



Article

# Flexible Trip-Planning Queries

Gloria Bordogna <sup>\*</sup>, Paola Carrara, Luca Frigerio and Simone Lella 

CNR IREA, Via A. Corti 12, 20133 Milano, Italy

\* Correspondence: bordogna.g@irea.cnr.it

**Abstract:** The current practice of users searching for different types of geo-resources in a geographic area and wishing to identify the most convenient routes for visiting the most relevant ones, requires the iterative formulation of several queries: first to identify the more interesting resources and then to select the best route to visit them. In order to simplify this process, in this paper a novel functionality for a geographic information retrieval (GIR) system is proposed, which retrieves and ranks several routes for visiting a number of relevant georeferenced resources as a result of a single query, named flexible trip-planning query. An original retrieval model is defined to identify the relevant resources and to rank the most convenient routes by taking into account personal user preferences. To this end, a graph-based algorithm is defined, exploiting prioritized aggregation to optimize the routes' identification and ranking. The proposed algorithm is applied in the proof-of-concept of a Smart cOMmunity-based Geographic infoRMation rETrieval SysTEM (SO-GREAT) designed to strengthen local communities: it collects and manages open data from regional authorities describing categories of authoritative territorial resources and services, such as schools, hospitals, etc., and from volunteered geographic services (VGSs) created by citizens to offer services in their neighbourhood.

**Keywords:** content-based retrieval; geographic information systems; geospatial analysis; query processing; search methods



**Citation:** Bordogna, G.; Carrara, P.; Frigerio, L.; Lella, S. Flexible Trip-Planning Queries. *ISPRS Int. J. Geo-Inf.* **2023**, *12*, 204.  
<https://doi.org/10.3390/ijgi12050204>

Academic Editors: Wolfgang Kainz, Baojie He, Deo Prasad, Ali Cheshmehzangi, Wu Deng, Samad Sepasgozar and Xiao Liu

Received: 28 February 2023

Revised: 28 April 2023

Accepted: 12 May 2023

Published: 16 May 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The diffusion of both the Internet and pervasive technologies such as social networks caused people to experience a widening of their virtual horizons: we can be in touch with anybody anywhere on Earth at any time, sharing interests or business. At the same time, increased mobility of people for jobs or leisure, reduction in long-term living in the same city, and fast modification of services (such as shop and office locations), are factors that increase the need to be rapidly and constantly aware of local resources and neighbours. In such a context, the Internet becomes a straightforward means to explore our local world and to avoid isolation.

A Google survey found that four out of five people use search engines to conduct local searches [1]. This trend has been reinforced in the recent pandemic period, when the Internet proved to be essential for survival, allowing to update day-by-day information on available local resources and services and to contact people who can offer help with daily chores in the neighbourhood.

Technologies that can support people to explore the territorial resources where they live include:

- Location-based social networks, in which users can send and receive messages from people living in an area of interest, such as [2];
- Geographic information retrieval systems (GIRs), concerned with improving the quality of geographically specific information retrieval [3,4];
- Location-based services (LBSs), where a user's current location is used as real time contextual information in the delivery of services [5];
- Collaborative mapping services such as OpenStreetMap [6] and Google map maker enabling crowdsourcing georeferenced contents.

Such systems allow to search different types of georeferenced resources and services, such as hotels and restaurants, in specific geographic areas. They also provide routing facilities among selected relevant resources by proposing routes for visiting them with a degree of personalization depending on the user location (in the case of location-based services), the transport means (for example, car, public transport, or on foot), and route types and conditions (for example, highways, traffic, etc.).

However, in such systems, the process to obtain the best route to visit several relevant resources of different types, is not simple and requires more than one step. In fact, a preliminary phase is searching for the resources and ranking them with respect to their relevance to the user needs; then the user must select the ones s/he deems interesting to visit, and finally s/he must instruct a routing facility by indicating a desired visiting order.

For example, with current means, a tourist in Rome wishing to identify an itinerary starting from a “3-star hotel” in the neighbourhood of an archaeological site at 10 minutes’ walk from a typical Italian restaurant must engage with possibly several different queries. In a first step, s/he must formulate queries for each type of resources (i.e., “3 star hotels”, “archaeological sites”, “typical Italian restaurant”) to retrieve the relevant resources of each interesting type. Then, in the retrieved results, s/he should select the resources that satisfy the desired spatial conditions (i.e., choosing 3-star hotels that are in the neighbourhood of the retrieved archaeological sites and are at 10 minutes’ walk from the retrieved restaurants). Finally, s/he should ask a routing system to indicate the itinerary by specifying a desired visiting order.

The above process can be performed by formulating a single query, the trip-planning query, an approach first proposed in [7–9]. In this paper, this approach is generalized and extended to allow a more flexible formulation and interpretation of the trip-planning query, so as to produce a list of ranked routes. This is made possible by taking into account the gradual content relevance of the resources, their gradual spatial relevance, and their relative priority. They are computed by evaluating the user needs and flexible spatial conditions specified in the query. An original graph-based algorithm is then defined to optimize routes’ identifications and ranking. This way the proposal allows to identify a number of visiting routes to resources of different types and to rank them.

The novelties of the proposal are several: the definition of original flexible trip-planning query semantics with distinct priorities for the resources of interest; the definition of a graph-based algorithm to optimize the routes’ identifications and ranking exploiting a prioritized aggregation; the definition of two flexible spatial operators, named “in neighbourhood” and “close”, to allow specifying tolerant spatial conditions while preserving the privacy of user location.

The model we propose is then exemplified in a proof-of-concept implementation, the prototypal service platform named Smart cOMmunity-based Geographic infoRmation rETrievAl SysTem (SO-GREAT); it enables local searches on both authoritative resources, published on the web portals of regional public administrations, and volunteered geographic services (VGSs) created by both citizens and voluntary organizations by means of a web application in order to offer services to neighbours on a voluntary basis [7]. From a social point of view, the advantages offered by the SO-GREAT system are the availability of reviewed information on resources and services recognized as authoritative by public administrations and thus with high veracity, the enrichment of such information with comments and ratings from social networks, and the enabling of smart local communities by allowing searches on VGSs.

In Section 1 of this paper, the related literature and background notions are introduced and presented. Section 2 defines the data model and the graph-based algorithm for identifying and ranking the routes. Section 3 discusses some case studies of flexible trip-planning-query evaluation and describes the implementation using the SO-GREAT service platform. The Conclusions report the main achievements and future work.

## 2. Materials and Methods

### 2.1. Related Work

Traditional spatial querying is intended as the selection of georeferenced resources, spatial objects in a spatial database satisfying spatial conditions. Examples are a range query, which retrieves all spatial objects that intersect with a specified spatial region, and a K-nearest neighbourhood query, which retrieves the K spatial objects that are closest to a given point.

On the other hand, in GIRs, LBSs, and mapping services, spatial keyword queries take user locations and user-supplied keywords as arguments and return web objects that are both spatially close to the query location and relevant to the search keywords [8–10].

These queries are fundamentally different from spatial path queries, in which the objective is to navigate a graph or a tree data structure to retrieve information that is connected in a specific way [8]. A spatial path query is used to retrieve an optimal path, for example to find the shortest path from A to B, where A and B can be spatial objects, or to find the path from A to B that satisfies some constraint, such as the minimization of the time needed to travel from A to B by crossing a number of different other spatial objects of interest, C, D, and E. In the graph, nodes are associated with distinct spatial objects, and edges represent the connections between pairs of spatial objects, associated with properties such as the distance, which can be either a Euclidean distance or a route distance, the time distance needed to reach the two spatial objects, and possibly other properties, for example the type of route, the traffic load, etc.

To recap the main characteristics of the reviewed trip-planning approaches we consider these two main categories:

- Approaches for best routing to relevant resources, which require to identify the relevant territorial resources belonging to categories of interest declared in the query [11];
- Best itinerary-planning approaches, which require to specify the starting and ending point of the trip and a cost function such as minimum distance, minimum time, and/or qualitative aspects of the roads specified by query keywords [12–17].

In the literature, the first kind of trip-planning query, a kind of spatial path query, was proposed for spatial databases in [11]. In this context, a query specifies a starting and an ending location and a set of categories of interest. This allows to plan itineraries passing a set of points belonging to the specified categories, for example to go from home to the office by passing through a post office and a bank. As in our case, the difficulty to evaluate this query is the fact that there are generally many points for each category, and thus the algorithm must choose the ones that minimize an overall cost function. In the original proposal, the cost function is defined as the overall distance of the trip from the starting point to the ending point, and several approximation algorithms to identify the best solution were defined on the graph representing the points in the database.

Our proposal differs from this in several aspects: firstly, we consider geo-referenced web resources and flexible criteria to produce ranked routes. Flexible criteria take into account both a visiting priority of the categories, the relevance of the spatial resources to the categories, and the satisfaction of spatial conditions between the spatial objects.

Moreover, we do not have a predefined graph representing all spatial objects in the database, but we build an evaluation graph for each trip-planning query during its evaluation process.

Other itinerary retrieval models have been defined, for example supporting travel searches for an upcoming vacation so as to exploit advice and experiences of previous visitors [12,13]. These models rank routes to territorial resources in a collection of stored itineraries, either detected in GPS-crowdsourced track logs or automatically detected within tables, based on a measure of similarity with respect to a sequence of locations. Our approach is different since we do not have predefined stored routes, i.e., trajectories crossing the territorial resources, but we build them at retrieval time, by identifying the resources as well.

Some route-querying approaches were defined taking into account the topology and metrics of the underlying road networks using textual documents containing geographic information [14–16]: their objective is finding a route with some qualitative characteristics expressed by a set of query keywords while minimizing a travel cost generally defined as the travel distance. These queries are Boolean keyword queries also intended for planning trips, for example for visiting specific types of points of interest (POIs), such as a route crossing a specific park and a specific cafeteria. These approaches are similar to our proposal, but they do not allow a flexible query interpretation to compute the semantic relevance of POIs to the query keywords.

