A Temporal Search Engine to Improve Geographic Data Retrieval in Spatial Data Infrastructures

Fabio Gomes de Andrade[1], Cláudio de Souza Baptista[2] and Ulrich Schiel[2]
[1]*Department of Informatic, Federal Institute of Education, Science and Technology of Paraíba, Cajazeiras, Brazil*
[2]*Department of Systems and Computing, University of Campina Grande, Campina Grande, Paraíba, Brazil*

Keywords: Geographical Information Retrieval, Spatial Data Infrastructures, Temporal Search Engine.

Abstract: Recently, spatial data infrastructures have become an important solution to ease the finding of geographical data offered by different organizations. Nevertheless, the catalog services provided by these infrastructures still have some important drawbacks that limit the geographic information retrieval based on temporal constraints. Examples of these drawbacks include the lack of both a more detailed description of temporal information, and ranking. Aiming to overcome these limitations, this paper describes a new temporal search engine for solving feature type retrieval offered by catalog services. To reach this goal, our search engine is based on a model that stores temporal information about each service and its respective feature types described in the SDI catalog service. Moreover, the paper proposes a temporal ranking metric to evaluate the relevance of each feature type retrieved for the user's query.

1 INTRODUCTION

Recently, spatial data infrastructures (SDIs) have become an important solution to ease the finding of geographical data offered by different organizations (Williamson et al., 2003). SDIs usually offer catalog services, which can be used by both providers and clients. Providers use this service to announce their resources, while clients use it to find the data of their interest.

The current catalog services improve geographic data retrieval, but still have serious limitations. One of these limitations is the low support to temporal searches. The time-based searches offered by the current catalogs are performed using attributes such as temporal extension and creation/modification date of the resources. Nevertheless, the values of these attributes are often omitted; or contain imprecise information. Such characteristic considerably reduces the quality of temporal searches. Other important drawback is the lack of mechanisms to evaluate the importance of each resource retrieved from a user's query.

Aiming to overcome these limitations, in this paper we propose a new search engine for solving temporal queries in SDIs. The proposed solution offers two contributions. The first one consists of a model that improves the description of the temporal features of the services described in the SDI catalog service. The second contribution consists in the development of a ranking mechanism that is based on the temporal features of each feature type and ideas from the classical information retrieval. Such ranking is used to evaluate how relevant each feature type is for the user's query, considering only the temporal dimension.

It is important to keep in mind that geographical data are characterized by three dimensions: space, theme and time. The solution described in this paper approaches only the time dimension. However, this solution has been integrated to a broader search engine, which is able to solve queries with spatial, thematic and temporal constraints. Details about the semantic ranking implementation can be found in (Andrade and Baptista, 2011).

The remaining of this paper is organized as follows. Section 2 describes how the current SDIs offer temporal information. Section 3 focuses on the model used to represent temporal information. Section 4 presents the temporal ranking measurement. Section 5 discusses the implementation and the experimental evaluation. Section 6 summarizes the main related works. Finally, in section 7, we conclude the paper.

56 Gomes de Andrade F., de Souza Baptista C. and Schiel U..
A Temporal Search Engine to Improve Geographic Data Retrieval in Spatial Data Infrastructures.
DOI: 10.5220/0004001800560065
In *Proceedings of the 14th International Conference on Enterprise Information Systems* (ICEIS-2012), pages 56-65
ISBN: 978-989-8565-10-5
Copyright © 2012 SCITEPRESS (Science and Technology Publications, Lda.)

2 TEMPORAL INFORMATION IN SDI

In order to ease the standardization and access to their geographical data, many SDIs are being implemented as a set of services (Bernard and Craglia, 2005). In such infrastructures, the datasets offered by a provider are commonly supplied as a set of feature types. These feature types, in turn, are encapsulated and delivered for the users through services standardized by the Open Geospatial Consortium [OGC], such as Web Map Service (WMS) (OGC, 2004) and Web Feature Service (WFS) (OGC, 2005).

When a provider registers a service in the SDI catalog, it must provide metadata that describe different features of the dataset offered by the service. Part of this information is related to the temporal extension, where the provider must inform the time interval with respect to the dataset. The way as this information is described depends on the metadata standard adopted by the infrastructure. In the ISO 19115 metadata standard, the temporal reference of a resource is usually described through a temporal interval. Such interval is defined by two attributes called *beginPosition* and *endPosition*, defining, respectively, the initial and final limits. The value of these attributes is commonly described in the ISO 8601 format.

One of the causes that limit the resolution of temporal queries in the present SDIs is the fact that the values of attributes that describe the temporal extension of the data are often omitted by the provider. In this case, the only information offered that is related to time is the creation/modification date of the metadata record, which do not describe the temporal extension with precision.

