UNSW at GeoCLEF 2006

You-Heng Hu and Linlin Ge

School of Surveying and Spatial Information Systems

University of New South Wales

Sydney, Australia

Emails: you-heng.hu@student.unsw.edu.au, l.ge@unsw.edu.au

Abstract

This paper describes our participation in the GeoCLEF monolingual English task of the Cross Language Evaluation Forum 2006. Our retrieval system consists of four modules: the geographic knowledge base; the indexing module; the document retrieval module and the ranking module. The geographic knowledge base provides information about important geographic entities around the world and relationships among them. The indexing module creates and maintains textual and geographic indices for document collections separately. The Boolean model is used in the document retrieval module to retrieve documents that meet both textual and geographic criteria. The ranking module applied ranking functions that are learned using Genetic Programming to the retrieved results. Performance evaluation of the implementation of these system modules is the main objective of this study. The results of our experiments show that the geographic knowledge base, the indexing module and the retrieval module are useful for geographic information retrieval tasks, but the proposed ranking function learning method doesn't work well.

Categories and Subject Descriptors

H.3 [Information Storage and Retrieval]: H.3.1 Content Analysis and Indexing; H.3.3 Information

Search and Retrieval; H.3.4 Systems and Software; H.3.7 Digital Libraries; H.2.3 [Database

Management]: Languages—Query Languages

General Terms

Measurement, Performance, Experimentation

Keywords

Geographic Information Retrieval, Geographic knowledge base, Genetic Programming

1   Introduction

GeoCLEF is a relative new task of the Cross Language Evaluation Forum (CLEF) campaign. The aims of GeoCLEF are to provide a standard test-bed for retrieval performance evaluation of Geographic Information Retrieval (GIR) systems using search tasks involving both geographic and multilingual aspects. A query topic in GeoCLEF generally consists of thematic criteria and geographic criteria that are described within data fields of title, description and narrative. Totally 25 topics are defined for GeoCLEF 2006. An example topic is illustrated in Figure 1. GeoCLEF 2006 tasks can be performed in two contexts, monolingual and bilingual. In the monolingual context both documents and topics are provided in the same language, while in the bilingual context documents and topics are given in different languages. Available language options for document and topic include English, German, Portuguese and Spanish. The English document collections used in GeoCLEF 2006 are the same one that were used in GeoCLEF 2005, which consists of total 169,477 documents including 56,472 documents are from the British newspaper The Glasgow Herald (1995) and 113,005 documents are from the American newspaper the Los Angeles Times (1994). Figure 2 shows an example of GeoCLEF documents.

```
<top>
  <num>GC026</num>
  <EN-title>Wine regions around rivers in Europe</EN-title>
  <EN-desc>Documents about wine regions along the banks of European rivers</EN-desc>
  <EN-narr>Relevant documents describe a wine region along a major river in European countries.
    To be relevant the document must name the region and the river.</EN-narr>
</top>
```

Figure 1 An example topic used in GeoCLEF 2006

```
<DOC>
<DOCNO>GH950407-000040</DOCNO>
<DOCID>GH950407-000040</DOCID>
<DATE>950407</DATE>
<HEADLINE>...in the shadow of Dumbarton Rock</HEADLINE>
<EDITION>1</EDITION>
<PAGE>9</PAGE>
<GRAPHIC>ILLUS</GRAPHIC>
<RECORDNO>979009754</RECORDNO>
<TEXT>
Dumbarton East voters cast their votes at Knoxland Primary School
under the shadow of Dumbarton Rock.
</TEXT>
</DOC>
```

Figure 2 An example GeoCLEF document

Five key challenges are involved in building a GIR system for GeoCLEF 2006 tasks:

(1) Collecting and organising geographic knowledge. A comprehensive geographic knowledge base that provides not only flat gazetteer lists but also relationships between geographic entities are essential for geographic references extraction and grounding during all GIR query parsing and processing procedures. Many flat gazetteer lists have been published by government agencies, research institutions and industry as public resources such as the Alexandria Digital Library gazetteer developed by the University of California, and the Geographic Names Information System (GNIS) developed by the United States Geological Survey (USGS). However, current availability of data necessary for acquiring geographic relationships is very limited.

(2) Parsing query topics. Two major differences have been observed between topics used in GeoCLEF 2005 and GeoCLEF 2006. Firstly, GeoCLEF 2006 no longer provides explicit expressions for geographic criteria. Topics must be geo-parsed first to identify and extract geographic references (e.g. geographic concepts, entities and relationships) that are embedded in the title, description and narrative tags as free text. Secondly, some new geographic relationships are used in GeoCLEF 2006, such as geographic distance (e.g. within 100km of Frankfurt) and complex geographic expressions (e.g. Northern Germany).