Another approach close to the one we propose is described in [17], where a bounded-cost informative route query retrieves the optimal route that is the most textually relevant to the user-specified query keywords, subject to a travel-cost constraint defined as the maximum travel distance or time. For example, a tourist may issue a query specifying the starting and ending locations of a route together with query keywords, “scenic, nature-friendly”, to find a route that is both scenic and nature-friendly. Then the computed route score is based on the textual description of the entire route. While in this case the query specifies the starting and ending locations and the characteristic of the route between them, in our approach the query contains keywords describing different categories of resources of interest and possibly spatial conditions between them.

The work proposed in this paper combines methods and techniques of several research fields: GIRs [3,9], LBSs [5,17–19], fuzzy databases and flexible querying [20–24], graph theory [24], and aggregation operators [25,26].

The originality of the flexible trip-planning query facility is several-fold:

- The modelling of the visiting priority of the distinct types of spatial objects, both resources and VGSs, thanks to the application of a prioritized aggregation operator [26];
- The evaluation of flexible spatial conditions expressed by the operators “in\_neighbourhood” and “close” between geolocations of pairs of resources while preserving the privacy of both users and services created by volunteers [20,27];
- The ranking of the different retrieved routes by taking into account both the priorities and the flexible constraints.

Finally, the implementation stems from the evolution the SO-GREAT system [7] by introducing several novelties. The implementation of the authoritative territorial resource descriptions collected from open data repositories are automatically georeferenced by exploiting fuzzy rules for geo-parsing and geo-coding [9]. Furthermore, since these descriptions generally consist of very short texts, they are enriched with comments filtered from social networks posts, specifically Twitter, by applying a social annotation approach of documents in which both terms extracted from the resource descriptions and their geo-reference are used as filtering criteria [28]. Finally, the open data collection is managed in an integrated way and complemented by a set of VGSs [18] freely created by citizens through a web application, so as to widen the availability of territorial resources in order to strengthen local communities.

## 2.2. Background Notions

In this subsection, we introduce some useful definitions that we will use in the formalization of the model for trip-planning-query evaluation.

### 2.2.1. Flexible Spatial Conditions

A flexible spatial condition is defined by a soft spatial relation, and identified by a linguistic value  $f_{sc}$  and an associated membership function  $\mu_{f_{sc}}(A,B) \rightarrow [0, 1]$  with  $(A,B) \in \mathbb{R}^2 \times \mathbb{R}^2$ , that associates a value in  $[0, 1]$  to each pair of spatial objects’ geometric components  $A$  and  $B$  defined on the spatial domain  $\mathbb{R}^2$ . When the geometric components  $A$  and  $B$  are points of the spatial domain, a flexible spatial condition is easier to compute.

In spatial queries, the utility of specifying flexible spatial conditions is to be able to rank the retrieved elements based on their degrees of satisfaction  $\mu_{f_{sc}}(A,B)$ , thus avoiding

the drawbacks of crisp spatial conditions assuming values in the Boolean domain  $\{0, 1\}$  that may produce either a huge number of not discriminated answers or even the empty set.

Their definition depends on several factors such as the application domain, the scope of the database, and the user. In a generic spatial database, we can consider these types:

- Flexible Geometric conditions for evaluating degrees of satisfaction of geometric relationships between pairs of spatial objects defined on the domain of some spatially derived attributes (area, ellipticity, etc.). These conditions can be expressed by means of linguistic values such as bigger, smaller, more circular, much longer, etc.: for example to retrieve and rank the European nations with a territory bigger than Italy;
- Flexible Topological conditions for evaluating degrees of satisfaction of topological relationships between pairs of spatial objects such as very overlapped, meeting, almost South, East, etc.: for example to retrieve the nations sharing a border with Italy, and to rank them based on the length of the shared border;
- Flexible directional conditions expressed by linguistic values such as almost South, East, South-West, etc., for computing degrees of satisfactions depending on the directional relationships between pairs of spatial objects: for example to retrieve and rank European nations that are North-East of Italy;
- Flexible Metric conditions expressed by linguistic values such as close, far, very far, etc., for computing degrees of satisfactions depending on the distance between pairs of spatial objects: for example to retrieve and rank European nations depending on their distance to Italy.

### 2.2.2. Prioritized Aggregation Operator

The prioritized aggregation operator was introduced by Yager in [26].

In the following, we recall its definition and main properties.

Given a collection of flexible selection conditions  $C$ , in our case the flexible selection criteria of the resources and VGSs, partitioned into  $q$  distinct types,  $H_1, H_2, \dots, H_q$ , such that  $H_i = \{C_{i1}; C_{i2}; \dots; C_{in}\}$ , are the conditions for selecting the resources and VGSs belonging to type  $H_i$ ; we assume a prioritization between these types depending on their order in the query  $q$  such that:

$$H_1 > H_2 > \dots > H_q.$$

For any alternative  $A \in D$ , a spatial object, for each condition  $C_{ij}$  we have a value  $C_{ij}(A) \in [0, 1]$  indicating its satisfaction.

The prioritized aggregation function that for each  $A$  computes its global satisfaction of all conditions is defined as follows:

$$C(A) = \sum_{i=1}^q \sum_{j=1}^{n_i} w_{ij}(A) * C_{ij}(A) = \sum_{i=1}^q T_i \sum_{j=1}^{n_i} * C_{ij}(A),$$

in which the weights  $w_{ij}(A) \in [0, 1]$  are calculated recursively to reflect the priority relationship.

For each priority type  $H_i$  we calculate  $S_i = \min_{j=1 \dots n_i} (C_{ij}(A))$ , where  $S_i$  is the value of the least satisfied condition of type  $H_i$  for the alternative  $A$ .

Assuming that  $S_0 = T_1 = 1$ , the weight  $T_i$  can be computed recursively by the following:

$$T_i = S_{i-1} \times T_{i-1}.$$

$C(A)$  is monotonic: if  $C_{kj}(A)$  increases then  $C(A)$  cannot decrease.

Finally, it can be demonstrated that the conditions of type  $H_i$  contribute proportionally to the product of the satisfaction of the conditions with greater priority (thus  $T_i \geq T_k$  for  $\forall i < k$ ). This implies that the poor satisfaction of any condition with greater priority reduces the ability of a condition with lower priority to compensate the result. This is the fundamental feature of the prioritization relationship.



### 2.3. Semantics of Flexible Trip-Planning Queries

In the context of this paper, queries are formulated with the aim of planning trips for visiting relevant resources of different types that are retrieved based on both their content relevance and the satisfaction of spatial conditions with respect to other relevant resources of greatest interest.

More generally, trip-planning queries are expressed in conjunctive normal form, in which free text keywords (possibly more than one, connected by OR) define distinct conjuncts, which are selection conditions for distinct types of resources and VGSs. Each conjunct has a given priority defined by its position in the query and can be constrained to satisfy a different spatial condition (*in\_neighbourhood* or *close*) to delimit the reachability of the resources and services satisfying the conjunct with respect to any of the resources and VGSs satisfying the preceding conjunct.

Conjuncts are connected by the prioritized “AND possibly” operator, that is an aggregation operator specifying the greatest priority of the left conjunct with respect to the right conjunct [26]. This means that the existence of a relevant and reachable resource retrieved by the left conjunct is mandatory in order to evaluate the right conjunct, which is considered optional. An example of flexible query is the following:

$$\begin{aligned} & (“kinder\ garden”) \text{ AND possibly } close (“recreation\ centre” \text{ OR } “library”) \\ & \text{ AND possibly } in\_neighbourhood (“baby\ sitter”) \end{aligned} \quad (1)$$

A default geographic area in the neighbourhood of which to search for the resources matching the first conjunct “kinder garden” is assumed. This can be identified by either a polygon (to preserve the user privacy) or a point on the map. Usually most geographic gazetteers geo-reference cities using points; thus, indicating a toponymal term allows to set the default reference area. Alternatively, one can provide a service to registered users whose profile contains their home location and preferences for personalizing their searches: in this case the first selection condition is assumed as constrained by the implicit spatial condition “*in\_neighbourhood*” referring to the location of the user.