Another limitation is related to the details supplied during the registration. Presently, most geographic data providers create a single metadata record to describe their dataset. Hence, only one temporal extension is defined to characterize all the feature types offered by a service. Figure 1 shows two metadata records from different providers, called M_1 and M_2. Each record describes the feature types offered by a service.

In the service described by M_1, there are feature types covering the periods of 2000, 2002 and 2005. In this record, it was defined that the temporal extension (TE) of the service is the interval between 2000 and 2005, which corresponds to the smallest interval covering all of its feature types. Such a situation leads to two kinds of disadvantages. In order to understand the first one, let us consider a query where the user looks for feature types concerning the year of 2003. In this case, as the extension of M_1 intersects the period defined in the query, the record ends up being retrieved, though it does not offer any feature type concerning the period defined in the request. The second disadvantage occurs because the catalog service always retrieves the service as a whole. So, the user is in charge of accessing the service and identifying the feature types that satisfy the criteria defined in the query. This task can be, often, tedious and time consuming, since many services offer a large number of feature types.

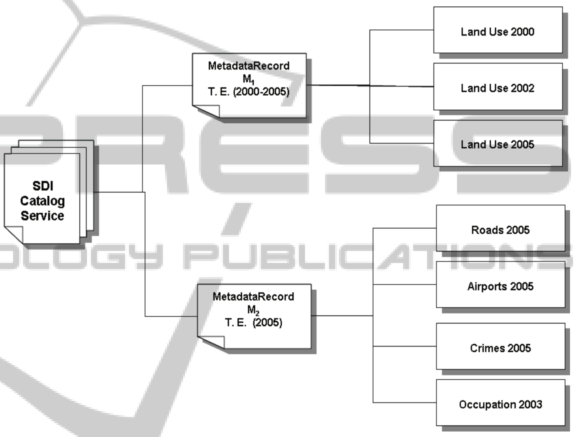


Figure 1: Example of services temporal description.

Other problems concerning the resolution of temporal queries occur due to inconsistencies between the temporal extension defined in the metadata record and the temporal extensions concerning the feature types. Observing Figure 1 again, it is possible to notice that the record M_2 defines the interval 2005 as its temporal extension, though it also has a feature type concerning another temporal interval. This kind of situation occurs because many providers use as temporal extension just the period associated to most of its feature types. So, in the case the user performs a query for feature types concerning the period of 2003, the catalog service will not retrieve the service described by M_2, though the service has data which satisfy the criteria defined in the request.

Besides the limitations previously described, another problem of the present catalog services is that they assume that all the resources that satisfy the selection criterion defined in the query have the same relevance for the user. So, a resource that covers the whole interval requested by the user is considered as relevant as one that covers only a part of the requested interval. This makes the possibly more relevant services to be presented later to the user. This is an undesired characteristic, especially in queries which return a large number of results. The main consequence is that, in these queries, the user may lose time in trying to find the service having the most relevant information or, in worse cases, this user my end up not evaluating the resource.

3 DESCRIBING TEMPORAL INFORMATION

The first stage in the development of our search engine consists in defining a model to represent temporal information of the services offered by the SDI. In order to fulfill requirement R_1, the designed schema stores the temporal information of each service and of each feature type that it offers. Figure 2 presents the conceptual schema.

It is important to have in mind that the simplicity of the schema is due to the fact that most part of the service information keeps being stored in the metadata record. So, our schema stores just the information needed to resolve the temporal queries.

The temporal extensions of a service and of its feature types are defined through the attributes *beginTime* and *endTime*, present in their respective entities. Both attributes are represented as timestamps. Moreover, the model stores some descriptive attributes of each service and of each feature type, such as name, title and textual description. These attributes are shown to the user during the exhibition of the query result. After evaluating the information shown by these attributes, the user may request a complete view of the record that describes the service offering the feature type of interest. Such retrieval is performed by the attribute *metadataIdentifier*, which keeps a reference for the service metadata record.

After defining the model to be used for data representation, we created a methodology to extract temporal information from the service and feature types. In order to facilitate the reader's understanding, this process will be referred to as *temporal annotation* throughout this paper.

3.1 Services Annotation

The first stage of the process of extracting temporal information consists in identifying the temporal interval covered by the service. The information is obtained through the processing of information in the metadata record of the service.

The service annotation is done through the value of the attributes that define its temporal extension. When the values of these attributes are provided by the metadata record that describes the service, they are retrieved and normalized to a temporal interval. This interval is then defined as the service temporal extension. Another attribute verifies the service update frequency. The verification is intended to check whether the service is continuously updated. If so, the model considers the service persistent and, consequently, it has no final limit. In our solution, we consider persistent just the services with the following values for update frequency: *annually, continually, daily, monthly* and *weekly*.