(3) Building a geo-textual indexing scheme for document collections. A geo-textual indexing scheme can be considered as a combination of two independent indexing schemes: a textual indexing scheme that indexing all textual keywords in the documents, and a geographic indexing scheme that indexing all geographic entities recognised from the documents. Both of the textual index and geographic index are necessary for efficient document accessing and searching. Although the computation cost is not considered in the system evaluation, a fast index and search algorithm is necessary for a practical retrieval system where large numbers of documents are evolved.

(4) Retrieving relevant documents. The relevance of a document to a given topic in GIR is determined by not only by thematic similarity measures, but also geographic associations between them. A retrieval model must be defined to constrain the search space to retrieve documents that meet both the thematic criteria and the geographic criteria.

(5) Ranking retrieved documents. A uniform ranking function that takes into textual and geographic similarity measures at same time must be specified to calculate a numerical ranking score for each retrieved document. Different with classical keyword-based ranking methods, semantic relationships between geographic entities and concepts should be employed in this procedure. Retrieval performance of GIR systems is largely depending on the design and optimisation of the ranking function.

This is our first participation in the CLEF tasks. The main objective of this study was to evaluate the performance of the GIR system developed at the School of Surveying and Spatial Information Systems at University of New South Wales, Australia.

Our proposed methodology in the development of a GIR system includes a geographic knowledge base for representation and organisation of geographic data and knowledge, an integrated geo-textual indexing scheme

for document searching, a Boolean model (Salton 1989) for document retrieval and a ranking function discovery algorithm based on Genetic Programming (GP) (Koza 1992). The major efforts of our research are focussed on the collection and utilising of geographic information in information retrieval tasks, existing linguistic techniques are integrated into our system using application program interface provided by various related software packages. For GeoCLEF 2006, we have conducted experiments and submitted runs of monolingual English tasks.

The reset of this paper is organised as follows: Section 2 describes the design and implement of our system for GeoCLEF 2006. Section 3 presents our runs carried out for the monolingual English task. Section 4 discusses the obtained results, and finally, Section 5 concludes the paper and gives future work directions.

2   Approaches for GeoCLEF 2006

This section describes the specific approaches for our participant in the GeoCLEF 2006 monolingual English task. The following subsections discuss the overall software architecture and the detail of each module of our GIR system.

2.1   System Overview

Figure 3 shows the system architecture of our GIR system used in the GeoCLEF 2006, which consists of four major modules: (1) the geographic knowledge base, which provides information about important geographic entities around the world and relationships among them; (2) the indexing module, which creates and maintains textual and geographic indices for document collections; (3) the document retrieval module, which retrieves documents that meet both textual and geographic criteria; and (4) the ranking module, which assigns a rank score to each retrieved document.
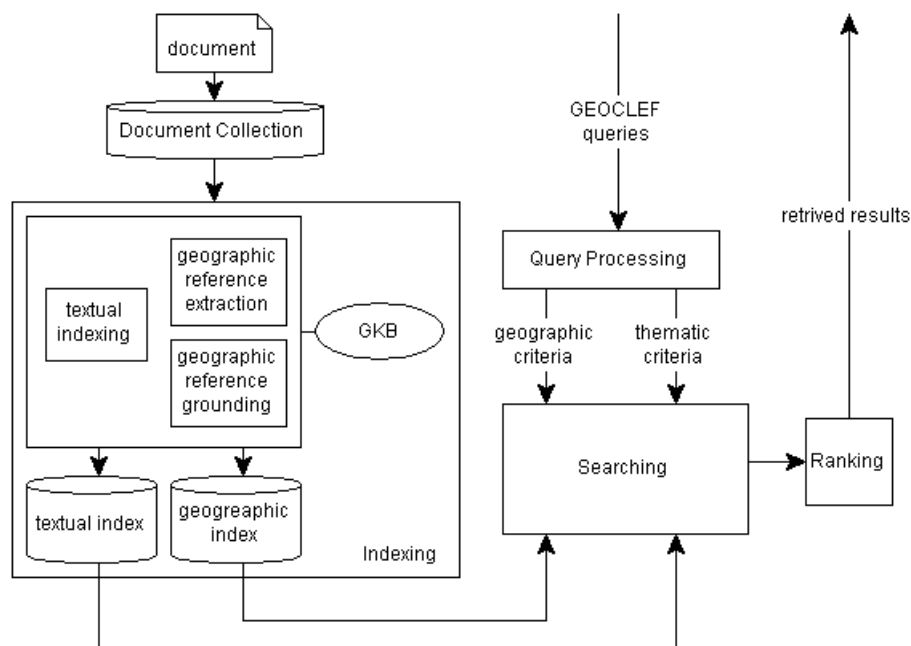

Figure 3 System architecture of the GIR system used in the GeoCLEF 2006

The control and data flow between these modules can be described from following two different perspectives.