The trip-planning query above asks to find a kindergarten in the neighbourhood of the user home (a circle around the user home delimits the neighbourhood); if some instances are retrieved, then one would prefer the best kindergarten close to a recreation centre or library; finally, if both previous conditions are satisfied, one would prefer the best route crossing a kindergarten close to a library or recreation centre that is in the neighbourhood of a babysitter. The babysitter’s geographic scope can be an extended area defined by a polygon on the map to preserve the privacy of the volunteer and its neighbourhood extends outside this area to a maximum distance as specified in the profile of the VGS author, that is, the babysitter who created the VGS. All possible routes are the following:

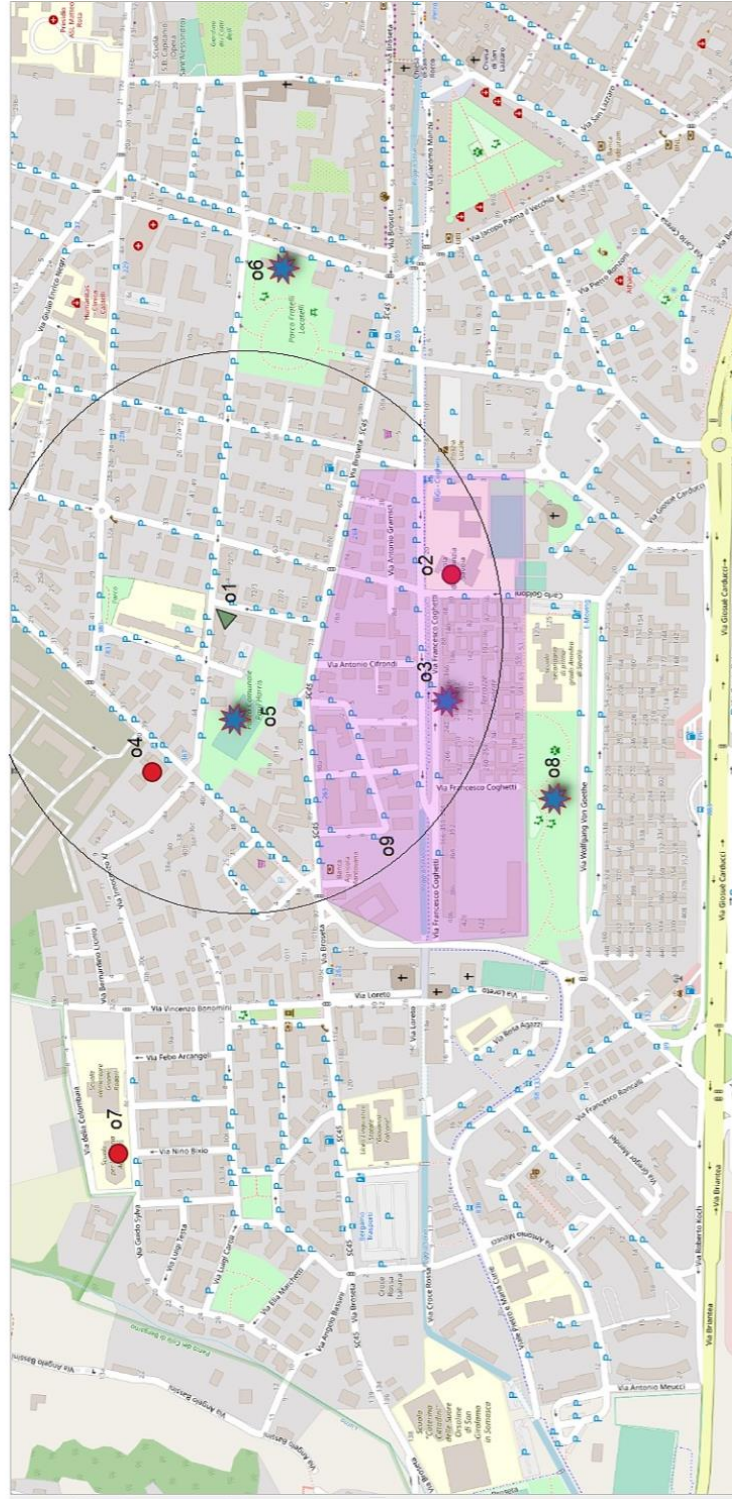
o1–o2–o3–o9; o1–o4–o3–o9; o1–o7–o3–o9;  
o1–o2–o5–o9; o1–o4–o5–o9; o1–o7–o5–o9;  
o1–o2–o6–o9; o1–o4–o6–o9; o1–o7–o6–o9;  
o1–o2–o8–o9; o1–o4–o8–o9; o1–o7–o8–o9;  
o1–o2–o3; o1–o4–o3; o1–o7–o3; o1–o2–o5; o1–o4–o5; o1–o7–o5;  
o1–o2–o6; o1–o4–o6; o1–o7–o6; o1–o2–o8; o1–o4–o8; o1–o7–o8;  
o1–o2; o1–o4; o1–o7.

The objective of the trip-planning-query evaluation is to select, from the set of all possible routes listed above, those that satisfy the query by trimming those that do not satisfy it, and then by ranking the routes passing through the most relevant and the most reachable resources of distinct types.

The selection and ranking of the routes of this running example will be discussed in Section 3.1.

The evaluation function that retrieves and ranks the routes computes a directed labelled hierarchical graph that codifies the visiting priority of the retrieved resources and VGSs in its hierarchy: at the k-th level the vertexes represent relevant retrieved resources or

VGSs belonging to the type having  $k$ -th priority in the query. Figure 1 reports a scenario depicting some resources and VGS of different types and the user home identified by the green polygon so as to preserve privacy.



**Figure 1.** Example scenario of resources and VGS of distinct types; the circle represents the external boundary of the neighbourhood of the user's home that is located in the green triangle. The violet polygon represents the babysitter's geographic scope, indicating her/his availability to provide the service within the area or close to it within a maximum distance  $\delta$  as defined in her/his profile.

The edges connecting vertexes of adjacent levels are labelled with reachability scores expressing the satisfaction of the spatial conditions between the pair of resources at the edge's vertexes. A route starts in the root vertex, which is associated with a reference location that can be either the user's location or a geographic area, and descends the graph hierarchy by crossing vertexes of the lower levels that are reachable from the preceding vertex on the route. A convenience score is iteratively computed at each vertex for ranking the routes or for trimming them by adding the not null contribution of the next vertex on the route to the intermediate convenience score computed so far. This is performed by applying a prioritized aggregation operator [26] whose semantics has similarities with a relevance score used to rank textual documents based on multiple dimensions [25].

A different query, although expressing interest for the same types of geo resources and VGS, is the following, in which the priority of the VGS "baby sitter" is greater than that of ("recreation centre" OR "library"):

("kinder garden") AND possibly close ("baby sitter")  
AND possibly in neighbourhood ("recreation centre" OR "library").

This different priority affects the ordering and the retrieved trips as it will be discussed in Section 3.1.

#### 2.4. Graph-Based Algorithm for Flexible Trip-Planning-Query Evaluation

Let us consider to have a reference geographic area  $u.home$ , a positive value  $\delta \in \mathcal{R}^+$  (for example expressed in meters) that defines the neighbourhood of a given geometry where to search for the resources and VGS, and a set of stored spatial objects:

$$O := \cup o,$$

where  $o$ : uniquely identifies either a resource or a VGS of a given type  $H_i$  with  $H := \cup H_i$ ;  $o.geo$ : is a 2D geometry representing the geolocation of the resource or the geographic scope of the VGS, so that  $o.geo$  is either a point or a convex polygon; and  $o.r\_rank$ : is the semantic relevance of the object  $o$  to a trip-planning query.

A trip-planning query  $q \in Q$  is an expression in conjunctive normal form over free text keywords and operators as follows:

$$q := \langle key [OR key]^* \rangle_1 [AND possibly sop \langle key [OR key]^* \rangle_{p+1}]^*, \quad (2)$$

where square brackets indicate optional elements, \* indicates zero or more repetitions, OR is the Boolean operator for defining alternative selection conditions, "AND possibly" is a prioritized aggregation operator between a mandatory (left) and an optional condition (right),  $key$  identifies a free text keyword, and  $sop \in \{in\_neighbourhood, close\}$  is a spatial operator specifying a spatial condition between the geo-reference of pairs of resources  $o_p$  and  $o_{p+1}$  retrieved by two distinct keywords appearing in  $p$  and  $p+1$  adjacent query conjuncts, respectively.

$p \in \mathbb{N}$  identifies the priority of visit and is a positive integer inversely proportional to the priority.

The query is expanded so that the first conjunct is assumed as constrained by the  $in\_neighbourhood$  condition with respect to a default reference area  $u.default$  that can be specified by a toponymal term of a city or region. In the case of the user being registered, the default reference area is the user's home:  $u.default = u.home$ .

The explicit query form is the following:

$$q := u.default in\_neighbourhood \langle key [OR key]^* \rangle_1 [AND possibly sop \langle key [OR key]^* \rangle_{p+1}]^*. \quad (3)$$



A registered user who specifies a single keyword demands for resources retrieved by key whose location is in the neighbourhood of  $u.home$ , i.e., within a maximum distance  $\delta$  from it. This simple query has the following explicit form:

$$u.home \text{ in\_neighbourhood } <key> \quad (4)$$

#### 2.4.1. Ranked Route Definition

Let us define a query graph  $G_q(V, E)$  for query  $q$  defined as in (3) in which:

$v = (o, p, o.r\_rank) \in V$  is a vertex identifying object  $o$  with  $p$  priority and semantic relevance score  $o.r\_rank \in [0, 1]$  with respect to a query keyword  $key$  appearing in the conjunct of priority  $p$ .

A special vertex is defined as  $v_0 = (u, 0, 1)$  associated with the geographic reference area  $u.default$ . Vertexes are identified during query evaluation;  $e_{i,j} = (v_i, v_j, s\_rank_{i,j}) \in E$  is an edge connecting two vertexes  $v_i = (o_i, p-1, o_i.r\_rank)$  and  $v_j = (o_j, p, o_j.r\_rank)$  corresponding to two distinct objects retrieved by adjacent conjuncts in the query having priorities  $p-1$  and  $p$ , respectively.

$s\_rank_{i,j} \in (0, 1]$  is the reachability score of the object in vertex  $v_j$  by the object in vertex  $v_i$ . It is computed as the degree of satisfaction of the flexible spatial condition  $sop_{p-1,p}$  in  $q$ ; it can be a degree of satisfaction of either *close* or *in\_neighbourhood* computed between the geolocations  $o_i.geo$  and  $o_j.geo$ , respectively. Thus, the degree of  $s\_rank_{i,j}$  depends on the geographic coordinates of the locations of the objects (resources or VGSs) associated with vertexes  $j$  and  $i$ .

We define a ranked route as a branch:

$$\rho\rho: = (v_0, v_1, v_2, \dots, v_M; RSV((\langle v_0, v_1, v_2, \dots, v_M \rangle))) \mid \forall v_{i-1}, v_i \wedge i = 1, \dots, M \exists e_{i-1,i} = (v_{i-1}, v_i, s\_rank_{i-1,i}) \wedge s\_rank_{i-1,i} > 0 \quad (5)$$

in which  $M \leq N$ , with  $N$  being the number of conjuncts in  $q$ .