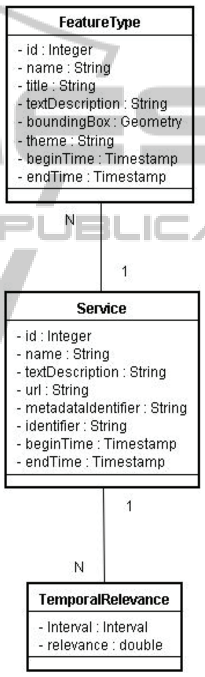


Figure 2: Conceptual schema.

When the temporal extension of the service is not present in its metadata, its temporal extension is identified through the values of other attributes. The attributes are queried in a priority order. In the case an attribute contains temporal information, its value is extracted, standardized and associated to the service, ending the annotation process. In the opposite case, the next attribute is queried and the process is repeated until all attributes are checked. Currently, the attributes queried in order to obtain temporal information of a service, in priority order, are: keywords, title and textual description. Finally, in the case temporal information is not found in any of these attributes, the creation date of the metadata record is used as temporal reference for this service.

Among the attributes queried during the annotation process, some are represented directly as dates, as the temporal extension and the inclusion date of the record. When the temporal information is obtained through these attributes, it is necessary to convert the attribute value into a temporal interval, in the format used by the data schema.

The date standardization process depends on the format of the temporal information. In the case the attribute value is a simple date (for instance, 1/10/2010), its conversion to an interval can be done in two forms, depending on its update frequency. If the service is continuously updated, the date is converted into a persistent interval, which means that the final limit corresponds to the moment at which the query is performed. Otherwise, the value is standardized to an interval containing the limits of the value found for the attribute. The granularity of this interval (day, month, year) is identical to the granularity of the attribute value. For example, if the value represents a day, the generated interval will represent a day too. The same occurs for attributes that represent a month or a year.

While attributes concerning temporal extension and inclusion date of the record already supply the values in the form of dates, some of the queried attributes, such as service name and textual description, are offered as a set of strings. When these attributes are queried, their text values must be processed in order to extract temporal information. This processing, which is occurs with use of machine learning techniques, is performed in two stages.

In the first step, the text corresponding to the attribute value is processed for analysis of the parts of the speech. This stage is intended to identify and classify the radical of the elements that appear in the text. This task is done by a framework called *TreeTagger*.

In the second stage, the result of the previous stage is processed in order to find temporal expressions. This information can be found by both numerical values, such as dates, and textual elements, such as the words *today, yesterday and tomorrow*. As the result of this stage, an XML file containing all the temporal expressions identified in the text is generated. This file is coded in the *TimeML* standard (Pustejovsky et al., 2003), an XML-based standard for specification of temporal expressions in documents. This task is carried out by a framework called *Heideltime*.

The processing made by *Heideltime* automates the recognition of temporal expressions. However, each identified expression is treated and annotated individually, with no relationship among them. So, the file generated by this framework must be processed, and the temporal information that is found must be converted into a temporal interval. The standardization of temporal annotations found in a text depends on the number of annotations found. If the file has just one temporal annotation, the feature type interval corresponds to the smallest interval that covers all the temporal information found. Finally, the absence of temporal annotations means that no temporal element was found in the analyzed text. This situation indicates that the temporal reference cannot be obtained through this attribute.

3.2 Feature Types Annotation

Differently from the annotation of services, the temporal annotation of a feature type is performed with basis on the information contained in the document describing the capabilities of the service. Such a document is obtained by invoking the operation *getCapabilities* of the service being annotated.

An important characteristic of the capabilities document is that, differently from the metadata record, it has no specific attribute to define temporal information of the feature types offered by the service. So, the temporal annotation of these elements must be done through the identification of temporal expressions present in the values of some attributes. In order to perform this task, for each feature type, their keywords, titles and textural descriptions are queried following the priority order.

Since all attributes used for feature types annotation are textual, the temporal information contained in these elements must be extracted through the processing of the text corresponding to their values. The procedure adopted to perform this task is the same used in the annotation of services. The values obtained after this process are used as the temporal extension of the feature type that is being processed. In the case the temporal extension of the feature type cannot be obtained from any of the verified attributes, we assume that its value is the same one obtained for its respective service.

4 TEMPORAL RANKING

After defining how the temporal information will be retrieved from the services and stored in the database, we developed a search engine for retrieval of feature types that meet a certain temporal constraint. To implement it, we defined a metric that evaluates how relevant each feature type is for the query. Such a metric is computed from other two measurements: the degrees of overlap and of temporal relevance.

