Retrieving Documents with Geographic References Using a Spatial Index Structure Based on Ontologies[*]

Miguel R. Luaces, Ángeles S. Places, Francisco J. Rodríguez, and Diego Seco

Databases Laboratory, University of A Coruña
Campus de Elviña, 15071 A Coruña, Spain
{luaces,asplaces,franjrm,dseco}@udc.es

Abstract. Both *Geographic Information Systems* and *Information Retrieval* have been very active research fields in the last decades. Lately, a new research field called *Geographic Information Retrieval* has appeared from the intersection of these two fields. The main goal of this field is to define index structures and techniques to efficiently store and retrieve documents using both the text and the geographic references contained within the text.

We present in this paper a new index structure that combines an inverted index, a spatial index, and an ontology-based structure. This structure improves the query capabilities of other proposals. In addition, we describe the architecture of a system for geographic information retrieval that uses this new index structure. This architecture defines a workflow for the extraction of the geographic references in the document.

1 Introduction

Two research fields that have received much attention during the last years are those of Information Retrieval [1] and Geographic Information Systems [2]. These fields have produced industry product lines such as digital libraries, document databases, web search engines, and spatial data infrastructures [3]. During the last decades these two research fields have advanced independently. Although it is very common that textual and geographic information occur together in information systems, the geographic references of documents are rarely used in information retrieval systems. Few index structures or retrieval algorithms take into account the spatial nature of geographic references embedded within documents. Pure textual techniques focus only on the language aspects of the documents and pure spatial techniques focus only on the geographic aspects of the documents. None of them are suitable for a combined approach to information retrieval because they completely neglect the other type of information.

[*] This work has been partially supported by "Ministerio de Educación y Ciencia" (PGE y FEDER) ref. TIN2006-16071-C03-03, by "Xunta de Galicia" ref. PGIDIT05SIN10502PR and ref. 2006/4, and by "Consellería de Educación e Ordenación Universitaria da Xunta de Galicia" for Diego Seco.

I.-Y. Song et al. (Eds.): ER Workshops 2008, LNCS 5232, pp. 395–404, 2008.

As a result, there is a lack of system architectures, index structures and query languages that combine both types of information.

Some proposals have appeared recently [4,5,6] that define new index structures that take into account both the textual and the geographic aspects of a document. These proposals are the origin of a new research field called Geographic Information Retrieval (GIR). However, there are some specific particularities of geographic space that are not taken into account by these approaches. Particularly, concepts such as the hierarchical nature of geographic space and the topological relationships between the geographic objects must be considered in order to fully represent the relationships between the documents and to allow new and interesting types of queries to be posed to the system.

In this paper, we present an index structure that takes these issues into account. We first describe some basic concepts and related work in Section 2. Section 3 describes our index structure and the procedures used to built it. Then, in Section 4, we present the general architecture of the system and describe its components. After that, in Section 5, we describe some types of queries that can be answered with this system and we sketch the algorithms that can be used to solve this queries. Furthermore, Section 6 presents some experiments that we made to compare our structure with other ones that use a pure spatial index. Finally, Section 7 presents some conclusions and future lines of work.

2 Related Work

Inverted indexes are considered the classical text indexing technique [1]. The main drawback of these indexes is that geographic references are mostly ignored because place names are considered words just like the others. If the user poses a query such as as *hotels in Spain*, the place name *Spain* is considered a word, and only those documents that contain exactly that word are retrieved. Regarding indexing geographic information, many different spatial index structures have been proposed along the years. A good survey of these structures can be found in [7]. A drawback of spatial index structures is that they do not take into consideration the geographic ontology of the real world. Internal nodes in the structure are meaningless in the real world and it is not possible to associate location-specific information to these nodes because there is no relation at all between the nodes in the spatial index structure and real world locations.

Some work has been done to combine both types of indexes. Finding geographical references in text is a very difficult problem and there have been many papers that deal with different aspects of this problem and describe complete systems such as Web-a-where [8], MetaCarta [9], and STEWARD [4]. The papers about the SPIRIT (Spatially-Aware Information Retrieval on the Internet) project [10,11,12,13] are a very good starting point. In [12], the authors conclude that keeping separate text and spatial indexes, instead of combining both in one, results in less storage costs but it could lead to higher response times. More recent works can be broadly classified into two categories depending on how they combine textual and spatial indexes. On the one hand, some proposals have appeared that combine textual and spatial aspects in an hybrid index [14,15]. On the other hand, other proposals define structures that keep separate indexes for spatial and text attributes [4,5,6]. Our index structure is part of this second group because this division has many advantages [6]. Nevertheless, none of these approaches take into account the relationships between the geographic objects that they are indexing.

