Indexing implicit locations for geographical information retrieval

Zhisheng Li[1], Chong Wang[2], Xing Xie[2], Xufa Wang[1], Wei-Ying Ma[2]

[1]Department of Computer Science,
University of Sci. & Tech. of China
Hefei, Anhui, 230026, P.R. China
zsli@mail.ustc.edu.cn, xfwang@ustc.edu.cn

[2]Microsoft Research Asia,
5F, Sigma Center, No.49, Zhichun Road,
Beijing, 100080, P.R. China
{chwang, xingx, wyma}@microsoft.com

ABSTRACT
Local search has become a hot topic recently in information retrieval research area. How to retrieve geographical information correctly and efficiently is a key challenge to location-based search services. In this paper, we present a GIR (geographical information retrieval) system which uses implicit locations to improve retrieval performance. Experimental results based on Geo-CLEF 2006 (a cross-language geographical retrieval track which is part of Cross Language Evaluation Forum) data sets show that the proposed method can retrieve geographical information better than previous approaches.

Keywords
Location extraction, implicit location, geographical information retrieval.

1. INTRODUCTION
A geographical web search engine allows users to constrain queries to specific locations, while traditional search engines do not give extra notice on the geographical information. Recently many research works have been carried out in this direction. Location extraction and geo focus detection of web pages have been discussed in [1][5]. Efficient query processing and comparison of different indexing algorithms have been introduced in [2][3][8]. In [3], the authors designed a geo-indexing scheme based on geo-scope which is the geographical area of web pages. In [8], the authors proposed to use a grid-based spatial indexing scheme and a spatial-textual combined index to reduce the query time for large data sets. However, they didn't consider implicit locations. For example, for a query "snowstorms in North America", traditional methods simply return all the web pages that include "North America". In fact, if a web page includes "Canada", "United States of America", or "Mexico", it is also relevant to the query. "North America" can be seen as the implicit location for "Canada". In this paper, we define implicit location as the ancestors of the explicit locations mentioned in the documents and propose an implicit location based geographical index structure and compare its performance with different indexing methods. Experimental results show that our approach is better than previous ones.

GIR'06, August 10–11, 2006, Seattle, Washington, USA.

2. IMPLICIT LOCATION
The index structure of a GIR system usually composes of two parts: text index and geo-index. On the other hand, a geographical query usually consists of three parts: textual terms, spatial relationship and geographical terms. For example, one query in GeoCLEF 2006 [7] is "Snowstorms in North America", where "Snowstorms" indicates what the user intends to know, and "North America" is the scope of the area that the user is interested in. We can retrieve a set of documents related to the textual terms (snowstorms) through the text index and another set of documents whose geographical focuses are related to the query area through geographical index. These two sets will be merged and ranked according to a combined ranking function. In this paper, we suppose the query location input is through text, not map.

There are several possible solutions to the implicit location problem. One method is to use query expansion based on pseudo-feedback [4]. The idea is to find the most relevant locations to the original query location from the returned documents, and then use those locations to compose a new query. The shortcoming of the pseudo-feedback approach is that it depends on the documents so much that many irrelevant locations might be added to the new query. Another method is to expand the query based on gazetteers. The locations that are covered by the query location will be used for expansion. Unfortunately, we may come up with a very long query after such an expansion because too many children locations will be added. Moreover, such a long expansion will greatly affect the retrieval speed. Geo-index structures like grid-based index and R*-tree can solve the implicit location problem too. In this paper, we assume the query location is input by text which is common for current search engines.

In our approach, explicit locations and implicit locations are indexed together and different geo-confidence scores are assigned to them. The advantage of this mechanism is that no query expansion is necessary and implicit location information can be computed offline for fast retrieval.

In [6], the authors concluded that the most common geographical relationship used in queries is "in". Actually, if no spatial relationship exists in the query, we can safely assume the relationship is "in". For example, when a user wants to know

information about "Independence movement in Quebec" (a query in GeoCLEF 2006), it means he/she wants to get the information covered by the query geographical area "Quebec", and would feel uncomfortable if a GIR system returns documents about "Independence movement in Canada" or "Independence movement in North America". Therefore, in our index algorithm, the implicit locations will not be expanded to lower levels because users usually don't need information about upper locations when they seek for information about lower ones. When "Canada" appears in one document, the document ID shouldn't be appended to the inverted lists of "Quebec" or other children locations of "Canada". In order to obtain the implicit locations, we first adopt the focus detecting algorithm described in [5] to get the geo focuses of web pages. Afterwards we add the ancestors of these explicit focuses as implicit locations, but with lower confidence values.

In this paper, we adopt two types of geo-indexes: one is called focus-index, which utilizes the inverted index to store all the explicit, and implicit locations of documents (see Figure 1); the other is called grid-index, which divides the surface of the Earth into 1000 × 2000 grids. All the documents will be indexed by these grids according to their geo focuses. The reason for adopting grid-index is that some topics in GeoCLEF can't be solved by only focus-index due to the spatial relationship other than "in" (like "near").

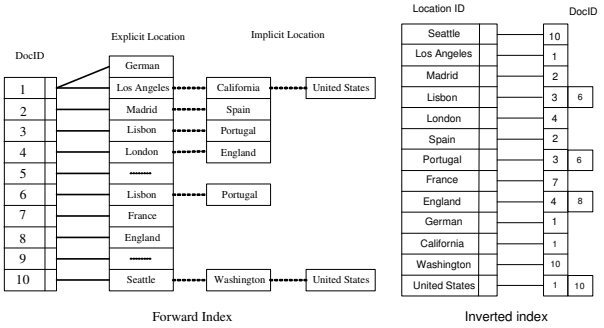

Figure 1. Inverted index of both explicit and implicit locations.