4.1 The Degree of Overlap

The first measurement used to evaluate the temporal ranking is the *degree of overlap*, which evaluates the similarity between the temporal interval requested in the query and the temporal interval covered by the feature type under evaluation. This metric is computed by means of the Tversky equation (Tversky, 1977). This equation was chosen because it considers, during the evaluation of similarity between objects, the characteristics that they may have or not in common. Let t_1 be the temporal interval defined in the user's query and t_2 the interval associated to the feature type under evaluation. Then, the *degree of overlap* between them is computed by equation 1.

$$od(t_1,t_2) = \frac{|t_1 \cap t_2|}{|t_1 \cap t_2| + \alpha|t_1 / t_2| + (1-\alpha)|t_2 / t_1|} \quad (1)$$

where:

- $| t_1 \cap t_2 |$ represents the extension, in milliseconds, of the intersection between the intervals t_1 and t_2;
- $| t_1 / t_2 |$ represents the extension of the interval in t_1, but not in t_2;
- $| t_2 / t_1 |$ represents the extension of the interval in t_2, but not in t_1;
- The constant α represents the weight that the complement of the interval t_1 has to the evaluation of the overlap between the intervals. Presently, the value 0.9 is used for this constant. To determinate this value, we used a technique called weighting (Fox and Shaw, 1993). To estimate the value of α we used the Pearson correlation coefficient.

4.2 The Temporal Relevance

The second metric used to evaluate the temporal ranking of a feature type is the degree of *temporal relevance*. As in the classical information retrieval, this metric evaluates how relevant a certain temporal interval is to a certain service (Baeza-Yates and Ribeiro-Neto, 1999). Hence, when two feature types offered by different services have the same degree of overlap (or close values) with respect to the user's query, the model prioritizes feature types offered by services whose temporal extension has higher relevance.

In order to compute the relevance of a temporal interval to the service, it is necessary to evaluate first the frequency in which this interval occurs in the service. This metric is called *raw frequency (temp_f)* and is computed by comparing the temporal interval under evaluation with the temporal extension covered by each feature type offered by the service.

During the development of this metric, we evaluated three forms to compute the frequency of a temporal interval *t* in a service *S*. The first solution consisted in computing the number of associated feature types whose temporal extension was identical to *t*. Despite being easier to compute, this kind of approach did not consider that the intervals were different from *t*, but that they completely contained that interval, and also did not consider that some parts of the interval were present in intervals that intersected it. In the second approach, the frequency was computed with basis on the number of intervals that were identical or totally covered the interval under evaluation. The disadvantage of this solution is that it does not consider the overlap between the evaluated interval and the other intervals that do not cover it totally.

In the third approach, which ended up being adopted for the proposed metric, the frequency is computed by summing the *degree of overlap* between t and the temporal interval associated to each feature type offered by the service. So, all the presences (total or partial) on the interval t in the temporal interval associated to each feature type are considered when evaluating the frequency of this interval. Equation 2 describes the computation of this frequency. In this equation, t represents the temporal interval under evaluation, while S is the service to which the relevance of t is being computed. Moreover, Ti_{temp} represents the temporal extension of a feature type T offered by the service, and n represents the number of feature types offered by the service.

$$temp\_f(t,S) = \sum_{i=1}^{n} overlap(Ti_{temp}, t) \quad (2)$$

After computed, the value of the *raw frequency* of the temporal interval is used to compute its *normalized frequency* (*temp_tf*). This frequency is computed by the proportion between the *raw frequency* and the number of feature types offered by the service (Equation 3). As in the previous equation, t represents the temporal interval which is under evaluation and S represents the service to which the relevance will be computed.

$$temp\_tf(t,S) = \frac{temp\_f(t,S)}{n} \quad (3)$$

Another variable used to evaluate the degree of relevance of a temporal interval is the *inverse frequency* (*temp_isf*). Its objective is to evaluate how important a temporal interval is to the whole set of services offered by the SDI. The value of the *inverse frequency* (Equation 4) is computed by the proportion between the number of services offered and the number of services in which the interval is totally covered by at least one feature type.

$$temp\_isf(t) = \log \frac{N}{ni} \tag{4}$$

After computing the *normalized frequency* and the *inverse frequency*, their values are used to determine the degree of temporal relevance of an interval to the service. This *degree of relevance* is computed through the product of both frequencies, as in Equation 5.

$$tr(t,S) = temp\_tf(t,S) * temp\_isf(t) \tag{5}$$

4.3 The Temporal Similarity

Once computed, the values of the *degrees of overlap* and *temporal relevance* are combined to determine the *temporal ranking* of a feature type. The value of this metric is used to classify and sort the feature types that are retrieved form a user's query.

Given a temporal interval t defined in the user's query and a feature type T offered by the SDI, the *temporal ranking* of T is obtained through Equation 6. In this equation, T_T represents the temporal interval covered by T, while S represents the service that offers this feature type.

$$ranking(t,T) = w_1 * od(t,T_T) + w_2 * tr(T_T,S) \tag{6}$$

where:

- *ranking* represents the temporal similarity between a temporal interval t defined in the user's query and a feature type T being evaluated;
- *od* represents the *degree of overlap* between the interval defined in the query (t) and the interval covered by the feature type under evaluation (T_T);
- *tr* represents the *degree of relevance* of the temporal interval associated to the feature type under evaluation to its respective service;
- w_1 and w_2 represent the weights that the degree of overlap and the degree of temporal relevance has for the calculation of the degree of temporal similarity. Each weight must have a value between 0 and 1, and their sum must always be equal to 1. Presently, we use values of 0.84 and 0.16, respectively, for w_1 and w_2. These values were defined using the same statistical technique used to determinate the weights in Equation 1.