A structure that can properly describe the specific characteristic of geographic space is an *ontology*, which is a formal explicit specification of a shared conceptualization [16,17]. An ontology provides a vocabulary of classes and relations to describe a given scope. In [18], a method is proposed for the efficient management of large spatial ontologies using a spatial index to improve the efficiency of the spatial queries. Furthermore, in [10,13] the authors describe how ontologies are used in query term expansion, relevance ranking, and web resource annotation in the SPIRIT project. However, as far as we know, nobody has ever tried to combine ontologies with other types of indexes to have a hybrid structure that captures both the spatial and the semantic relationships between the geographic objects indexed.

3 Index Structure

Our index architecture has three main components, a textual index, a spatial index, and a place name hash table to optimize the resolution of a particular type of very common queries. The textual index is built using Lucene [19] by parsing and inserting each of the documents into the index. The *place name hash table* stores for each location name its position in the spatial index structure. This provides direct access to a single node by means of a keyword that is returned by the Geographic Space Ontology Service (see Figure 2) if the word processed is a location name.

The spatial index is based on an ontology [16,17] of the geographic space that describes the concepts in our domain and the relationships that hold between them. There are different ontology languages that provide different formal and reasoning facilities. OWL [20] is a W3C standard language to describe ontologies and can be categorised into three species or sub-languages: OWL-Lite, OWL-DL and OWL-Full. Our spatial ontology is described in OWL-DL and it can be downloaded from the following URL: `http://lbd.udc.es/ontologies/spatialrelations`.

OWL classes can be interpreted as sets that contain individuals (also known as instances). Individuals can be referred to as being *instances of classes*. Our ontology describes eight classes of interest: *SpatialThing*, *GeographicalThing*, *GeographicalRegion*, *GeopoliticalEntity*, *PopulatedPlace*, *Region*, *Country*, and *Continent*. In our ontology there are hierarchical relations among *SpatialThing*, *GeographicalThing*, *GeographicalRegion*, *GeopoliticalEntity* because:

- *GeopoliticalEntity* is subclass of *GeographicalRegion*
- *GeographicalRegion* is subclass of *GeopoliticalEntity*
- *GeopoliticalEntity* is subclass of *GeographicalThing* and
- *GeographicalThing* is subclass of *SpatialThing*.

That is, these four classes are organised into a superclass-subclass hierarchy, which is also known as a taxonomy. Subclasses specialise (are subsumed by) their superclasses. *GeopoliticalEntity* has four subclasses: *PopulatedPlace*, *Country*, *Continent*, and *Region*. All the individuals are members of these subclasses. These four subclasses have an additional necessarily asserted condition regarding their relations with each other. They are connected by the property *spatiallyContainedBy* that describes the existence of an spatial relationship among them. For instance, all the individuals of class *PopulatedPlace* are *spatiallyContainedBy* individuals of class *Region* (described in OWL as *PopulatedPlace spatiallyContainedBy only (AllValuesFrom) Region*). Figure 1 shows an example of these relationships. Ontology classes are represented as circles, individuals as rectangles, and the relationships as labelled lines.