From the viewpoint of document indexing, the indexing module creates index entries for all documents in the English document collections. These index entries will be used for the searching module to answer user queries. This phase consists of four operations, one for textual indexing and three for geographic indexing: (1) Extract keywords from the document and then add them to the textual index subsystem; (2) Extract geographic references from the document; (3) Ground geographic references, i.e. associate each geographic reference with a place, and (4) Create geographic index entries and add them to the geographic index subsystem. For geographic

indexing operations, three facilities are essential: (1) the Named Entity Recognition (NER) subsystem, which was used to extract geographic entities from documents; (2) the geographic knowledge base, which provides information (e.g. names, locations, administrative hierarchy, boundaries) about important geographic entities such as countries, subdivisions, cities, oceans, seas, rivers and regions around the world and relationships among them; and (3) the geographic-enabled database, which was used as a geographic index subsystem for storing and searching of geographic indices.

On the other hand, from the viewpoint of the system users, the system takes query topics a starting point. The query processing procedure can be specified as following:

Query parsing: A query topic is parsed and transformed into our internal format, which consists of four main components: keywords, geographic entities, geographic concepts and geographic relationships.

Document Retrieval: During this phase, the retrieval module searches the textual and geographic indices and then retrieves all documents that meet both the thematic criteria and the geographic criteria. The Boolean model was used in the document retrieval module. The textual indexing scheme is used to retrieve all documents that satisfy keyword-based thematic criteria, and the geographic indexing scheme is used to retrieve all documents that satisfy geographic criteria. A Boolean AND operator was used to retrieve documents that appear in both result sets

Document Ranking: After the system retrieves all relevant documents, the ranking module calculates a numeric score for each retrieved document based on the similarity measure between the document and the user query. These scores are then used to rank results. GP was used to learn ranking functions for the ranking module. The reason GP is selected is that firstly both linear and nonlinear functions can be discovered using GP, and secondly previous work in conventional IR systems has shown that ranking functions learned using GP could achieve significant improvement in retrieval performance (Fan, Gordon & Pathak 2005).

The JAVA programming language was used to implement the whole system and the MySQL database (c.f. http://www.mysql.com), an open source relational database management system (RDBMS) was used as the backend database for geographic indexing and searching. The Lucene search engine (c.f. http://lucene.apache.org), an Apache open source project that provides full-text search functionalities, was used for textual indexing and searching, and the Alias-I LingPipe system (c.f. http://www.alias-i.com/lingpipe) was used for NER.

2.2   *Geographic knowledge base*

A geographic knowledge base is a repository for representation and organisation of geographic data and knowledge. Similar with the approaches adopted by Chaves, Silva & Martins (2005) and Souza et al. (2005). The data schema of our geographic knowledge base is defined using the object-orient modelling method (Rumbaugh et al. 1991). Figure 4 shows the class diagram for the schema. Geographic entities are described by their names, types and associated geometric features (e.g. coordinate pairs, minimum bounding rectangles). Another important class is the *Relationship* class, which are used to describe semantic relationships between geographic entities. Two subclasses of *Relationship* are explicated included in the model: *part-of* and *adjacency*. Instances of *part-of* include, for examples, a geographic entity is at lower administrative hierarchical level of another geographic entity (e. g. the state of New South Wales is part of Australia), and a geographic entity is physically inside another geographic entity (e.g. the Tasman Sea is part of the Pacific Ocean). Instances of *adjacency* include, for examples, two geographic entities have adjacent boundaries (e.g. the United States and Canada, Australia and the Indian Ocean). Other relationships can be derived from these two relationships as well. The one very useful for grounding is the *similar* relationship. Two geographic entities are *similar* if they are both *part-of* another geographic entity. For example, the state of New South Wales, Australia and the state of Queensland are *similar*, because both of them are state level administrative subdivisions of Australia. Australia and China are *similar*, because both of them are country-level geographic entities.
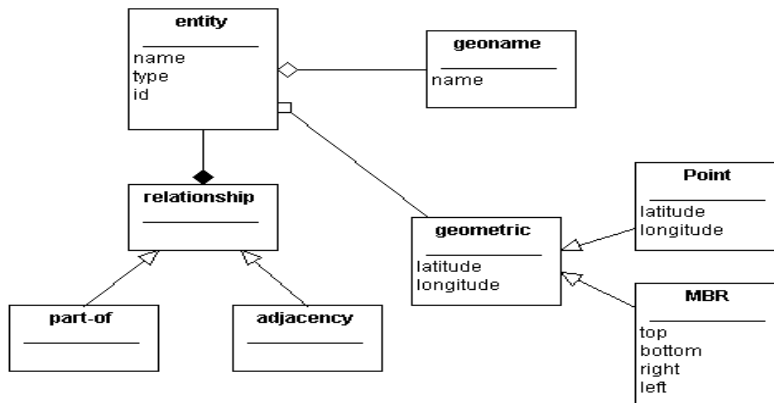
Figure 4 Class diagram of the geographic knowledge base data schema