5 IMPLEMENTATION AND EVALUATION

Once defined the temporal ranking metric, our temporal search engine was implemented. This section first focuses on the implementation issues. After, the results obtained from the experimental evaluation are presented.

5.1 Implementation Issues

After defining the model used to represent information and the metric used to retrieve this information, we implemented a prototype for our search engine. This engine was incorporated to a tool called SESDI (Semantic-Enabled Spatial Data Infrastructures) (Andrade and Baptista, 2011), used for discovery of geographic data in SDIs. The architecture used for its implementation is comprised of two main subsystems: the data acquisition module and the query resolution module. The first module contains the components responsible for the extraction of the temporal data from the services registered in the SDI and from their respective feature types. The query resolution module, in turn, has the components that use the temporal ranking, describe is the previous section, to resolve temporal queries.

5.1.1 The Data Acquisition Process

The data acquisition process consists in collecting temporal data that will be used during the query resolution process. The data acquisition process is performed periodically, in order to find new services included and/or updated, keeping the database updated.

The process consists in obtaining information about new services. For this, the acquisition module calls the *getRecords* operation of the registration service of the infrastructure. This call has a filter that selects only the services whose inclusion/modification dates are more recent than the last verification made by the module. Each service retrieved in the first stage is then processed in order to make its temporal annotation.

The processing of each service is subdivided into several stages. The first one consists of extracting the temporal extension covered by the service. After obtaining this information, the module calls the *getCapabilities* operation of the service to obtain a document with information about the feature types offered by this service. After this, each feature type is processed to make its temporal annotation.

After extracting the temporal information from the service and from its feature types, the next stage consists in using the obtained information to generate the temporal relevance data of the service. For this, the module computes the temporal relevance of each temporal interval associated to at least one of the feature types offered by the service.

In the last stage of the data acquisition process, the information obtained after the extraction of temporal information and generation of temporal relevance of the service are made persistent in the database of the prototype. In this process, occurs the storage of the information about geographic data services and their feature types, with their temporal information, besides temporal relevance data generated for each service. Presently, these data are made persistent in a database implemented in the DBMS PostgreSQL/PostGIS.

5.1.2 The Query Resolution Process

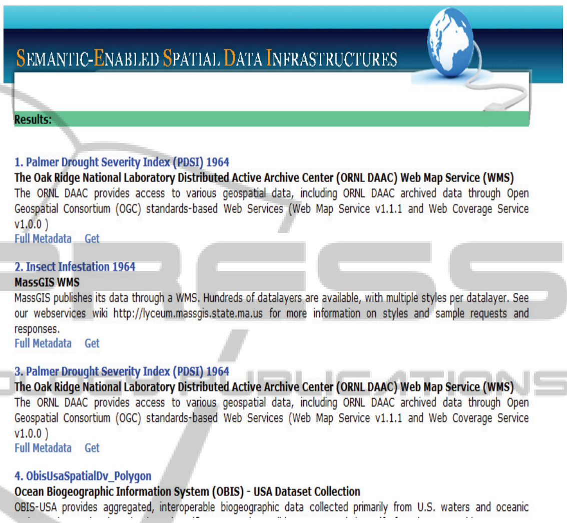
The query module is responsible for the resolution of temporal queries. All queries are performed through a set of dynamic web pages featured by the tool. Each requisition received by the query module has as input parameter a temporal interval of the user's interest. The processing of queries is divided in four steps: temporal filtering, matchmaking, relevance filtering and ordering.

In the temporal filtering step, the search tool selects, among all the feature types recorded in the database, those whose temporal expression intersects the interval defined in the query. This task is done through a simple query in the SQL database. The result of this step is a set containing all the feature types that meet this constraint.

The second step consists in using the temporal ranking to compute the relevance of each feature type retrieved in the first stage. During the matchmaking process, persistent intervals end up receiving as final limit the timestamp value corresponding to the time at which the query was requested. At the end of this step, for each retrieved result, the result obtained for the temporal ranking is associated.

In many situations, temporal queries may return results with very low relevance for the query. This characteristic, in some situations, may be undesired, especially during the processing of queries that return a large number of results. In order to avoid such a situation, the user may define a minimum threshold at the moment of the query. When this happens, the third step in the processing of a query consists in removing from the final result all feature types whose relevance is below this threshold.

Finally, the last step consists in organizing the remaining results according to their relevance values. For this, a sorting algorithm is executed, in order to list the retrieved feature types in descendant relevance order. Figure 3 shows the result of a requisition for historic feature types concerning the year of 1964.