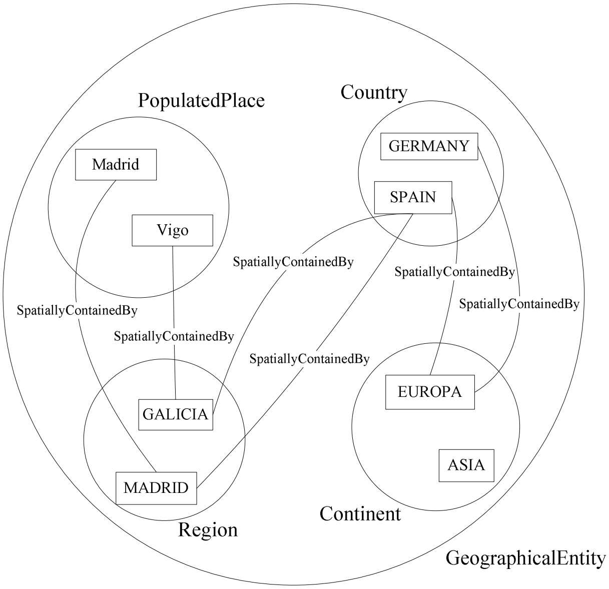

Fig. 1. Ontology instances

We build the ontology using a *Gazetteer* (in our test implementation we use *Geonames* [21]). However, Geonames (and Gazetteers in general) does not provide geometries for the location names other than a single representative point whereas our spatial index needs the real geometry of the location name (for example, the boundary of countries). Hence, we defined a *Geometry Supplier* service to obtain the geometries of those location names. As a base for this service we used the *Vector Map* (VMap) cartography [22]. The Geographic Space Ontology Service is composed of both the Gazetteer service and the Geometry Supplier service.

The spatial indexing stage comprises three steps. First, the system extracts *candidate location names* (words that are likely to be location names) from the text. The documents are parsed in order to discover the place names contained within. We use the *Natural Language Tool LingPipe* [23] to find the candidate location names. In the system prototype we use LingPipe trained with the MUC6 corpus (http://www.ldc.upenn.edu) labelled with locations, people and organizations. After the LingPipe processing, the module filters the resultant named entities selecting only the locations and discarding people and organization names.

In a second step, the candidate locations are processed in order to determine whether the candidates are real location names, and, in this case, to compute their geographic locations. There are some problems that can happen at this point: a location name can be ambiguous (*polysemy*), and there can be multiple names for the same geographic location. We developed a module, called Geographic Space Ontology Service, based on the ontology of the geographic space to geo-reference location names. This module returns for a candidate location name an *ontology graph* with the individual that represents the location name and all the individuals related by means of *spatiallyContainedBy* relationships. If the ontology does not have an individual for the candidate location name, it is discarded.

Finally, the third step consists in building the spatial index with the ontology graphs of the geo-referenced locations computed in the previous step together with references to the documents containing them. The spatial index is a tree composed by nodes that represent location names. The tree structure depends on the ontology that is used in the system. In the case of the ontology described previously, the nodes are connected by means of inclusion relationships (for instance, *Galicia* is included in *Spain*). In each node we store: (i) the keyword (a place name), (ii) the bounding box of the geometry representing this place, (iii) a list with the document identifiers of the documents that include geographic references to this place, and (iv) a list of child nodes that are geographically within this node. If the list of child nodes is very long, using sequential access is very inefficient. For this reason, if the number of children nodes exceeds a threshold, an R-Tree is used instead of a list.

This structure has two main drawbacks. First, the tree that supports the structure is possibly unbalanced penalizing the efficiency of the system. We present some experiments in Section 6 trying to prove that this is not a very important problem. Second, ontological systems have a fixed structure and thus our structure is static and it must be constructed *ad-hoc*.

4 System Architecture

Figure 2 shows our proposal for the system architecture of a geographic information retrieval system. The architecture can be divided into three independent layers: the index construction workflow, the processing services, and the user interfaces. The bottom part of the figure shows the index construction workflow, which, in turn, consists of three modules: the document abstraction module, the index construction module, and the index structure itself.