$v_0$  is the vertex corresponding to  $u.default$ ;  $RSV \in R^+$  is the retrieval status value of the route, named convenience score, which is computed based on the prioritized aggregation inspired by the prioritized operator introduced in the previous subsection [26] and defined recursively as follows:

$$\begin{aligned} RSV_1 &= RSV(\langle v_0, v_1 \rangle) = \min(o_1.r\_rank_1, s\_rank_{0,1}) \\ \forall p > 1 \quad RSV_p &= RSV(\langle v_0, \dots, v_p \rangle) = (RSV(\langle v_0, \dots, v_{p-1} \rangle) + \min(o_p.r\_rank_p, \\ &s\_rank_{p-1,p})) * \min(o_p.r\_rank_p, s\_rank_{p-1,p}) \\ &= (RSV_{p-1} + \min(o_p.r\_rank_p, s\_rank_{p-1,p})) * \min(o_p.r\_rank_p, s\_rank_{p-1,p}), \end{aligned} \quad (6)$$

in which  $s\_rank_{0,1}$  is the satisfaction degree of the *in\_neighbourhood* operator by the geolocation of object in  $v_1$  with respect to the reference geographic area  $u.default$  represented by vertex  $v_0$ . This means that for the objects of highest priority its  $RSV_1$  is given by the minimum of their semantic relevance score and the satisfaction degree of their flexible spatial condition from  $v_0$ .

As the priority decreases by increasing  $p$ , a reward to the  $RSV$  is accounted for each successive vertex on the route proportional to the minimum between the semantic relevance score and the satisfaction of the spatial condition of the last vertex  $v_p$  on the route, thus modelling its lowest priority. Thus, the final ranking of the route depends on a combination of both the semantic relevance of the objects, i.e., their appropriateness to the type of interest, and the spatial conditions' satisfactions degrees, i.e., on their mutual geographic locations. In the following Sections 2.4.2 and 2.4.3, the criteria to compute the  $s\_rank$  will be defined, while the  $r\_rank$  computation will be described in Section 3.2.3.

When a vertex does provide no increment to the  $RSV$  the route ends at the preceding vertex. This way, longer routes can be rewarded more than shorter ones, since they allow to reach more objects of the interesting types, and the more relevant and reachable the objects with high priority are, the more the route is convenient and ranked in top positions.

#### 2.4.2. “in\_neighbourhood” Definition

The flexible spatial condition *in\_neighbourhood* evaluates the intersection between two geometries consisting of convex polygons or points with a broad boundary.

Let us indicate, by  $A^\circ$ ,  $A^+$  and  $B^\circ$ ,  $B^+$ , distinct geometries defined on the bidimensional spatial domain, with  $A^\circ \subseteq A^+$  and  $B^\circ \subseteq B^+$ , and represented as either points or georeferenced convex regular polygons by a closed chain of connected line segments.

Let us indicate, by  $A = A^\circ \cup A^+$  and  $B = B^\circ \cup B^+$ , two convex regular polygons with broad boundaries, and by  $A^-$  and  $B^-$  the complement of  $A$  and  $B$ . We name  $A^\circ$  and  $B^\circ$  yolks, and  $A^+ - A^\circ$  and  $B^+ - B^\circ$  broad borders.

*in\_neighbourhood*( $A, B$ ) is defined by the maximum of the values in Table 1, when the intersection of the correspondent convex regular polygons is true.

**Table 1.** Definition of the *in\_neighbourhood* operator based on the 9-intersection model for 2D regular convex geometries with broad boundaries. Below the table some examples of overlapping geometries are shown.

$A^\circ \cap B^\circ = 1$ $A^+ \cap B^\circ = 0.5$ $A^- \cap B^\circ = 0$	$A^\circ \cap B^+ = 0.5$ $A^+ \cap B^+ = 0.25$ $A^- \cap B^+ = 0$	$A^\circ \cap B^- = 0$ $A^+ \cap B^- = 0$ $A^- \cap B^- = 0$
$A^\circ \cap B^\circ = 1$	$A^\circ \cap B^+ = 0.5$	
$A^+ \cap B^+ = 0.25$	$A^- \cap B^+ = 0$	

The algorithm can be defined as follows:

if ( $A^\circ \cap B^\circ$ ) then *in\_neighbourhood*( $A, B$ ) = 1  
 else if ( $A^\circ \cap B^+$ ) OR ( $A^+ \cap B^\circ$ ) then *in\_neighbourhood*( $A, B$ ) = 0.5  
 else if ( $A^+ \cap B^+$ ) then *in\_neighbourhood*( $A, B$ ) = 0.25  
 else *in\_neighbourhood*( $A, B$ ) = 0.

Notice that  $A^+$  and  $B^+$  can be obtained by combining  $A^\circ$  and  $B^\circ$ , i.e., the reference geographic area and the geo-reference of both resources and VGSs, with a positive value  $\delta$ , by applying the Minkowski sum  $\oplus$  defined as follows [20]:

$$A^+ = A^\circ \oplus \text{circle}(b, \delta) = \{b + \rho, \forall b \in A^\circ \mid \rho \in \text{circle}(b, \delta)\}. \quad (7)$$

The Minkowski sum is defined as the union of all the translations of a circle of radius  $\delta$  centred in any points  $b$  of  $A^\circ$ . In the implementation of the SO-GREAT system described in the following sections, the geometric type values are encoded in GeoJSON and  $A^\circ$  can be either a point or a convex regular polygon with  $m$  vertices; the Minkowski sum determines a polygon either with circumradius  $\delta$  if  $A^\circ$  is a point, or with circumradius equal to  $|r + \delta|$  when  $A^\circ$  is a convex regular polygon with circumradius  $r$ . By changing  $\delta$  we can customize the flexibility.

Assuming these implementation constraints, if the two yolks overlap each other then *in\_neighbourhood*( $A, B$ ) = 1, i.e., the result is maximum. If only one broad border overlaps the yolk of the other geometry then *in\_neighbourhood*( $A, B$ ) = 0.5: in fact, in this case  $A$  and  $B$  are less overlapped than in the previous case, since one of the two yolks does not intersect the second geometry at all. Finally, if only the broad borders overlap each other but do not overlap the yolks then *in\_neighbourhood*( $A, B$ ) = 0.25: in fact, in this case both yolks do not intersect the other geometry. This definition is based on heuristics so as to reward the broad boundary objects that overlap one another more with a greater value.

With this definition, since it is sufficient that the two yolks overlap one another to get the maximum satisfaction degree 1, the privacy of both the user and the VGSs locations can be preserved.

#### 2.4.3. Definition of “close”

The flexible spatial condition *close* is defined as the Euclidean distance (*Dist*) between  $A^\circ$  and  $B^\circ$  if  $A^+$  overlaps  $B^+$ , otherwise it is zero:

$$close(A, B) = \begin{cases} \frac{1}{1 + \frac{Dist(ctr(A^\circ), -ctr(B^\circ))}{MAXDist}} & \text{if } Dist(ctr(A^\circ), ctr(B^\circ)) < MAXDist \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

where *Ctr* is a function computing the centroid coordinates of a polygon, *Dist* is the Euclidean distance between two points, and  $MAXDist = \delta$  is the maximum distance for two objects to be reachable. This parameter in the implementation is defined in the user profile so as to be able to flexible adapt the semantics of *close* to the user.

When  $close(A, B) > 0$ , the two objects are considered reachable to a degree. By this definition, it is not necessary to have precise geolocations to compute their closeness. This way we can preserve the privacy of the geolocations of both the user and VGS. When A and B are perfectly overlapping their closeness is maximum. By changing  $\delta$  we can customize the flexibility.

#### 2.4.4. Graph-Based Algorithm for Query Evaluation

Given a query  $q$  defined as in (3) the creation of a query graph  $G_q$  is performed in maximum  $N$  cycles, where  $N$  is the number of query conjuncts.

First the root vertex  $v_0 = (o_0, 0, 1)$  is generated, in which  $o_0 = u.default$  extended by  $\delta$  by applying (7) (for registered users  $u.default = u.home$ ).

At each cycle  $I > 0 \wedge i \leq N$ ,  $\forall o_{r,i} \in O_i$  retrieved by a query keyword  $kw_i$  in the  $i$ th conjunct and  $\forall o_{s,i-1} \in v_{i-1} \in e_{i-2,i-1}$ , the reachability score  $s\_rank_{i-1,i}$  is computed. This means evaluating the degree of satisfaction of  $sop_{i-1,i}$  between their geolocations  $o_{s,i-1}.geo$  and  $o_{r,i}.geo$ :  $s\_rank_{i-1,i} = sop_{i-1,i}(o_{s,i-1}.geo, o_{r,i}.geo)$ .

If  $s\_rank_{i-1,i} > 0$ , a new vertex  $v_i$  is generated linked to  $v_{i-1}$  by means of the new edge  $e_{i-1,i}$ . Notice that objects that are not reachable by any vertex already created are trimmed. This allows reducing the operations of the next cycles, thus reducing the complexity of the algorithm.

We can prove that the query graph  $G_q$  has the following characteristics:

- There exists only one root vertex  $v_0 = (u.default, 0, 1)$  that identifies the reference geographic area with maximum  $|O_1|$  departing edges (branches) and no incoming edge;
- There exist a number of vertices  $v_i$ , named leaf, with no departing branches and at least an incoming branch:  $\exists e_{i-1,i} \mid v_i \in e_{i-1,i} \wedge \neg \exists e_{i,i+1} \mid v_i \in e_{i,i+1}$ ;
- All other vertexes have both at least an incoming branch and a departing branch;
- The maximum depth of a branch from root to a leaf is equal–smaller than  $N$ , the number of query conjuncts.