Data in our geographic knowledge base is collected from various public sources and compiled into the MySQL database. The main resources for carrying out our experiments are listed in Table 1. The statistics of our geographic knowledge base are given in Table 2.

Resource	Geographic data
The Federal Information Processing Standard Publication 10-4: *Countries, Dependencies, Areas of Special Sovereignty, and Their Principal Administrative Divisions*	countries, administrative divisions
The World Factbook published by the Central Intelligence Agency of the United States.	border countries, coastlines, country capital cities
The Wikipedia (c.f. http://en.wikipedia.org/)	oceans, seas, gulfs, rivers, regions
Large cities in the world collected from TravelGIS.com	Cities
The Standard Country and Area Codes Classifications (M49) published by the United Nations Statistics Division	regions, continents
The ESRI Gazetteer Server developed by the Environmental Systems Research Institute, Inc.	Minimum Boundary Rectangle (MBR) of countries
The WordNet developed by the Cognitive Science Laboratory at Princeton University	variant place names

Table 1 Resources used for the geographic knowledge base

Description	Statistic
Number of distinct geographic entities/names	7817/8612
- Number of countries/names	266/502
- Number of administrative divisions/names	3124/3358
- Number of cities/names	3215/3456
- Number of oceans, seas, gulfs, rivers/names	849/921
- Number of regions/names	363/375
Average names per entity	1.10
Number of relationships	9287
- Number of part-of relationships	8203
- Number of adjacency relationships	1084
Number of entities that have only one name	7266 (92.95%)
Number of entities without any relationship	69 (0.88%)
Number of entities without any part-of relationship	123 (1.57%)
Number of entities without any adjacency relationship	6828 (87.35%)

Table 2 Statistics for the geographic knowledge base

2.3 Textual-Geo indexing

Our system creates and maintains the textual index and the geographic index separately. The textual index was built using Lucene with its build-in support for stop words removing and stemming. The indexing technology implemented by Lucene is called inverted index (Araujo, Navarro & Ziviani 1997), which composed of four elements: *documents*, *fields*, *terms* and *occurrences*. A Lucene index contains all documents in the collection. Each document is composed of a list of fields. Each filed is composed of a sequence of named terms. Each term is a *<name, value>* pair that both of the name and the value are represented as textual strings. The *occurrences* store the documents and the positions of each term where it appears. Lucene index entries are stored using file systems. Two fields are used in our textual indexing scheme: *docno* and *text*. The *docno* field is used as the unique identification of each document and the *text* field contains the text body of the document.

Stop words are words that do not have semantic meaning (e.g. a, the) or occurs in many of the documents in the collection (e.g. say, you). These words are not useful for information retrieval tasks and can be eliminated during the indexing procedure in order to reduce the size of index files. The SMART stop word list compiled by Salton (1971) was used in our system. Another important technique for improving retrieval performance and reducing index size is stemming, which reduces words to their grammatical root. For examples, all words of attractive, attraction, attracted and attracting are stemmed to the word attract. Our system used the Porter stemming algorithm (Porter 1980).