For focus-index, the matched docID list can be retrieved by looking up the locationID in the inverted index. For grid-index, we can get the docID list by looking up the grids that the query location covers. We first retrieve two lists of documents relevant to the textual terms and the geographical terms respectively, and then merge them to get the final results. A combined ranking function $R_{combined} = R_{text} \times \alpha + R_{geo} \times (1- \alpha)$, where R_{text} is the textual relevance score and R_{geo} is the geo-confidence score, is computed and used to re-rank the results. Experiments show that textual relevance scores should be weighted higher than geo-relevance scores.

3. EXPERIMENTS

We now present the results of our experimental evaluation. Our experiments used the corpus of GeoCLEF 2006 as the data set. We implemented a GIR system that composes of location extraction component, geo-index component and geo-ranking component. The location extraction component deals with location extraction, disambiguation and focus-detection. The geo-index and ranking components are based on the methods described in previous sections.

We ran experiments on 9 topics in GeoCLEF 2006 with $\alpha = 0.8$ and label the top 20 resulting documents for each topic (if the number of resulting documents were less than 20, we chose all of them). Table 1 shows the precision of three different methods: pure-text index, explicit location index without query expansion and implicit location index. In the pure-text index, we only used the textual relevance to rank the results. The number of correct results and returned results are presented too.

Table 1. P@20 of three indexing methods.

Topic title	Pure-text index	Explicit location index	Implicit location index
C26 Wine regions around rivers in Europe	0.50	0.59 (10/17)	0.85 (17/20)
C28 Snowstorms in North America	0.25	0.0 (0/1)	0.75 (15/20)
C30 Car bombings near Madrid	0.50	0.60 (6/10)	0.50 (6/12)
C31 Combats and embargo in the northern part of Iraq	0.85	0.95 (19/20)	0.95 (19/20)
C37 Archeology in the Middle East	0.75	0.85 (12/14)	0.87 (15/17)
C38 Solar or lunar eclipse in Southeast Asia	0.25	0.60 (12/20)	0.80 (16/20)
C42 Regional elections in Northern Germany	0.20	0.85 (6/7)	0.75 (15/20)
C45 Tourism in Northeast Brazil	0.10	0.43 (3/7)	0.75 (15/20)
C46 Forest fires in Northern Portugal	0.15	1.0 (2/2)	1.0 (2/2)

From Table 1, we can see that implicit location index is better than the other two indexing methods for most of the queries. For C28, the implicit location index largely outperforms the explicit location index because few documents mention "Snowstorms" and "North America" together in our test set. In this case, implicit locations are very helpful. For C31, the precision of pure-text index is close to that of implicit location index due to that "Iraq" always appeared with explicit locations like "Bagdad" in the test set. Therefore, the implicit location index does not outperform the other two in such a situation. Compared with the explicit location index, implicit location index can improve the recall but it may also bring in some irrelevant documents.

Table 2 shows the running time for explicit location index with query expansion (based on gazetteer) and implicit location index. Though the precision of explicit location index with query expansion is equal to that of implicit location index, the running time of implicit location index is usually smaller for "in" queries. The time costs in Table 2 seem to be large because we include the I/O time and the system hasn't been optimized.

There are also some shortcomings for implicit location index. For example, if the explicit locations are not extracted correctly, the errors will propagate so that the retrieval precision will decrease.

Table 2. Running time (in seconds) for explicit location index with query expansion and implicit location index.

Topic title	Explicit location index with query expansion	Implicit location index
C26 Wine regions around rivers in Europe	9.391	5.75
C28 Snowstorms in North America	359.75	11.96
C30 Car bombings near Madrid	4.516	4.203
C31 Combats and embargo in the northern part of Iraq	5.172	2.906
C37 Archeology in the Middle East	0.297	0.172
C38 Solar or lunar eclipse in Southeast Asia	0.625	0.484
C42 Regional elections in Northern Germany	4.953	3.625
C45 Tourism in Northeast Brazil	0.985	1
C46 Forest fires in Northern Portugal	2.828	3.109

4. CONCLUSIONS

Geographical information retrieval is a new area that builds on both traditional information retrieval and geographical information systems. This paper presents a geo-indexing method using implicit location. Experimental results show that implicit location based indexing method can achieve better performance than those indexing methods without implicit location and faster than query expansion method in most of the time. In the future, we will try to further improve the retrieval accuracy and speed of GIR systems.

5. REFERENCES.

[1] E. Amitay, N. Har'EI, etc. Web-a-where: Geotagging Web Content. SIGIR 2004.

[2] Y.Y. Chen, T. Suel and A. Markowitz. Efficient Query Processing in Geographical Web Search Engines. SIGMOD'06, Chicago, IL, USA.

[3] B. Martins, M. J. Silva and L. Andrade. Indexing and Ranking in Geo-IR Systems. GIR'05, Bremen, Germany.

[4] A. Chen. Cross-Language Retrieval Experiments at CLEF 2002. Lecture Notes in Computer Science 2785, Springer 2003.

[5] C. Wang, X. Xie, etc. Detecting Geographical Locations from Web Resources. GIR'05, Bremen, Germany.

[6] M. Sanderson and J. Kohler. Analyzing Geographical Queries. GIR'04, Sheffield, UK.

[7] GeoCLEF 2006. http://ir.shef.ac.uk/geoclef/

[8] Jones, C. B., Abdelmoty, A.I., etc. The SPIRIT Spatial Search Engine: Architecture, Ontologies and Spatial Indexing. Lecture Notes in Computer Science 3234, 2004.