Given the query graph  $G_q$  each route as defined in (5) is a branch starting at the root and ending in a leaf. Furthermore, all possible routes are represented in the graph and for each of them their RSV is computed based on Formula (6).

The query graph can be regarded as a union of hierarchical trees with a common root vertex  $v_0$  and a variable number of distinct branches, i.e., routes, ending in leaf vertexes. The branches of the trees are different as far as at least one vertex. The RSV computation is performed in a depth of first tree traversal which is a recursive algorithm that starts from the root, then follows a branch as deep as possible until a leaf is reached. At each vertex at level  $p$  the partial RSV is computed by applying (6) until the root is reached whose RSV corresponds to that of the whole route.

Given a query with  $p$  disjuncts, the maximum depth of the tree is  $p$ ; assuming  $n$  nodes and  $m$  edges the complexity for the depth first traversal is  $O(n + m)$ . In our case, if the prioritization operator of a node at level  $k < p$  is not satisfied, a trimming of the subnodes occurs so that the complexity is reduced.

### 3. Results

#### 3.1. Running Example of Flexible Trip-Planning-Query Evaluation

Let us consider the query defined in (1) in which  $u.default: = u.home$ . The algorithm parses  $q$  from left to right and builds the graph in Figure 2, whose vertexes and edges are reported in Table 2.

Starting from the root vertex  $o_1$ , each level of the graph corresponds to resources and VGSs with a given decreasing priority.

Each vertex has an object identifier, a priority level, and a semantic relevance score  $r\_rank$ . Each edge has a weight that is the reachability score  $s\_rank$ .

Table 3 reports the Euclidean distances between pairs of objects with their degree of closeness defined based on (8).

By evaluating the convenience score from top to bottom by applying (6) we obtain the ranked routes in Table 4. Notice that when the reachability score is null the correspondent semantic score is not computed since the vertex is not included in the query graph.

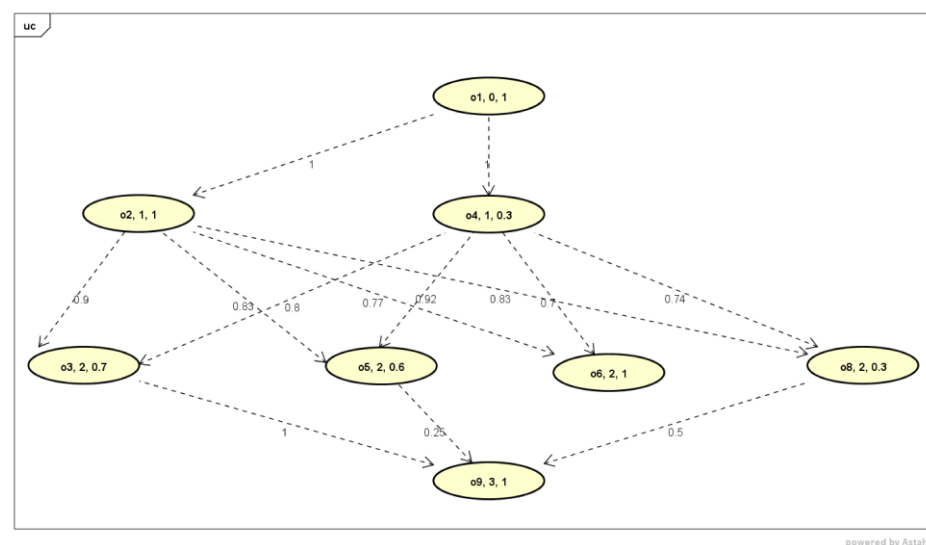
It can be noticed that not all routes satisfy all query conditions. Specifically, route  $o_1-o_2-o_6$  and  $o_1-o_4-o_6$  do not include the babysitter VGS since the recreation centre  $o_6$  is far from the neighbourhood of  $o_9$ .

Nevertheless, in the ranked list these routes are not in the last positions since they comprehend semantic relevant resources, specifically  $o_6$  as the most relevant recreation centre.

Let us consider a second query asking for the same types of resources of query (1) but in a different order, i.e., with a different priority, as follows:

$$\begin{aligned} &(\text{"kinder garden"}) \text{ AND possibly close}(\text{"baby sitter"}) \\ &\text{AND possibly in neighbourhood} \\ &(\text{"recreation centre"} \text{ OR } \text{"library"}) \end{aligned} \quad (9)$$

The corresponding graph and ranked routes are reported in Figure 3 and Table 5.



**Figure 2.** Query graph determined using retrieved resources and VGS as a result of query (1): the vertexes report the OID of the retrieved resource/VGS, its priority, and the semantic relevance score.  $o_1$  is the OID of the user home. The edges report the reachability score between the connected resources.



**Table 2.** Vertexes and edges identified by evaluating query (1). The r-rank semantic relevance scores of the vertexes are computed as follows: r-rank = 1 if the titles of the resources contain all the terms in the query either “kinder garden” = “Scuola infanzia”, or “recreation centre” = “ludoteca”, or “library” = “libreria” and “baby sitter”; the r-rank is a lower value if they contain just one of the query keywords or a synonym: r-rank = 0.3 for “school” = “Scuola”, r-rank = 0.7 for “library” = “biblioteca”, r-rank = 0.6 for “sport centre” = “centro sportivo”, r-rank = 0.3 “public garden” = “giardinetto”. The values of the edges are the s\_rank computed by applying the definition of close in Formula (8) considering the distances between resources shown in Table 3 below.

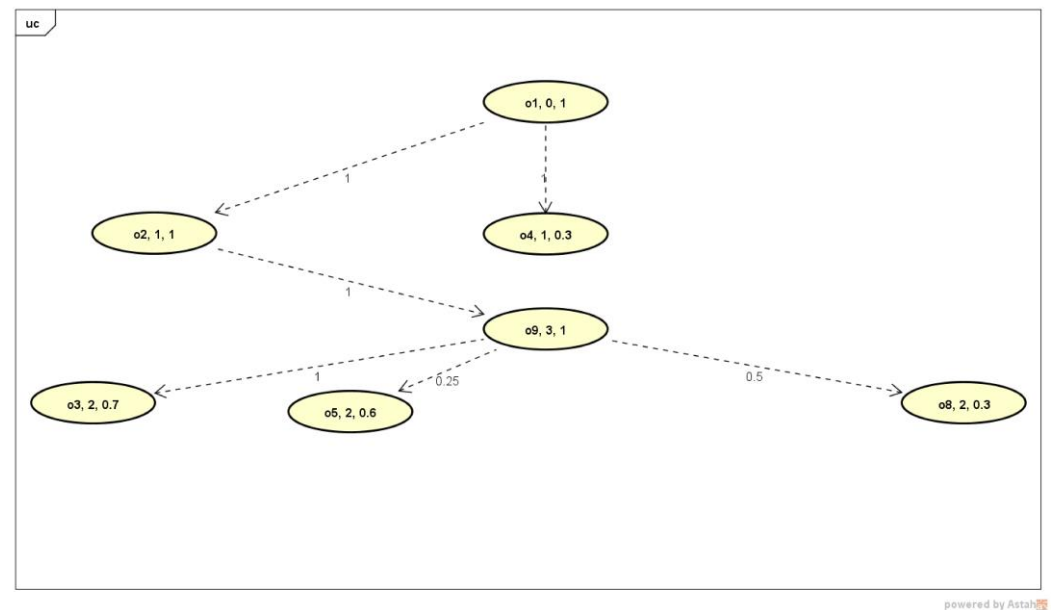
V Set of Vertexes									
Query Keyword			Retrieved Resources and VGSSs					$p$	$r\_rank$
$u.home$			o1 “user home”					0	1
“Kinder garden”			o2 “Scuola infanzia Coggetti”					1	1
			o4 “Scuola privata Virgo . . . ”					1	0.3
			o7 “Scuola dell’infanzia Sylva”					1	1
“Library OR recreation center”			o3 “Biblioteca Coggetti”					2	0.7
			o5 “Centro Sportivo Diaz”					2	0.6
			o6 “Ludoteca Locatelli”					2	1
			o8 “Giardinetto Scuri”					2	0.3
Baby sitter			o9 “Sig.ra Clelia”					3	1
E Set of Edges									
$s\_rank$	o1	o2	o3	o4	o5	o6	o7	o8	o9
o1		1		1			0		
o2			0.90		0.83	0.77		0.83	
o3									1
o4			0.80		0.92	0.70		0.74	
o5									0.25
o6									0
o7									
o8									0.50
o9									

**Table 3.** Distances between pairs of resource locations and their satisfaction degrees of the spatial condition *close* defined by Formula (8).

Objects	Dist (Mt)	Closeness
o2 o3	165	0.90
o2 o5	311	0.83
o2 o6	436	0.77
o2 o8	317	0.83
o4 o3	378	0.80
o4 o5	132	0.92
o4 o6	636	0.70
o4 o8	520	0.74

**Table 4.** Routes identified by evaluating query (1), ranked based on decreasing RSV.

Ranked Routes	RSV
o1–o2–o3–o9	2.40
o1–o2–o6	1.77
o1–o2–o5–o9	1.75
o1–o2–o8–o9	1.45
o1–o4–o5–o9	1.05
o1–o4–o6	1.00
o1–o4–o8–o9	0.75
o1–o4–o3–o9	0.72

**Figure 3.** Query Graph determined using retrieved resources and VGS as a result of query (9).**Table 5.** Routes identified by evaluating query (9), ranked based on decreasing RSV.