SEMANTIC-ENABLED SPATIAL DATA INFRASTRUCTURES

Results:

1. Palmer Drought Severity Index (PDSI) 1964
The Oak Ridge National Laboratory Distributed Active Archive Center (ORNL DAAC) Web Map Service (WMS)
The ORNL DAAC provides access to various geospatial data, including ORNL DAAC archived data through Open Geospatial Consortium (OGC) standards-based Web Services (Web Map Service v1.1.1 and Web Coverage Service v1.0.0)
Full Metadata Get

2. Insect Infestation 1964
MassGIS WMS
MassGIS publishes its data through a WMS. Hundreds of datalayers are available, with multiple styles per datalayer. See our webservices wiki http://lyceum.massgis.state.ma.us for more information on styles and sample requests and responses.
Full Metadata Get

3. Palmer Drought Severity Index (PDSI) 1964
The Oak Ridge National Laboratory Distributed Active Archive Center (ORNL DAAC) Web Map Service (WMS)
The ORNL DAAC provides access to various geospatial data, including ORNL DAAC archived data through Open Geospatial Consortium (OGC) standards-based Web Services (Web Map Service v1.1.1 and Web Coverage Service v1.0.0)
Full Metadata Get

4. ObisUsaSpatialDv_Polygon
Ocean Biogeographic Information System (OBIS) - USA Dataset Collection
OBIS-USA provides aggregated, interoperable biogeographic data collected primarily from U.S. waters and oceanic

Figure 3: Temporal query result.

5.2 Experimental Evaluation

After implementing our temporal search engine, an experimental evaluation was performed, in order to compare its performance to that of the present catalog service. To make this validation, the catalog service of the North-American SDI was used as case study. The information of this service was collected and processed to perform the temporal annotation of each service and their respective feature types. The results obtained after this processing were stored in the database of the search engine. Presently, this database has 103 services and 12,914 feature types.

During the validation process, several queries were made for different time intervals. For each temporal interval used, two queries were performed. The first query was performed in the catalog service, and retrieved any record whose temporal extension intersected the interval defined in the query or whose publication date intersected the query interval (in the case of records with no temporal extension value defined). The second query was performed using the SESDI tool, which retrieved all the feature types whose temporal extension was intersected by the interval defined in the requisition. The results of both queries were used to compare the performance of these two approaches. This comparison was performed according to recall and precision metrics. Recall corresponds to the proportion between the number of relevant results retrieved and the total of existing relevant results. Precision, in turn, is obtained through the proportion between the number of relevant results retrieved and the total of retrieved results.

5.2.1 Recall Evaluation

The experiments results were analyzed in two steps. In the first one, we made the evaluation at service level. This evaluation was intended to check the impact of our solution with respect to the number of different services retrieved by a query. This verification allows us to observe the number of services that have at least one feature type which is relevant to the query, but which end up not being retrieved by the catalog service. Figure 4 shows a graphic obtained through the comparison of results with respect to coverage. This graphic shows the performance of both SESDI and catalog service along the queries that have been executed during the evaluation process. Moreover, axis x represents the queries, while axis y represents the obtained performance for each approach.

The analysis of Figure 4 allows us to see that the model used by SESDI led to a big improvement in the coverage of the queries. The model used by the tool obtained a mean recall of 88.45%, while the catalog service had a mean recall of 45.17%. This difference can be explained by the fact that our search engine stores temporal information of services and feature types, while the catalog service makes queries only on information stored at services level. This difference allows our search engine to retrieve any services that offer at least one feature type whose temporal extension matches the constraints defines in the query, even if the temporal extension of the service does not meet the criteria defined in the query. This characteristic is impossible for the catalog service. Moreover, the large number of services that do not have a defined value for the temporal extension contribute to this difference, since the creation and modification dates of the metadata do not express this information precisely.

In the second step of the validation process, the two approaches were compared through the number of retrieved feature types. This evaluation is intended to obtain an overview of the number of feature types that are not retrieved due to the limitations of the present catalog services, as well as measuring how much the model used by our search engine improves the retrieval of this kind of information. The recall comparison of the two approaches with respect to the retrieval of feature types is shown in the graphic of Figure 5.

Figure 5 shows that, when the recall of the queries is compared at feature type level, the performance difference between the two approaches is still high. In this kind of evaluation, the model used by our search engine presented a mean recall of 95.56%, while the catalog service obtained a mean recall of 43.78%. The reasons that led to this difference are the same that cause the recall difference with respect to the retrieval of services. However, the large number of feature types offered by some services makes the difference between the performances of both approaches to be still bigger than what happens in the comparison at services level.