The geographic index was built as a procedure of three steps. The first step performed a simple string matching against all documents in the collections utilising the place name list derived from our geographic knowledge base. Similarly with the stop words in textual indexing, there are some place names were eliminated during this step, examples include: Mobile (a city in Alabama, U.S.), Orange (a city in Texas, U.S.) and Reading (a town in Berkshire, U.K.). The second step performed a NER process using the Alias-I LingPipe APIs to tag three types of named entities: PERSON, LOCATION and ORGANISATION. The final step matched result sets from the two previous steps using following rules: (1) for each string that found in the first step, it was eliminated if it was tagged as a non-location entity (i.e. PERSON or ORGANISATION) in the second step, otherwise it was added to the geographic index; (2) for each place name in the stop word list of the first step, it was added to the geographic index if it was tagged as a location entity in the second step.

Geographic index entries in our system consist of three fields: *docno, place name*, and *appearnum*, where *docno* is the same one in the textual index, the *appearnum* filed was the number of how many times the place name appears in the document.

The geographic index was implemented using the MySQL database server. Two database indices were then applied to the *docno* and *place name* fields to achieve fast search and retrieval. The *docno* fields were also used to link textual and geographic index entries.

2.4 Document Retrieval

The retrieval of relevant documents in our system is a four-phase procedure that involves query parsing, textual searching, geographic searching and Boolean intersection.

The GeoCLEF query topics were in general modelled as a tuple *Q = (textual criteria, geographic criteria)* in our system. However, the query parser is configured in an ad hoc fashion for the GeoCLEF 2006 tasks at hand. Given a topic, the parser performs the following steps: (1) Removes guidance information, such as "Documents about" and"Relevant documents describe", description about irrelevant documents is removed as well. (2) Extracts geographic criteria using string matching with names and types data obtained from the geographic knowledge base. The discovered geographic entities, geographic relationships and geographic concepts are added to the geographic criteria. Then geographic related words are removed. (3) Stop words are removed using the SMART list (4) the remaining text is treated as textual keywords. All-capitalised abbreviations are expanded using WordNet APIs (e.g. *ETA* in GC049). Examples of parsing results are shown in Table 3 and Table 4 for GC029 and GC036 respectively.

Textual keywords	Diamond trade
geographic entities	Angola, South Africa
geographic concepts	
geographic relationships	In

Table 3 Topic parsing results of GC029 using title and description

Textual keywords	Automotive industry coastal factories shore economic social events happening planned joint-ventures strikes
geographic entities	Sea of Japan
geographic concepts	Cities
geographic relationships	Adjacency

Table 4 Topic parsing results of GC036 using title, description and narrative

After query topics are parsed, the Lucene search engine is used to retrieve all documents that contain the textual keywords, and the geographic index is used to retrieve all documents that meet the geographic criteria. For textual searching, the same stemming algorithm is used to transform each keyword into its stem. For geographic searching, the geographic knowledge base plays an essential role. To determine whether a document meets the geographic criterion, not only geographic entities found from the document, but also their related entities found from the geographic knowledge base are taken into account.

In the current implementation, the textual searching and the geographic searching are performed sequentially. It is possible to apply advanced parallel computing techniques to reduce the system computation time.

Having retrieved the two results sets, the final step intersects them using the Boolean AND operator, only documents that appear in both result sets are considered as relevant documents.

2.5 *Document Ranking*

A Genetic Programming-based algorithm is developed in our system to discover ranking functions. This algorithm utilises genetic operators such as reproduction, crossover and mutation on each generation of individuals to produce new generations of better solutions. The implementation of our GP algorithm consists of three elements: (1) A set of terminals and functions that can be as logic unit of a ranking function; (2) A fitness measure evaluates how well each individual in the population is for the problem; and (3) An evolution strategy specifies control mechanisms of GP evolution process.

- **Terminals and Functions**

Terminals reflect logical views of documents and user queries. Terminals can be categorised into two groups: local and global. The local data reflects content of one particular document. In contrast, global data reflects content of the whole collection. Terminals used in our system are listed in Table 5, in which *DOC_LENGTH, LUCENE_SCORE, GEO_NAME_NUM, GEO_NAME_COUNT, GEO_ENTITY_COUNT, GEO_RELATED_COUNT* and *GEO_COUNT* are examples of local data. *DOC_COUNT, GEO_NAME_DOC_COUNT, NAME_COUNT* and *ENTITY_COUNT* are examples of global data.