Ranked Routes	RSV
o1–o2–o9–o3	2.7
o1–o2–o9–o8	2.3
o1–o2–o9–o5	2.2
o1–o4	0.3

### 3.2. SO-GREAT System Design and Implementation

#### 3.2.1. User Requirements for Strengthening Local Communities

Within the “Future Home for Future Communities” project ([www.FHfFC.it](http://www.FHfFC.it), accessed on the 28 February 2023), the objective of which was designing innovative solutions to improve the quality of life of local communities, a system for strengthening local communities was designed. It targets people with mobility difficulties bound to stay at home or close to their residence: examples of communities are those involving elderly people, people with disabilities, and families with children. A survey was conducted by submitting questionnaires to around 100 potential users living in the Lombardy region, northern Italy, in order to elicit their needs and desiderata for the kind of territorial resources and services they might be interested in searching and visiting. The results of the survey, besides revealing that 84% of the people search information by using a smart phone, made it possible to identify a set of resource categories of potential interest to them (public administrations, health, commerce, public transport, sport, education, leisure, parks, open Wi-Fi). Furthermore, some needs concerning the quality of the resources were expressed: it was considered

relevant if a resource was recognized by the public administration authority, i.e., was in the list of the open data of the regional authority, and if its reputation was high among those who had experienced it. For example, hospitals of the public health service that had a good reputation among patients as reported in their comments on social networks.

Additionally, the possibility to search and retrieve private advertisements offering services on voluntary basis having high ratings and the possibility to contact the volunteers privately was considered of interest.

Finally, a routing service allowing to explore convenient routes for visiting the relevant retrieved resources was also considered useful.

Three typical use cases of the system were identified for the three categories of users.

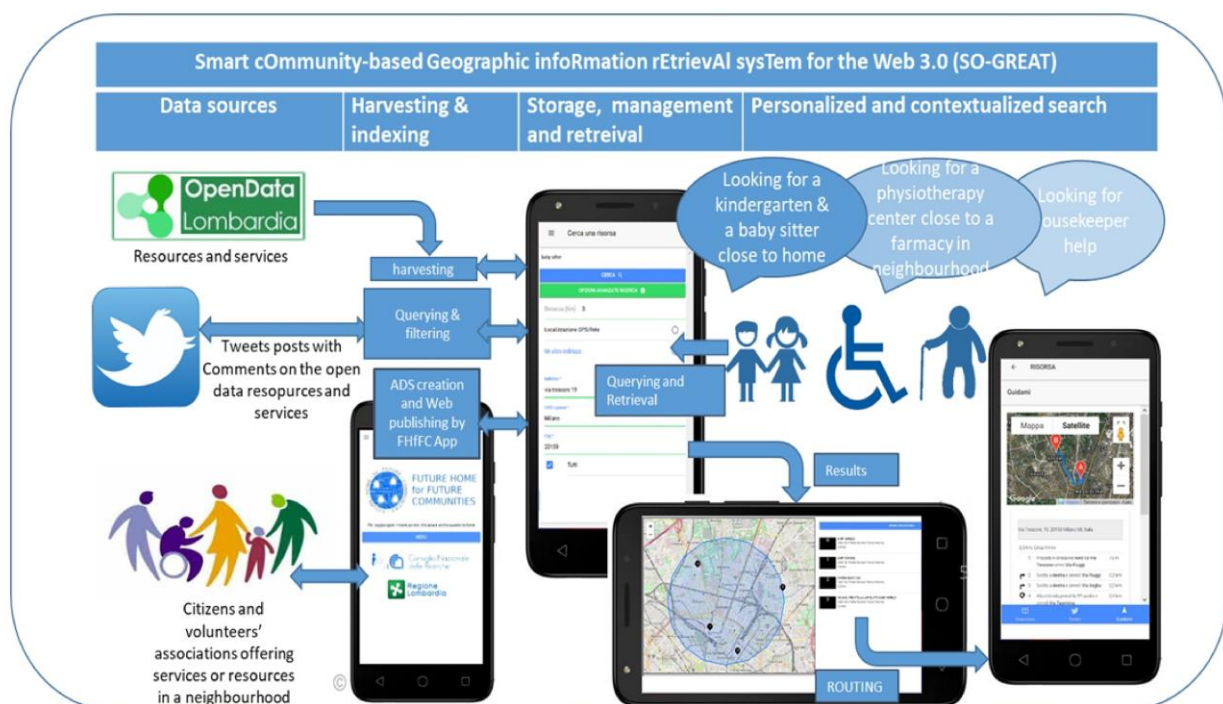
A family with children just moved to the city who had to choose both the kindergarten where they can register their children and a babysitter, who can collect children after lessons and accompany them to a recreational centre or library in the neighbourhood of the area where (s)he offers his/her service.

A person bound in rehabilitation at home who seeks a volunteer nurse nearby who can pass by a pharmacy to buy medicines before visiting him.

An elderly person looking for neighbouring peers to share a walk with them in a park and then a chat in a cafe located at a small walking distance.

### 3.2.2. The SO-GREAT System Architecture

To satisfy the above requirements the information workflow and functionalities of the Smart cOMmunity-based Geographic infoRmation rETrieval SysTem (SO-GREAT) system have been identified as depicted in Figure 4: from left to right, the system is capable to ingest heterogeneous documents from three distinct data sources, i.e., Open Data from the Lombardy regional authority, Tweets, and VGSs; it can index contents and manage and retrieve them in an integrated way on the basis of flexible queries that are interpreted by considering both the personal user interests and his/her spatial context.



**Figure 4.** Information workflow and functionalities of SO-GREAT system.

To design the system architecture, the following aspects of the resources have been considered:

- Heterogeneity of the resources and services as far as their semantics, structure, and formats go;
- Missing or implicit geolocation, often encoded by a postal address;
- Short descriptions of the resource facilities: in most cases just metadata are available, such as the type of resource, name, and postal address, while information on comments and ratings from users who experienced the resource/services can help evaluate their trustfulness.

As far as the functionalities go, the following have been included to meet user requirements:

- The possibility to enrich the descriptions of the resources and services with comments collected from social network posts; thus, a harvesting of posts about the resources was a preliminary function.
- The ability to enable the creation of VGSs by volunteers willing to offer some kind of help in their neighbourhood. This possibility can contribute to improve the reciprocal knowledge and cohesion of a local community. Nevertheless, it is necessary to preserve the privacy of both the authors and users to be compliant with privacy regulations in force.
- The possibility to search and retrieve both local open data resources and VGSs in an integrated way, so as to provide the comprehensive knowledge of both the authoritative resources and volunteers' services offered in the neighbourhood of a user.
- The possibility to express flexible queries expressing both user preferences and priorities in order to identify the most convenient route through a set of relevant retrieved resources and VGSs.
- The possibility to use the system from different mobile phones and tablets, thus asking for a cross-platform implementation also including mobile computing facilities such as off-line creation of VGSs and detection of geographic location by the GPS receiver.

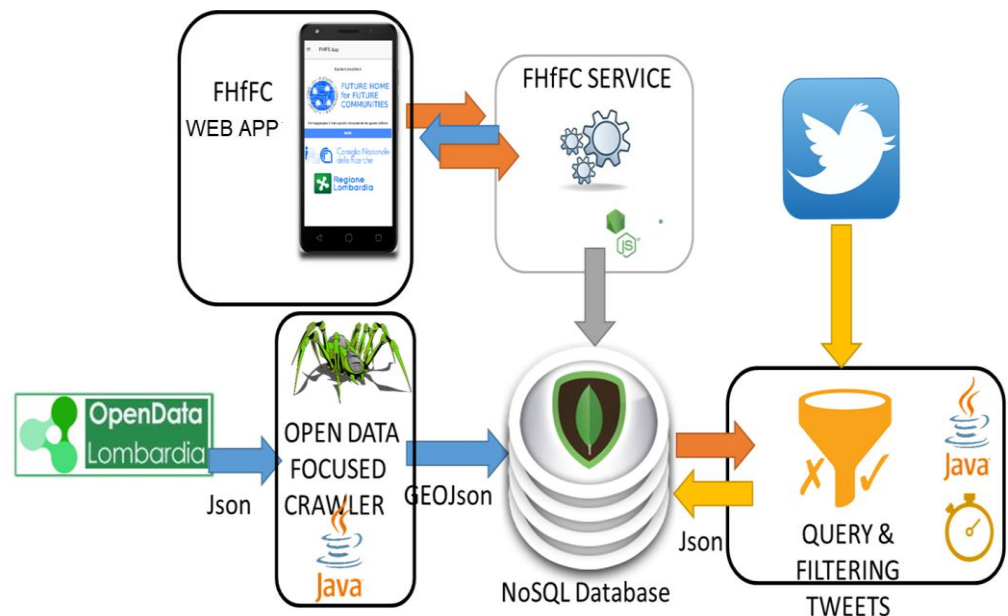
SO-GREAT has been implemented as a platform consisting of independent Web service components as depicted in Figure 5; it can be seen that the integration of the heterogeneous information is performed at the level of storage within a NoSQL database. The service components do not directly communicate with one another but exchange information through data in the common database. The sources and kind of documents and the service components developed to collect them are the following:

- Open Data from the Lombardy regional authority are harvested by a focused crawler developed in Java that can be customized to specific categories of users: in the case study, information was collected on resources and services of interest for families with children, elderly people, and people with disabilities. The user categories were associated with resource types based on the results of the requirement analysis. Lexical analysis and indexing in full-text were applied relying on Lucene library [29]. The mapping between user categories and resource types can be flexibly customized using a look-up table that can be configured before starting the harvesting, associating a set of resources' types of possible interest with each user category. Moreover, since the explicit geo-reference of resources is rarely available, it was necessary to apply geographic information retrieval indexing functions to identify it: geo-parsing was firstly performed by applying Name Entity Recognition techniques and Part of Speech Tagging to detect postal addresses [9] and, secondly, geocoding was performed to associate geographic coordinates using the "nominatim" geocoding functionality of the OpenStreetMap project [6].
- Tweets expressing comments on one of the collected open data resources or services are collected, indexed, and stored in the database so as to be able to retrieve them when an expansion of a retrieved resource or service is demanded by the user. This is an annotation approach, performed by running periodic queries with the names of the collected resources using the Standard Tweeter API searches against the free archive of recent Tweets [30]. The filtering functions were developed in Java so that posts retrieved using a resource name were selected if their georeference was close to



that of the resource. This was implemented in order to try to reduce ambiguities, since many resources have similar names and it was assumed reasonable that they were visited by users in their neighbourhood. All filtered Tweets associated with a given resource or service are considered as a unique document and are indexed in full text by adding a significance score based on the frequency of occurrences of terms.

