates that the schemes learned for the respective collections correctly weight the quality of each expanded term. As an example, we take two queries from the Medline collection and look at the weights assigned to the top 8 most similar terms according to the Medline solution. The 21^{st} Medline query which is stemmed to the following stems:

```
Medline Query 21: {languag develop infanc pre-school ag}
```

and has its 8 most similar terms, according to the Medline solution, shown in Table 6. Similarly, the 23^{rd} query which is preprocessed to the following:

```
Medline Query 23: {infantil autism}
```

is also shown in Table 6.

Table 6. Scores for Expansion terms for two sample queries

Query 21		Query 23	
Terms	TSV Score	Terms	TSV Score
deaf	0.891525	autist	2.12915
children	0.659627	mental	1.17509
learn	0.645482	child	0.733794
speech	0.497686	children	0.606834
word	0.356095	schizophrenia	0.569791
impair	0.323454	contact	0.407135
spoken	0.314593	symptom	0.324127
teach	0.302923	situat	0.27934

From these tables we can see that the evolved schemes can promote terms which seem to be related to the query concept and provides a weighting which is related to the quality of the expansion term. It can also promote different forms of query terms that Porter's stemming algorithm has failed to conflate. However, although solutions can be evolved that correctly find good expansion terms for a query, these solutions seem to be domain specific.

5 Conclusion

The results of this approach seem to confirm many previous approaches in that global co-occurrence data is unlikely to bring about a substantial general increase in the performance of IR systems [3]. However, we have learned domain specific formulas for finding good expansion terms. Importantly, the approach adopted also learns a mechanism for weighting these terms in relation to the original query without having to develop the weighting of such expansion terms analytically.

References

1. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, MA, USA (1992)
2. Qiu, Y., Frei, H.P.: Concept-based query expansion. In: Proceedings of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval, Pittsburgh, US (1993) 160–169
3. Peat, H.J., Willett, P.: The limitations of term co-occurrence data for query expansion in document retrieval systems. JASIS **42** (1991) 378–383
4. Jing, Y., Croft, W.B.: An association thesaurus for information retrieval. Technical report, University of Massachusetts, Amherst, MA, USA, Amherst, MA, USA (1994)
5. Xu, J., Croft, W.B.: Query expansion using local and global document analysis. In: SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (1996) 4–11
6. Robertson, S.E., Walker, S., Hancock-Beaulieu, M., Gull, A., Lau, M.: Okapi at TREC-3. In: In D. K. Harman, editor, The Third Text REtrieval Conference (TREC-3) NIST. (1995)
7. Goldberg, D.E.: Genetic Algorithms in Search, Optimisation and Machine learning. Addison-Wesley (1989)
8. Porter, M.: An algorithm for suffix stripping. Program **14** (1980) 130–137
9. Cummins, R., O'Riordan, C.: Determining general term weighting schemes for the vector space model of information retrieval using genetic programming. In: 15th Artificial Intelligence and Cognitive Science Conference (AICS 2004). (2004)