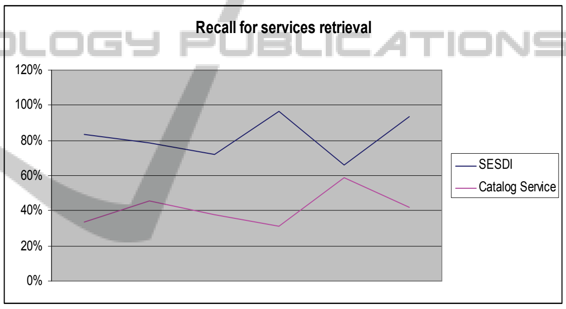


Figure 4: Graph of recall for services retrieval.

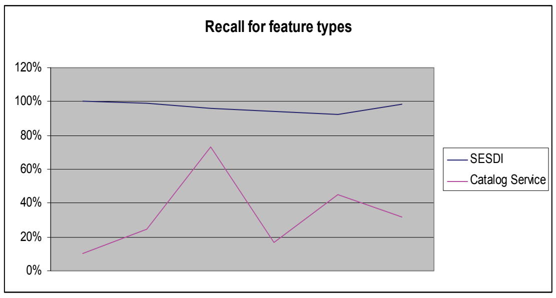


Figure 5: Graph of recall for feature types retrieval.

5.2.2 Precision Evaluation

Besides recall, the approaches were compared with respect to precision. Figure 6 shows the comparison of precision between the two approaches with respect to the number of services retrieved by each solution. The results show that the model used by SESDI had a better performance in some queries, while the catalog service achieves more precise results in some requisitions. The mean precision of the catalog service was of 93.72%, while SESDI achieved a mean precision of 89.48%. The analysis of the results shows that the precision loss of the SESDI is not caused by the model used by its search engine, but is due to the performance of the temporal annotation process. While the catalog service performs its searches with basis on information of the catalog service, which are manually provided, the temporal search engine used by SESDI performs its queries with basis on information extracted manually during the temporal annotation process. This process is subject to errors, since some temporal expressions may be misinterpreted when extracted from textual attributes.

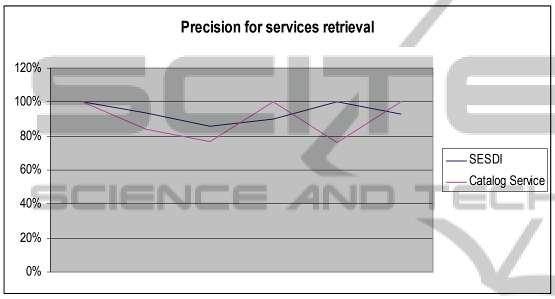


Figure 6: Graph of precision for services retrieval.

When the precision of the approaches is compared at feature type level, the performance of the two approaches is similar to the performance at services level. The graphic obtained for this comparison is shown in Figure 7. While the catalog service had a mean precision of 92.14%, SESDI achieved a mean precision of 88.95%.

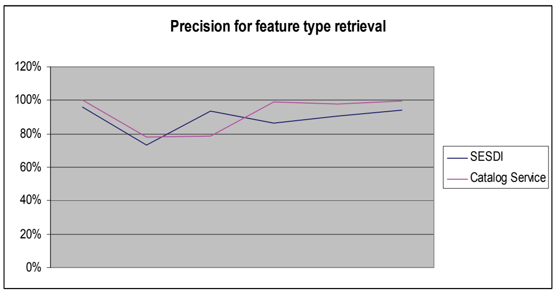


Figure 7: Graph of precision for feature types retrieval.

In general, the results obtained through the validation prove the feasibility of the model proposed in this paper. The main element that leads to this conclusion is that the model increases considerably the coverage of temporal queries, allowing the retrieval of many feature types even if the temporal description of their respective service is not supplied or not described in a consistent fashion.

6 RELATED WORK

The use of the temporal dimension to improve the information retrieval has been addressed by many studies over the years. Hübner and Visser (2003) proposed a solution during the implementation of the BUSTER project (Vögele et al., 2003). In that work, the temporal extension of each resource is represented through temporal periods. The limits of these periods can be described in precise, persistent or fuzzy forms, or can be defined with respect to other periods. During the information retrieval process, an algorithm based on Allen's temporal logic (Allen, 1998) is used to infer the semantic relationships between each period. The use of logic and inference allows the development of more powerful search tools. Nevertheless, this kind of approach does not offer a ranking. Moreover, the high computational cost of this kind of solution hinders its application to collections with a large amount of data.

A ranking-based solution was developed by Alonso, Gertz and Baeza-Yates (2006). In their work, documents are grouped in clusters into a timeline according to their temporal information. Inside each cluster, the documents are organized by a ranking, which is obtained through the metrics *tf-idf*, with respect to the matching of the text in the document and the text in the query. Manica et al., (2010) developed a search engine which enables the retrieval of temporal information in XML documents. In that work, the processing of queries is performed in two stages: a keyword matching and a temporal query, which is applied to the nodes that are closer to those retrieved in the previous stage. In both works, the ranking used to organize the documents does not consider the temporal extension of each resource.