Name	Description
DOC_COUNT	number of documents in the collection
DOC_LENGTH	length of the document
LUCENE_SCORE	Lucene ranking score of the document
GEO_NAME_NUM	how many geographic names in the document
GEO_NAME_COUNT	total number of geographic names of all geographic entities discovered from the document
GEO_ENTITY_COUNT	how many entities that have the geographic name
GEO_RELATED_COUNT	how many entities that have the geographic name and related to the query
GEO_NAME_DOC_COUNT	number of documents that have the geographic name
GEO_COUNT	how many times of the geographic name appears in the document
NAME_COUNT	number of geographic names in the geographic knowledge base
ENTITY_COUNT	number of entities in the geographic knowledge base

Table 5 Terminals used in the ranking function learning process

Functions reflect the relationships between terminals. Functions used in our experiments include addition (+), subtraction (-), multiplication (×), division (/) and natural logarithm (log). Additional controls are added to the function definitions to handle exception cases, such as divided by zero, and logarithm of non-positive numbers. Using above terminals and functions, various ranking functions can be represented as a combination (linear or non-linear) of them. A tree structure that has terminals as leaf nodes and functions as inner nodes is used to visualise GP individuals (e.g. ranking functions).

- **Fitness Functions**

Fitness functions play a crucial role in a GP implementation. An individual with higher fitness value has more chance of being selected for genetic operations due to the probabilistic-based nature of GP evolution. A fitness function that correctly reflects how well each individual is will help to reduce learning time and to produce better solution. Three fitness functions are used in our system. All of them take into account the order of retrieved results, higher fitness values are assigned to the solutions that retrieve relevant documents quickly. The return value of all these four fitness functions is granted to be between 0 and 1, inclusively.

F_P50. The first fitness function returns the arithmetic mean of the precision values at 50% recall for all queries as results. The definition of this function is given as following:

$$F\_P5 = \frac{1}{Q} \times \sum_{i=1}^{Q} P_{i,5}$$

where Q is the total number of queries, $P_{i,5}$ is the precision value at fifth recall level of the 11 standard recall level (i.e. 50% recall) of the ith query. This fitness function is referred to as *F_P5* in our experiments.

F_MAP. The second fitness function utilises the idea of average precision at seen relevant documents. The definition of this function is given as followings.

$$AP_i = \frac{1}{R_i} \times \sum_{j=1}^{D_i} \left(r(d_j) \times \frac{\sum_{k=1}^{j} r(d_k)}{j} \right)$$

$$F\_MAP = \frac{1}{Q} \times AP_i$$

where R_i is the total number of relevant documents for the ith query, $r(d_j)$ is a function that returns 1 if the document d_j is relevant for ith query and returns 0 otherwise. AP_i is the average of precisions at seen relevant documents (i.e. a precision value is calculated when a new relevant document is observed) for the ith query, and Q is the total number of queries. This function first calculates AP_i for each query, and then returns the arithmetic mean of all AP_i as the result. This fitness function is referred to as F_MAP in our experiments.

F_WP. The last fitness function is of our own design, which utilises the weighted sum of precision values on the 11 standard recall levels. The definition of this function is given in as following:

$$WP_i = \sum_{i=0}^{10} \frac{1}{(i+1)^m} P_{i,j}$$

$$F\_WP = \frac{1}{Q} \times \sum_{i=1}^{Q} WP_i$$

where $P_{i,j}$ is the precision value at the jth recall level of the 11 standard recall levels for the ith query, m is a positive scaling factor determined from experiments, WP_i is the weighted sum of $P_{i,j}$ and Q is the total number of queries. This function first calculates WP_i for each query, and then returns the arithmetic mean of all WP_i as the result. This fitness function is referred to as F_WP in our experiments.

- **GP Evolution Strategy**

GP evolution strategy specifies control mechanisms of the evolution process. The key elements of our implementation are described as following.

Initialisation of the first generation. N individuals are created and are added to the population. The creation of individuals of the first generation can be described as a random selection procedure, which assumes each terminal and function has a same probability of being selected. A node is first selected and is added to the tree as the root node. If the selected is a terminal, the procedure is finished. If the selected is a function, zero or more sub-trees are needed created. The number of sub-trees the function node has is decided by the function definition. Sub-trees are recursively created using the same method until no sub-tree is required.

Genetic operators. The evolution procedure in GP can be described as a repeated procedure that creates a new generation by applying various genetic operators on the previous generation. Four genetic operators are used in our method to create new generations, including: creation, crossover, reproduction and mutation.

Selection of parents. The crossover, reproduction and mutation operators require one or two individuals as parents. The roulette wheel selection, which is the most often used selection strategy in GP, is used in our method for parent selection.

Termination of the evolution. Our learning procedure is terminated after G generations are generated and evaluated, where G is decided by experiments. The best individual of all generations is selected as the final results (e.g. the ranking function).

