Evolution and Evaluation of Document Retrieval Queries

Robert Steele, David Powers
Department of Computer Science
The Flinders University of South Australia

rsteele@cs.flinders.edu.au, powers@acm.org

1. Introduction

This short paper introduces some ideas for the application of Genetic Programming to the task of producing queries for the accurate retrieval of documents on a particular topic of interest. This application will involve expanding the initial query given by the user, using extra words that have various semantic relationships to the words of the original query via the use of a system such as WordNet, and using Genetic Programming to optimize how this expansion is done. The aim is to produce an optimal general template of expansion, that can be used for any particular query a user may make. The method suggested is one that will be used to produce a search enhancer that can be used with existing search engines.

1.1. Definitions

Firstly, Genetic Programming (Koza, 1992; Koza 1994) is a way of evolving a program that meets some objective criteria, and is closely related to Genetic Algorithms (Goldberg, 1989). With Genetic Programming, the "programs" are represented as expression trees consisting of operators at internal nodes of the tree eg. plus, minus, and values, at the leaf (terminal) nodes of the tree. The Genetic Programming algorithm involves making a population of such expression trees (randomly at first), evaluating the fitness of each of the trees (done with some evaluation function defined by the application), and then creating a new population by crossover or mutation of the fit individuals (crossover would involve choosing a random node and the subtree which starts with it, from tree A, and choosing a random node and the subtree which starts from it, from tree B, and interchanging these two subtrees, to create two new tree expressions).

Query expansion refers to adding more terms to the original query. For example, if a search engine user made the query 'tree and paper' the query might expand to '(tree or forest) and paper'.

A semantic net, such as WordNet (Miller et al., 1993), is a database of words and their various relationships. So WordNet , given a particular word, will be able to tell you such things as its synonyms, antonyms, hyponyms and meronyms amongst many other possibilities. This will give the building blocks for query expansion in this application.

2. Aims and Discussion

Our basic aim is to produce a search engine enhancement that is practical in an application such as searching the World Wide Web. This manifests itself in two ways.

First, the application requires no change to the underlying search engine. It uses the search operators of the underlying search engine, and simply extends queries so that they will lead to higher precision in the returned documents

Second, the expression trees must be generic and should be applicable to any particular user query. This contrasts with the use of Genetic Programming to refine a particular oft-used query (Kraft, Petry, Buckles & Sadasivan, 1994) rather than produce a general template of expansion.

The choice of Genetic Programming as the method for the automatic refinement of the query expansion is supported in two ways: other learning methods, select their next search node based on a single promising node. With Genetic Programming the permutation makes use of a subcomponent (subtree) of a second promising node. This should be beneficial in this case, where some subtrees in the population of expression trees will have a high individual fitness.

Moreover, in this application, the points in the search space are in fact programs (mappings from the user query to a useful query expansion) and so this is well suited to the assumed 'program nature' of the objects in the search space that Genetic Programming investigates.

3. Methodology

The first point to note is that the optimized query expansion method will be evolved in a development phase, prior to everyday use of the search enhancement.

Robert Steele and David Powers (1998) **Evolution and Evaluation of Document Retrieval Queries.** In D.M.W. Powers (ed.) *NeMLaP3/CoNLL98: New Methods in Language Processing and Computational Natural Language Learning,* ACL, pp 163-164.

In constructing a Genetic Programming system, there are three basic variables that need to be defined.

Firstly, the internal nodes for the application. Here they will be either 'and', 'or', or 'not'. The reason for this choice, is that these are the operators already commonly available in search engines.

Secondly, the leaf nodes must be chosen. Here they will be the words of the original query, and various related words produced by WordNet. The important feature of these, is that they will not be fixed words, but rather of the form A, synonym(A) or hyponym(A) for example (where A is an original search term), and it is this that allows the evolved expression trees to be applicable to any search that may be made.

Thirdly, the fitness evaluation function is required. Fitness in this case is determined by the relevance of the documents returned by a query. A number of possible measures exist (Harter, 1996);

- frequency of original search word in document,
- nearness of multiple search words in the full document,
- correct relative frequencies of the words desired,
- cluster signatures can be used to indicate if the retrieved documents are similar to each other. Greater homogeneity is better.
- location and frequency of various related words, suggested by WordNet in the full document.

The evaluation function will weigh up all the data that can be extracted from the full returned documents, and weight according to which is deemed the best indicator of relevance.

4. Implementation

A problem with development of the system is that it will require the retrieval of many documents. For this reason it is best to develop it off-line. The TIPSTER CD used at the TREC conferences, represents a good benchmark. To make use of this, we will create a basic indexing system, similar to those of existing search engines.

This will involve creating a file for each word that occurs in the database (excepting very frequent words, and possibly words that do not occur in any document more times than some threshold number), and storing in the file, a reference to each document the file occurs in, the corresponding frequency and its first location in the document.

This simulated search engine will order the importance of documents with the following rules:

- Rf(A) = freq(A)/ freq(most frequent word)
- Rf(A and B) = sqrt(Rf(A)*Rf(B))
- Rf(A or B) = (Rf(A) + Rf(B))/2
- Rl(A) = (document length - first occurrence)/ document length
- Rl(A and B) = sqrt(Rl(A)*Rl(B))
- Rl(A or B) = (Rl(A) + Rl(B))/2
- R = Rf + bRl

Where Rf is the relevance based on frequency, Rl is the relevance based on location, b is some weight and R is the overall relevance value.

5. Conclusion

This abstract describes the starting point of a project we are undertaking on the evolution of useful rules for search enhancement. There are a number of probable advantages to using the approach given above, to produce a search enhancer. Firstly, the use of Genetic Programming allows the discovery of optimal search expressions that would not necessarily be intuitively chosen. Secondly, the query expansion methods produced will be general, so that they can be utilized in any particular query made, and thereby much increases the general usefulness of the system. Thirdly, the system does not assume improvements in the underlying search engine technology and so is more easily applied.

Although the present query expansion method is static - that is, evolution stops on delivery due to the cost of on-line evolution, we are also developing low cost ideas for continuing the evolution using cache information, pre-fetch and post-fetch, thereby allowing dynamic user profiling.

6. References

Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley.

Harter, S.P. (1996). Variations in relevance assessments and the measurement of retrieval effectiveness. *Journal of the American Society for Information Science* vol.47, no. 1, p. 37-49.

Koza, J. (1992). Genetic Programming, On the Programming of Computers by Means of Natural Selection. MIT Press.

Koza, J.R. (1994). Genetic Programming 2: Automatic Discovery of Reusable Programs. Bradford Book, MIT Press.

Kraft, D.H., Petry, F.E., Buckles, B.P. & Sadasivan, T. *(1994).* The use of genetic programming to build queries for information retrieval. *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence (Cat. No.94TH0650-2), p.2 vol. (xx+xiv+862), 468-73 vol.1.*

Miller, G.A., Beckwith, R., Felbaum, C., Gross, D.& Miller, K.(1993). Introduction to WordNet: An On-line Lexical Database. *http://www.cogsci.princeton.edu/~wn/.*

Petrie, C. (1997). What is an Agent?. *Intelligent Agents III, Agent Theories, Architectures and Languages, ECAI'96 Workshop (ATAL) Proceedings.*