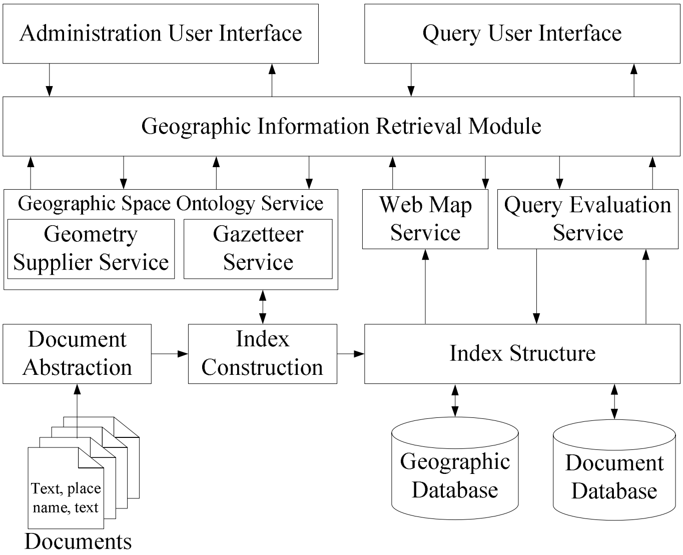

Fig. 2. System Architecture

The processing services are shown in the middle of the figure. On the left side, the *Geographic Space Ontology Service* used in the spatial index construction is shown. On the right side, one can see the two services that are used to solve queries. The rightmost one is the *query evaluation service*, which receives queries and uses the index structure to solve them. Section 5 describes the types of queries that can be solved by this service, as well as the algorithms that are used to solve these queries. The other service is a *Web Map Service* following the OGC specification [24] that is used to create cartographic representations of the query results. On top of these services a *Geographic Information Retrieval Module* is in charge of coordinating the task performed by each service to response the user requests.

The topmost layer shows the two user interfaces that exist in the architecture: the *Administration User Interface* and the *Query User Interface*. The administration user interface was developed as a stand-alone application and it can be used to manage the document collection. The *Query User Interface* interface was developed as a web application using the *Google Maps API* [25]. This user interface allows the user to indicate both the textual and the spatial aspects of queries. The spatial context can be introduced in three ways that are mutually exclusive: typing the location name, selecting the location name in a tree, and visualizing the spatial context of interest in the map.

5 Supported Query Types

The most important characteristic of an index structure is the type of queries that can be solved with it. Our index structure support three types of queries:

pure textual queries, pure spatial queries, and queries with a textual and a spatial component. In this last type, the spatial component can be given both as a location name or as a geographical area.

Pure textual queries such as "*retrieve all documents where the words* hotel *and* sea *appear*" can be solved by our system because a textual index is part of the index structure. Similarly, pure spatial queries such "*retrieve all documents that refer to the following geographic area*" can also be solved because the index structure is built like a spatial index. Each node in the tree is associated with the bounding box of the geographic objects in its subtree. Therefore, the same algorithm that is used with spatial indexes can be used with our structure.

Furthermore, the index structure that we propose can be used to solve queries that involve a textual and a spatial component. In this case, the textual index is used to retrieve the list of documents that contain the words, and the spatial index structure is used to compute the list of documents that reference the geographic area. The result to the query is computed as the intersection of both lists. In the case of queries such as "*retrieve all documents with the word* hotel *that refer to Spain*", our system uses the Geographic Space Ontology Service to discover that *Spain* is a geographic reference and then it uses the *place name hash table* to retrieve the index node that represents *Spain*. Thus, we save some time by avoiding a tree traversal.

Another improvement over text and spatial indexes is that our index structure can easily perform query expansion on geographic references because the index structure is built from an ontology of the geographic space. Consider the following query "*retrieve all documents that refer to Spain*". The query evaluation service will discover that Spain is a geographic reference and the place name index will be used to quickly locate the internal node that represents the geographic object *Spain*. Then all the documents associated to this node are part of the result to the query. Moreover, all the children of this node are geographic objects that are contained within Spain (for instance, the city of Madrid). Therefore, all the documents referenced by the subtree are also part of the result of the query. The consequence is that the index structure has been used to expand the query because the result contains not only those documents that include the term *Spain*, but also all the documents that contain the name of a geographic object included in Spain (e.g., all the cities and regions of Spain).

6 Experiments

In the previous section we showed that our structure has a qualitative advantage over systems that combine a textual index with a pure spatial index because query expansion can be performed directly with our index structure. Hence, our index structure supports a new type of query that cannot be implemented with a pure spatial index. However, unlike pure spatial index structures, our index structure is not balanced and therefore, the query performance can be worse. In this section we describe the experiments that we performed to compare our structure with other ones based on a pure spatial index. We used the TREC FT-91 (Financial Times, year 1991) document collection [26], which consists of 5,368 news documents. Then, we built two indexes over this collection: one using our index structure as described in this paper, and another one using a textual index and an R-Tree. Furthermore, we developed an algorithm to generate random spatial query windows based on the performance comparisons of the R*-Tree in [27]. We compared the structures with respect to four different query window areas, namely 0.001%, 0.01%, 0.1% and 1% of the world. We generated 100,000 random query windows for each area, and we averaged the computing time of each query execution. Table 1 shows the results of this experiment.