3   Experiments

Totally five runs were submitted for GeoCLEF 2006 monolingual English tasks. Summarises of our submissions are given in Table 6.

Run	Description
unswTitleBase	(Mandatory) title and description, ranked using Lucene score only.
unswNarrBaseline	(Mandatory) title, description and narrative, ranked using Lucene score only.
unswTitleF46	title and description, ranked using the ranking function discovered using f_WP with m = 6
unswNarrF41	title, description and narrative, ranked using the ranking function discovered using f_WP with m = 1
unswNarrMap	title, description and narrative, ranked using the ranking function discovered using F_MAP

Table 6 The runs submitted to the GeoCLEF 2006 monolingual English tasks

The detail of each submitted runs are described as following:

title_baseline: This run uses the title and description tags of the topics for query parsing and searching. After relevant documents are retrieved, the Lucene ranking scores are used to rank results.

narr_baseline: This run uses the title, description and narrative tags of the topics for query parsing and searching. After relevant documents are retrieved, the Lucene ranking scores are used to rank results.

The above two were mandatory runs requested. In addition, these two runs utilise the Lucene ranking scores to rank retrieved documents.

unswTitleF46: This run uses the title and description tags of the topics for query parsing and searching. After relevant documents are retrieved, the ranking function given below was used to rank results. This ranking function was discovered using fitness function *F_WP* with m = 6.

$$LUCENE\_SCORE * LUCENE\_SCORE * LUCENE\_SCORE / GEO\_NAME\_COUNT$$

unswNarrF41: This run uses the title, description and narrative tags of the topics for query parsing and searching. After relevant documents are retrieved, the ranking function given below was used to rank results. This ranking function was discovered using fitness function *F_WP* with m = 1.

$$LUCENE\_SCORE * LUCENE\_SCORE * LUCENE\_SCORE * GEO\_RELATED\_COUNT / DOC\_LENGTH$$

unswNarrMap: This run uses the title, description and narrative tags of the topics for query parsing and searching. After relevant documents are retrieved, the ranking function given below was used to rank results. This ranking function was discovered using fitness function *F_MAP*.

$$GEO\_RELATED\_COUNT * LUCENE\_SCORE / DOC\_COUNT / DOC\_COUNT$$

The ranking functions used in above three runs were discovered using our GP learning algorithm which utilised the GeoCLEF 2005 topics and relevance judgments as training data. It is important to note the same query parsing and document retrieval processing were applied to the GeoCLEF 2005 topics, which means those GeoCLEF 2005 geographic-related tags were ignored.

4   Results

Table 7 summarises the results of our GIR system at the GeoCLEF 2006 Monolingual English tasks using evaluation metrics include Average Precision, R-Precision and the increment over the mean average precision

(19.75%) obtained from all submitted runs. The precision average values for individual queries are shown in Table 8.

Run	AvgP. (%)	R-Precision (%)	Δ AvgP. Diff over GeoCLEF Avg P. (%)
unswTitleBase	26.22	28.21	+32.75
unswNarrBaseline	27.58	25.88	**+39.64**
unswTitleF46	22.15	26.87	+12.15
unswNarrF41	4.01	4.06	-79.70
UnswNarrMap	4.00	4.06	-79.75

Table 7 GeoCLEF 2006 monolingual English tasks results

Topic	UnswTitleBase (%)	unswNarrBaseline (%)	unswTitleF46 (%)	unswNarrF41 (%)	UnswNarrMap (%)
GC026	30.94	30.94	15.04	0.58	0.56
GC027	10.26	10.26	12.32	10.26	10.26
GC028	7.79	3.35	5.09	0.36	0.31
GC029	24.50	4.55	16.33	0.53	0.53
GC030	77.22	77.22	61.69	6.55	6.55
GC031	4.75	5.37	5.09	3.31	3.31
GC032	73.34	93.84	53.54	5.71	5.71
GC033	46.88	38.88	44.77	33.71	33.71
GC034	21.43	2.30	38.46	0.14	0.13
GC035	32.79	43.80	28.06	3.19	3.11
GC036	0.00	0.00	0.00	0.00	0.00
GC037	21.38	21.38	13.17	0.81	0.81
GC038	6.25	14.29	0.12	0.12	0.12
GC039	46.96	45.42	34.07	3.50	3.50
GC040	15.86	15.86	13.65	0.34	0.30
GC041	0.00	0.00	0.00	0.00	0.00
GC042	10.10	36.67	1.04	0.33	0.33
GC043	6.75	16.50	4.33	0.55	0.54
GC044	21.34	17.23	13.80	4.78	4.78
GC045	1.85	3.96	2.38	1.42	1.42
GC046	66.67	66.67	66.67	3.90	3.90
GC047	8.88	11.41	9.80	1.02	0.98
GC048	58.52	68.54	51.55	8.06	8.06
GC049	50.00	50.00	50.00	0.09	0.09
GC050	11.06	11.06	12.73	11.06	11.06