- To create VGSs offered by registered citizens or voluntary associations the FHfFC Web application was developed in Ionic [31] an open-source mobile user interface toolkit for building cross-platform native web applications. Users willing to create a VGS must register and must indicate an area where they are willing to provide the service, a valid email address to be contacted by potentially interested users, and designate their as one of the available predefined types of services (babysitter, care-giver, nurse, house keeper, etc.). These VGSs may also contain free text and images and are categorized in predefined classes and geo-located to indicate the geographic scope where the service can be provided. The richer the description of the VGS is the more likely it is that its semantic relevance score is high when it is retrieved. Additionally, the more the VGS is commented in a filtered Tweet, the more information is available to the user to assess its trustfulness.



**Figure 5.** Service components of SO\_GREAT platform.

All information is stored in GeoJSON format and managed in a NoSQL database (MongoDB [32]). Querying and retrieval, resource content access and visualization, expansion with comments, and routing can be performed by means of the same Web application. If the user is registered, he/she will be able to specify his/her category (family, disabled or elderly) and thus the retrieval will be personalized to his needs and home location.

### 3.2.3. The Data Model and Find Routes Function

The data model is defined by a triple  $(UC, U, DB)$  in which the predefined set of user categories is defined as follows:

$$UC = \cup uc := \cup (v, \langle H_1, \dots, H_v \rangle, \delta). \quad (10)$$

where  $v \in \{family, disabled\ people, elderly\ people\}$  specifies the name of a user category;  $\langle H_1, \dots, H_v \rangle$  is the list of resources and VGS types that are of interest to the user category; and  $\delta$  (expressed in metres) defines the  $\delta$  parameter defining the broad boundary of a given geometry in definition (7), for the *in\_neighbourhood* operator computation. For example,

we can define different extents for the broad boundary depending on the user category as follows:

(family, <"kinder garden", "school", "recreation center for children", "library", "Childhood Social Offer Unit", "playground">, 500 mt);  
 ("disabled people", <"Rehabilitation Structure", "hospital", "library", "nursing service">, 200 mt );  
 (elderly people, <"recreation center for elderly", "hospital", "library", "park">, 500 mt);  
 (Any, <Any>, 5000 mt)

This way, an elderly person is willing to reach a resource from home at a maximum distance of 5000 mt, while a family with children can reach a place not further than 500 mt away, and a disabled person can reach a location at a maximum of 200 mt.

$$U = \cup u = \cup (ID, name, email, uc, q, home) \quad (11)$$

is a set of user profiles  $u$  in which: ID uniquely identifies a user with nickname  $name$ , a verified  $email$ , belonging to a category  $uc \in UC$ , with a query  $log\ q := \{q_i\}$  where  $q_i$  is a query as defined in (2), and a geographic location  $home$  that is a 2D geometry (a spatial index is created) that, together with  $u.uc.\delta$ , defines the neighbourhood where  $u$  is interested to find the mandatory resources, and/or to offer services in the cases when  $u$  creates a VGS.

Unregistered users have the following settings: ID = Null, name = email = Unknown, home = "Lombardy", uc = Any, q = Null.

Notice that  $home$  can be an area to preserve user privacy.

$$DB = \langle \cup H, \cup T_H, \cup o, \cup M \rangle \quad (12)$$

is a set of tuples in which  $h$  is the name of a type of the resources and VGSs, for example "Rehabilitation Structure", "Childhood recreation center", "Accredited hospice", "Community For Minors", "baby sitting service", "nursing service".  $T_H$  is a thesaurus of terms relevant for the  $H$  type. For example:  $T_{\text{"Rehabilitation Structure"}} = \{\text{disabled, muscular and skeletal rehabilitation, Physiotherapy, physiatrist, osteopath, rehabilitation gymnastics, postural gymnastics, ...}\}$ ;  $T_{\text{"Childhood recreation centers"}} = \{\text{sport center, playroom, playground, ...}\}$ .  $T_H$  is defined automatically by extracting significant terms from the information corpus of resources and services belonging to the  $h$  type, and can be revised manually.

$$o = \langle ID, owner, name, H, text, img, geo, M, Fs\_M \rangle \quad (13)$$

$o$  indicates a database object uniquely identified by an ID. It is either a resource or a service collected from the open data portal, in this case  $owner = "RL"$  that stands for "Lombardy region", or a VGS freely created by a registered user, in this case  $owner = u.ID$ ;  $name$  is a human interpretable string naming the resource or VGS, for example "Montessori school", "Volunteer Ms Clelia" (a textual index is created on this field);  $H$  is the type of the resource or VGS to which  $o$  belongs to, for example "baby sitter service";  $text$  is a textual description of the resource or VGS (a textual index is created on this field);  $img$  is an image;  $geo$  is a 2D geometry representing the geolocation of the resource or the geographic scope of the VGS (a spatial index is created on this field). The geometry of  $geo$  is either a convex regular polygon defined by a closed polyline or a point.

$M = \cup m$  indicates all the posts  $m$  collected from Twitter by a query specifying as selection conditions the name of the resource  $o.name$  and a distance of 50 km from its geolocation  $o.geo$ .

$m = \langle ID, hashtag, time, text, geo \rangle$  represents a Twitter post, uniquely identified by an ID, created at timestamp  $time$  in location  $geo$  with  $hashtag$  and content  $text$ .

$Fs\_M$  is a fuzzy set of terms extracted from  $M$  in which the degree represents the significance of the term;  $M$  is generated by selecting terms from  $M.hashtag \cup m.hashtag$  and  $M.text = \cup m.text$  with  $m \in M$  through applying lexical analysis (stopword removal and stemming and term weight computation by the usual  $tf/IDF$  using Lucene library).

The FHfFC Web application implemented in *node.js* framework enables the user interaction by instantiating five main functions:

- *Register\_User*: it allows a user to register by filling the information in the user profile. When registering, a user can choose the category of interest *uc*, can specify the geolocation of *u.home*, by drawing a polygon, and can specify a personal maximum distance *u.uc.δ*; alternatively one can accept to share the GPS location detected by the smart device that is used at run time as value of *u.home*. Having a personal profile, the query log is locally stored and used for auto completion;
- *Create\_VGS*: it allows a registered user to create VGSs. In this case, the user can create several VGSs of distinct type: it is required that the registered user has a valid email to create VGSs. In this case, *u.uc.δ* delimits the area around *u.home* in which the VGS can be performed;
- *Access\_Object*: it allows accessing the content of a retrieved object, i.e., to see the description of a resource or VGS;
- *Expand\_Object*: it allows expanding the content of an object with the content of the associated Tweet *M*. This way one can see the text of all comments on the selected resource or VGS;
- *Find\_Routes*: it allows a user to formulate a trip-planning query and to retrieve a ranked list of convenient routes for visiting the relevant retrieved objects:

$$Find\_Routes: U \times Q \rightarrow \wp(O)^K \times R^K \quad (14)$$

in which  $U := \cup u$  is a set of registered users,  $Q := \cup q$  is a set of queries of the form defined in (2); and  $\wp(O)$  is the power set of all objects in the collection  $O = \{o_1, \dots, o_z\}$ .

*Find\_Routes* is instantiated by a user *u* who specifies *q* with the aim of retrieving and ranking routes ordered by decreasing RSV as defined in Formula (6) and crossing possibly *N* objects.

*Find\_Routes* (*u*, *q*) parses query *q* from left to right so as to create a query graph that is visited in depth of first order so as to compute the partial RSV of each sub-branch at each intermediate vertex until the root is reached, where the final RSV of each route is computed as described in Section 2.4.

### 3.2.4. The Personalized Keyword Search

To generate a query graph, an atomic operation is the one performing a keyword search in DB to retrieve the candidate objects to generate vertexes in the graph.

Given a query keyword *kw* it is searched in DB index and the retrieved objects are filtered to select only those whose type *o.H* matches a type of interest for the user category *u.uc*, defined in the user profile: *u.uc*. $\langle H_1, \dots, H_k \rangle$ . This allows personalizing the search to categories of users. This matching is performed by expanding the terms in *o*. $\langle H_1, \dots, H_k \rangle$  with the terms in the associated thesauri  $\cup T_{Hi}$ :

$$select\ o \mid \rightarrow kw \subset (o.name \cup o.text \cup o.H.v) \wedge (o.H.v \cup (\cup T_H)) \cap (u.uc.\langle H_1.v, \dots, H_k.v \rangle) \neq \emptyset \quad (15)$$

#### Vertex creation

Each vertex of the graph corresponds to a retrieved object.

For each *o* selected by keyword *kw* with priority *p*, a vertex *v* is created if the geometry *o.geo* is reachable from an existing vertex of the graph  $v_{i,p-1}$  with priority *p*−1:

$$\exists v_{i,p-1} \in V \in G_q \mid s\_rank_{i,o} > 0 \rightarrow v = (o, p, r\_rank). \quad (16)$$

The semantic relevance score *r\_rank* of an object *o* is computed based on the frequency of *kw* within the fields *o.name*, *o.text*, *o.H.v*, and on the text of all associated posts *o.M.text*, so as to reward the resources with many comments.