Table 1. Ontology-based index versus R-Tree

	Overall				**High density**				**Low density**			
Query area (%)	0.001	0.01	0.1	1	0.001	0.01	0.1	1	0.001	0.01	0.1	1
Our index	0.013	0.017	0.052	0.360	0.03	0.11	1.05	9.84	0.02	0.03	0.09	0.4
R-Tree	0.010	0.016	0.057	0.370	0.07	0.22	1.64	12.85	0.02	0.03	0.07	0.2

The first row of the table shows the results obtained with our structure (in milliseconds), and the second one shows the results obtained with the structure using an R-Tree. Both index structures have similar performance. The performance of our structure is a bit worse than the R-Tree when the query window is small but, surprisingly it is a bit better than the R-Tree when the query window is bigger. In order to explain this surprising result, we analyzed the performance in particular zones. We distinguished two relevant types of zones and we repeated the experiment generating random queries in both zones.

First, we studied the performance of the structures when the document density is high. In this case, the performance of our structure is higher than the R-Tree performance. We believe this is because our structure stores a list of documents for each location while the R-Tree uses a node for each document.

Then, we studied the performance when the document density is low. In this case, the R-Tree performance is better because the number of nodes in both structures is similar and the R-Tree is balanced whereas our structure may be unbalanced. For this reason, in the general case, when the query window is small the probability of that query window being in a high document density zone is small and, therefore, the R-Tree performance is better. However, when the query window is bigger that probability is higher and, therefore, the R-Tree performance is lower.

7 Conclusions and Future Work

We have presented in this paper a system architecture for an information retrieval system that takes into account not only the text in the documents but also the geographic references included in the documents and the ontology of the geographic space. This is achieved by a new index structure that combines a textual index, a spatial index, and an ontology-based structure. We have also presented how traditional queries can be solved using the index structure, and new types of queries that can be solved with the index structure are described and the algorithms that solve these queries are sketched. Finally, we performed some experiments that show that the performance of our structure is acceptable in comparison with index structures using pure spatial indexes.

Future improvements of this index structure are possible. We are currently working on the evaluation of the performance of the index structure, particularly we are performing experiments to determine the precision and recall. Moreover, *Toponym Resolution* techniques must be implemented to solve ambiguity problems when we geo-reference the documents. Another line of future work involves exploring the use of different ontologies and determining how each ontology affects the resulting index. Furthermore, we plan on including other types of spatial relationships in the index structure in addition to inclusion (e.g., adjacency). These relationships can be easily represented in the ontology-based structure, and the index structure can be extended to support them. Finally, it is necessary to define algorithms to rank the documents retrieved by the system. For this task, we must define a measure of spatial relevance and combine it with the relevance computed using the inverted index.

References

1. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison Wesley, Reading (1999)
2. Worboys, M.F.: GIS: A Computing Perspective. CRC, Boca Raton (2004)
3. Global Spatial Data Infrastructure Association: Online documentation (Retrieved, March 2008), http://www.gsdi.org/
4. Lieberman, M.D., Samet, H., Sankaranarayanan, J., Sperling, J.: STEWARD: Architecture of a Spatio-Textual Search Engine. In: Proceedings of the 15th ACM Int. Symp. on Advances in GIS (ACMGIS 2007), pp. 186–193. ACM Press, New York (2007)
5. Chen, Y.Y., Suel, T., Markowetz, A.: Efficient query processing in geographic web search engines. In: SIGMOD Conference, pp. 277–288 (2006)
6. Martins, B., Silva, M.J., Andrade, L.: Indexing and ranking in Geo-IR systems. In: GIR 2005: Proceedings of the 2005 workshop on Geogr. Inform. retrieval, pp. 31–34. ACM Press, New York (2005)
7. Gaede, V., Günther, O.: Multidimensional access methods. ACM Comput. Surv. 30(2), 170–231 (1998)
8. Amitay, E., Har'El, N., Sivan, R., Soffer, A.: Web-a-where: geotagging web content. In: SIGIR 2004: Proceedings of the 27th ACM SIGIR, pp. 273–280. ACM, New York (2004)
9. Rauch, E., Bukatin, M., Baker, K.: A confidence-based framework for disambiguating geographic terms. In: Proceedings of the HLT-NAACL 2003 workshop on Analysis of Geogr. references, Morristown, USA, pp. 50–54. Association for Computational Linguistics (2003)