Another ranking-based solution was developed by Jin et al., (2010). In that work, the keywords that form a document are associated to temporal intervals, which are obtained from expressions contained in the document. For each combination formed by a keyword and a temporal interval, a ranking value is computed through *tf-idf* techniques. The disadvantage of that work is that its ranking considers just the importance of the keywords, not considering the relevance of each temporal expression found in the document.

Another work that addresses temporal information retrieval in documents was developed by Strötgen and Gertz (2010). In that work, spatiotemporal information of a document is extracted through the processing of the text and can be explored by users after a query. However, the means to evaluate the temporal ranking of each document are not supplied.

The analysis of the above studies shows that the temporal information retrieval is still an open problem. This analysis also shows that many studies explore the temporal dimension during the resolution of queries, but do not use (or use superficially) this information to establish the ranking of the retrieved results. This highlights the need for a more specific ranking, generated from a deeper analysis of this kind of information. Moreover, we can notice the lack of effective solutions to retrieve temporal data in the geospatial domain. This limitation, allied to the key importance that the time represents for this domain, highlights the importance of the work presented in this paper.

7 CONCLUSIONS

The temporal dimension has great importance for the retrieval of geographic data. However, the retrieval of geographic data with basis on temporal criteria is still a hard task for the present SDIs. The absence of a detailed description of the temporal extension of the services and the lack of a temporal ranking are some of the characteristics that cause this limitation.

Aiming to overcome those limitations, this paper described a new temporal search engine. The main contributions consist in the development of a new model that improves the description of the temporal extension at service and feature type levels, and the development of a ranking for the feature types retrieved during a query.

Some future works still should be undertaken to improve our research. An important task to be developed consists of extending our approach to handle others types of temporal information, such as imprecise temporal information. Besides, we should improve the integration of our temporal search engine with the other similarity metrics. This task will enable us to evaluate the performance of our tool when solving queries concerning more then one dimension. Finally, other important improvement to be undertaken consists of integrating our solution to the current catalog service interface.

REFERENCES

Allen, J. F., (1998). Maintaining Knowledge about Temporal Intervals. *Communications of the ACM,* 26(11), 832-843.

Alonso, O., Gertz, M. and Baeza-Yates, R. A., (2006). Clustering of search results using temporal attributes. In *29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM Press.

Andrade, F.G. and Baptista, C. S., (2011). Using Semantic Similarity to Improve Information Discovery in Spatial Data Infrastructures. *Journal of Information and Data Management,* 2(2), 181-194.

Baeza-Yates, R. and Ribeiro-Neto, B., (1999). *Modern information Retrieval*. Wokingham: Addison-Wesley.

Bernard, L., and Craglia, M. (2005). SDI - From Spatial Data Infrastructure to Service Driven Infrastructure. In *Workshop on Cross-learning between Spatial Data Infrastructures, and Information Infrastructures.*

Fox, E. A. and Shaw, J. A., (1993). Combination of Multiple Searches. In *Second Text Retrieval Conference*. NIST Special Publications.

Hübner, S. and Visser, U., (2003). Temporal Representation and Reasoning for the Semantic Web. In *Twenty-First International Florida Artificial Intelligence Research Society Conference.* AAAI Press.

Jin, P., Li, X., Chen, H., and Yue, L., (2010). CT-Rank: A Time-aware Ranking Algorithm for Web Search. *Journal of Convergence Information Technology*, 5(6), 99-111.

Manica, E., Dorneles, C. F., and Galante, R. M., (2010). Supporting Temporal Queries on XML Keyword Search Engines. *Journal of Information and Data Management*, 1(3), 471-486.

Open Geospatial Consortium. (2004). *OGC Web Map Service Interface*. Retrieved February 26, 2012, from: http://portal.opengeospatial.org/files/?artifact_id=4756

Open Geospatial Consortium, (2005). *Web Feature Service implementation specification*. Retrieved February 26, 2012, from: https://portal.opengeospatial. org/files/?artifact_id=8339.

Pustejovsky J., Castaño J., Ingria R., Saurí R., Gaizauskas R., Setzer A. and Katz G., (2006). TimeML: Robust Specification of Event and Temporal Expressions in Text. In *Fifth International Workshop on Computational Semantics.*

Strötgen, J. and Gertz, M., (2010). A System for Exploring Spatio-Temporal Information in Documents. *Proceedings of the VLDB Endowment*, 3(1), 1569-1572.

Tversky A., (1977). Features of similarity. *Psychological Review*, 84 (4), 327-352.

Vögele, T., Hübner, S. and Shuster, G., (2003). BUSTER - An Information Broker for the Semantic Web. *Künstliche Intelligenz*, 17(3), 31- 34.

Williamson I., Rajabifard, A. and Feeney, M. F., (2003). *Developing Spatial Data Infrastructures: From Concept to Reality*. London: Taylor & Francis.