Table 8 Precision average values for individual queries

Several observations are made from the obtained results: firstly the geographic knowledge base and the retrieve model used in our system show their potential usefulness in GIR as we can see from the higher average precision values of unswTitleBase (26.22%) and unswNarrBaseline (27.58%), which shows a 32.75% and a 39.64% improvement comparing to the mean average precision of all GeoCLEF 2006 Monolingual English runs.

Secondly, the ranking function learning algorithm used in our system doesn't work well for GeoCLEF tasks, particular for those runs (i.e. unswNarrF41 and unswNarrMap) that utilises narrative information of the queries. We suppose such behaviour is due to a strong over-training effect. However, the unswTitleF46 run performed better than the two base line runs in a small set of topics (i.e. GC027, GC034, GC044 and GC059).

Thirdly, it is not immediately obvious that the narrative information should be included in the query processing. The unswTitleBase run achieves the same performance as the unswNarrBaseline run in 10 topics (i.e. GC026, GC027, GC030, GC036, GC037, GC040, GC041, GC046, GC049 and GC059), and it even achieves better results in 6 topics (i.e. GC028, GC029, GC033, GC034, GC039 and GC044).

Lastly, it is interesting to see that our system didn't retrieve any relevant document for topic GC036 and GC041. It is not surprised for GC036, as there hasn't any document is identified as relevant in the assessment result. While for GC041, which talks about "Shipwrecks in the Atlantic Ocean", the keyword "shipwreck" doesn't appear in any of the four relevant documents (i.e. GH950210-000051, GH950210-000197, LA071094-0080 and LA121094-0182).

5   Conclusions

This paper proposed the GIR system that has been developed for our participation in the GeoCLEF 2006 monolingual English task. The key components of the system including a geographic knowledge base, an integrated geo-textual indexing scheme, a Boolean retrieval model and a Genetic Programming-based ranking function discovery algorithm are described in detail. Using this system, several experiments were conducted and five runs were submitted for monolingual English tasks. The results shows that the geographic knowledge base and the retrieval model are useful for geographic information retrieval tasks, but the proposed ranking function learning method doesn't work well.

Clearly there is much work to be done in order to fully understand the implications of the experiment results. The future research directions that we plan to pursue include: (1) the establishing of a unified GIR retrieval model that is capable to combine textual and geographic representation and ranking of documents in a suitable framework: (2) the utilising of parallel computation techniques to improve the system computation performance and (3) the extending of our geographic knowledge base by adding more feature types, such as population number and economic importance, which may affect relevance judgment and ranking.

Reference

Araujo, M, Navarro, G & Ziviani, N (1997), 'Large text searching allowing errors', in Proceedings of the 4th South American Workshop on String Processing. pp. 2-20.

Chaves, M, Silva, M & Martins, B (2005), 'A Geographic Knowledge Base for Semantic Web Applications', *Proceedings of SBBD-05, the 20th Brazilian Symposium on Databases*.

Fan, WP, Gordon, MDP & Pathak, PP (2005), 'Genetic Programming-Based Discovery of Ranking Functions for Effective Web Search', *Journal of Management Information Systems*, vol. 21, no. 4, pp. 37-56.

Koza, JR (1992), *Genetic Programming: On the Programming of Computers by Means of Natural Selection. Complex Adaptive Systems*, MIT Press, 840.

Porter, MF (1980), 'An algorithm for suffix stripping', *Program*, vol. 14, no. 3, pp. 130–137.

Rumbaugh, J, Blaha, M, Premerlani, W, Eddy, F & Lorensen, W (1991), *Object-oriented modeling and design*, Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 500.

Salton, G 1971, *The SMART Information Retrieval System*, Prentice Hall, Englewood Clis, NJ.

---- (1989), *Automatic text processing: the transformation, analysis, and retrieval of information by computer*, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 530.

Souza, LA, Davis Jr, CA, Borges, KAV, Delboni, TM & Laender, AHF (2005), 'The Role of Gazetteers in Geographic Knowledge Discovery on the Web', *Proceedings of the 3rd Latin American Web Congress, Buenos Aires, Argentina*.