10. Jones, C.B., Abdelmoty, A.I., Fu, G.: Maintaining ontologies for geographical information retrieval on the web. In: Meersman, R., Tari, Z., Schmidt, D.C. (eds.) CoopIS 2003, DOA 2003, and ODBASE 2003. LNCS, vol. 2888, pp. 934–951. Springer, Heidelberg (2003)
11. Jones, C.B., Abdelmoty, A.I., Fu, G., Vaid, S.: The SPIRIT Spatial Search Engine: Architecture, Ontologies and Spatial Indexing. In: Egenhofer, M.J., Freksa, C., Miller, H.J. (eds.) GIScience 2004. LNCS, vol. 3234, pp. 125–139. Springer, Heidelberg (2004)
12. Vaid, S., Jones, C.B., Joho, H., Sanderson, M.: Spatio-Textual Indexing for Geographical Search on the Web. In: Bauzer Medeiros, C., Egenhofer, M.J., Bertino, E. (eds.) SSTD 2005. LNCS, vol. 3633, pp. 19–36. Springer, Heidelberg (2005)
13. Fu, G., Jones, C.B., Abdelmoty, A.I.: Ontology-Based Spatial Query Expansion in Information Retrieval. In: Meersman, R., Tari, Z. (eds.) OTM 2005. LNCS, vol. 3761, pp. 1466–1482. Springer, Heidelberg (2005)
14. Zhou, Y., Xie, X., Wang, C., Gong, Y., Ma, W.Y.: Hybrid index structures for location-based web search. In: Proceedings of CIKM 2005, pp. 155–162. ACM, New York (2005)
15. Hariharan, R., Hore, B., Li, C., Mehrotra, S.: Processing Spatial-Keyword (SK) Queries in Geographic Information Retrieval (GIR) Systems. In: Proceedings of the 19th Int. Conf. on Scientific and Statistical Database Management (SSDBM 2007). IEEE Computer Society, Los Alamitos (2007)
16. Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition 5(2), 199–220 (1993)
17. Gruber, T.R.: Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In: Guarino, N., Poli, R. (eds.) Formal Ontology in Conceptual Analysis and Knowledge Representation, Deventer, The Netherlands. Kluwer Academic Publishers, Dordrecht (1993)
18. Dellis, E., Paliouras, G.: Management of Large Spatial Ontology Bases. In: Proceedings of the Workshop on Ontologies-based techniques for DataBases and Information Systems (ODBIS) of the 32nd Int. Conf. on Very Large Data Bases (VLDB 2006) (September 2006)
19. Apache: Lucene (Retrieved, March 2008), `http://lucene.apache.org`
20. World Wide Consortium: Owl web ontology language reference (Retrieved March, 2008), `http://www.w3.org/TR/owl-ref`
21. Geonames: Gazetteer (Retrieved, March 2008), `http://www.geonames.org`
22. National Imagery and Mapping Agency (NIMA): Vector Map Level 0 (Retrieved, March 2008), `http://www.mapability.com`
23. Alias-i: LingPipe, Natural Language Tool (Retrieved, March 2008), `http://www.alias-i.com/lingpipe/`
24. Open GIS Consortium, Inc.: OpenGIS Web Map Service Implementation Specification. OpenGIS Project Document 01-068r3, Open GIS Consortium, Inc. (2002)
25. Google: Google Maps API (Retrieved, March 2008), `http://code.google.com/apis/maps/`
26. National Institute of Standards and Technology (NIST): TREC Special Database 22, TREC Document Database: Disk 4 (Retrieved, March 2008), `http://www.nist.gov/srd/nistsd22.htm`
27. Beckmann, N., Kriegel, H.P., Schneider, R., Seeger, B.: The R*-tree: an efficient and robust access method for points and rectangles. SIGMOD Rec. 19(2), 322–331 (1990)