Finally, the graph-based algorithm described in Section 2.4.4 is executed to identify and rank the routes.

#### 4. Discussion and Conclusions

In this paper, a novel semantics of local searches has been proposed aiming to perform flexible trip planning simplifying current practices in the GIR and LBS contexts.

The proposed approach makes it possible to avoid multiple iterations when expressing the wish to visit some interesting geo-resources via the most convenient route; it allows to express both priorities of the types of resources of interest and flexible selection conditions on their spatial relations using a single query. A graph-based aggregation algorithm defined here lets users identify resources relevant to the types of their interest, and ranks convenient routes crossing them on the basis of users' preferences.

The approach has been applied in the SO-GREAT system, a platform of Web services, designed with these purposes to help families and disabled and elderly people to find territorial resources and services in their neighbourhood based on personal preferences. In this system, by expressing flexible trip-planning queries, a better exploitation of local authoritative resources and services offered by volunteers is possible, taking into account specific constraints due to age and disabilities. With respect to other similar applications, SO-GREAT has the advantage to provide both trusted and authoritative information, which is a major concern when searching for some services such as health, education, and care givers. Besides the semantic relevance of the resources, computed based on their textual descriptions and by exploiting users' comments reported in Tweets, the SO-GREAT can better consider the spatial context, an important factor when selecting territorial resources for planning visiting itineraries.

A preliminary user evaluation of the platform from mobile devices performed by four different users indicated that it is simple and intuitive to use; the availability of comments reported in Tweets of the retrieved resources has been considered a useful feature to get an idea of their trust. A more complete evaluation is needed to assess the proposed approach and the results will be published in a subsequent paper.

This is a first proposal that opens new interpretations for trip-planning query semantics. For example, suitability analysis of places, environmental impact analysis, geo-resource allocation, and trip optimization, are currently performed by applying several spatial operations by a technician using a GIS. By reformulating these needs as flexible queries in a GIR context they could be performed even by non-experts.

In this perspective, local searches can be defined to exploit multidimensional information, semantic as well as spatial and temporal information, and flexible queries can be regarded as specifying multidimensional cost functions to rank resources, thus bridging the gap existing between IRSs and GISs, where currently, semantic cost functions are typically expressed by content-based queries and are evaluated by IRSs, while spatial cost functions are typically evaluated by means of GISs.

**Author Contributions:** Conceptualization, Gloria Bordogna and Luca Frigerio; implementation Luca Frigerio and Simone Lella; supervision, writing, Gloria Bordogna; testing, revision, Paola Carrara. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was partially supported by the project FHfFC—Future Home for Future Communities, jointly funded by CNR and Regione Lombardia—Italy (2017–2019) <http://www.fhffc.it/> (accessed on the 28 February 2023).

**Data Availability Statement:** The code developed can be downloaded at: <https://github.com/docHell> (accessed on the 28 February 2023); the web service is available at <https://www.fhffcapp.it/> (accessed on the 28 February 2023).

**Conflicts of Interest:** The authors declare no conflict of interest.



## References

1. Available online: [www.searchenginewatch.com/2014/05/07/google-local-searches-lead-50-of-mobile-users-to-visit-stores-study/](http://www.searchenginewatch.com/2014/05/07/google-local-searches-lead-50-of-mobile-users-to-visit-stores-study/) (accessed on 28 February 2023).
2. Available online: <https://nextdoor.com/> (accessed on 28 February 2023).
3. Jones, C.B.; Purves, R.S. Geographical information retrieval (editorial article). *Int. J. Geogr. Inf. Sci.* **2008**, *22*, 219–228. [\[CrossRef\]](#)
4. Purves, R.S.; Clough, P.; Jones, C.B.; Hall, M.H.; Murdock, V. Geographic Information Retrieval: Progress and Challenges in Spatial Search of Text. *Found. Trends Inf. Retr.* **2018**, *12*, 164–318. [\[CrossRef\]](#)
5. Reichenbacher, T.; De Sabbata, S.; Purves, R.S.; Fabrikant, S.I. Assessing geographic relevance for mobile search: A computational model and its validation via crowdsourcing. *J. Assoc. Inf. Sci. Technol.* **2016**, *67*, 2620–2634. [\[CrossRef\]](#)
6. Available online: [www.openstreetmap.org/about](http://www.openstreetmap.org/about) (accessed on 28 February 2023).
7. Bordogna, G.; Frigerio, L.; Rampini, A. Retrieval of visiting paths through relevant resources and services for enabling smart communities. In Proceedings of the 35th Annual ACM Symposium on Applied Computing, Brno, Czech Republic, 30 March–3 April 2020; pp. 714–716.
8. Huang, Y.W.; Jing, N.; Rundensteiner, E.A. Integrated query processing strategies for spatial path queries. In Proceedings of the 13th International Conference on Data Engineering, Birmingham, UK, 7–11 April 1997; pp. 477–486.
9. Bordogna, G.; Ghisalberti, G.; Psaila, G. Geographic information retrieval: Modeling uncertainty of user's context. *Fuzzy Sets Syst.* **2012**, *196*, 105–124. [\[CrossRef\]](#)
10. Bordogna, G.; Bovenzi, G.; Ghisalberti, G.; Psaila, G. Uncertainty Reduction in Location-Based Retrieval of Georeferenced Web Resources by Moving Users. In Proceedings of the Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM International Conference, Milan, Italy, 15–18 September 2009; pp. 163–166.
11. Li, F.; Cheng, D.; Hadjieleftheriou, M.; Kollios, G.; Teng, S.H. On Trip Planning Queries in Spatial Databases. In *Advances in Spatial and Temporal Databases, LNCS 3633*; Medeiros, C.B., Egenhofer, M., Bertino, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 273–290.
12. Doherty, A.R.; Gurrin, C.; Jones, G.J.F.; Smeaton, A.F. Retrieval of Similar Travel Routes Using GPS Tracklog Place Names. In Proceedings of the SIGIR GIR'06, Seattle, WA, USA, 10 August 2006.
13. Adelfio, M.D.; Samet, H. Itinerary Retrieval: Travelers, like Traveling Salesmen, Prefer Efficient Routes. In Proceedings of the 8th ACM SIGSPATIAL Workshop on Geographic Information Retrieval (GIR'14), Dallas, TX, USA, 4–7 November 2014.
14. Li, Y.; Yang, W.; Dan, W.; Xie, Z. Keyword-aware dominant route search for various user preferences. In Proceedings of the International Conference on Database Systems for Advanced Applications, Hanoi, Vietnam, 20–23 April 2015; pp. 207–222.
15. Zeng, Y.; Chen, X.; Cao, X.; Qin, S.; Cavazza, M.; Xiang, Y. Optimal route search with the coverage of users' preferences. In Proceedings of the 24th International Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015; pp. 2118–2124.
16. Li, W.; Cao, J.; Guan, J.; Yiu, M.L.; Zhou, S. Retrieving routes of interest over road networks. In Proceedings of the International Conference on Web-Age Information Management, Nanchang, China, 3–5 June 2016; pp. 109–123.
17. Li, W.; Cao, J.; Guan, J.; Yiu, M.L.; Zhou, S. Efficient Retrieval of Bounded-Cost Informative Routes. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2182–2196. [\[CrossRef\]](#)
18. Thatcher, J. From Volunteered Geographic Information to Volunteered Geographic Services. In *Crowdsourcing Geographic Knowledge: Volunteered Geographic Information (VGI) in Theory and Practice*; Sui, D.Z., Elwood, S., Goodchild, M.F., Eds.; Springer: Dordrecht, The Netherlands, 2013. [\[CrossRef\]](#)
19. Mountain, D.; MacFarlane, A. Geographic information retrieval in a mobile environment: Evaluating the needs of mobile individuals. *J. Inf. Sci.* **2007**, *33*, 515–530. [\[CrossRef\]](#)
20. Bordogna, G.; Pagani, M.; Pasi, G.; Psaila, G. Managing uncertainty in location-based queries. *Fuzzy Sets Syst.* **2009**, *160*, 2241–2252. [\[CrossRef\]](#)
21. Petry, F.E. *Fuzzy Databases*; Springer: Boston, MA, USA, 1996.
22. Kacprzyk, J.; Zadrozny, S.; Ziolkowski, A. FQUERY III+: A human-consistent database querying system based on fuzzy logic with linguistic quantifiers. *Inf. Syst.* **1989**, *14*, 443–453. [\[CrossRef\]](#)
23. Bosc, P.; Prade, H. An Introduction to the Fuzzy Set and Possibility Theory-Based Treatment of Flexible Queries and Uncertain or Imprecise Databases. In *Uncertainty Management in Information Systems*; Motro, A., Smets, P., Eds.; Springer: Boston, MA, USA, 1997.
24. Thulasiraman, K.; Swamy, M.N.S. *Graphs: Theory and Algorithms*; John Wiley & Sons: New York, NY, USA, 1992.
25. Pereira, C.; Dragoni, M.; Pasi, G. Multidimensional relevance: Prioritized aggregation in a personalized Information Retrieval setting. *Inf. Process. Manag.* **2012**, *48*, 340–357. [\[CrossRef\]](#)
26. Yager, R.R. Prioritized Aggregation operators. *Int. J. Approx. Reason.* **2008**, *48*, 263–274. [\[CrossRef\]](#)
27. Bordogna, G.; Psaila, G. Fuzzy-Spatial SQL. In *Flexible Querying Answering Systems*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 307–319.
28. Hu, Q.; Liu, Q.; Wang, X.; Tung, A.K.H.; Goyal, S.; Yang, J. DocRicher: An Automatic Annotation System for TextDocuments Using Social Media. In Proceedings of the SIGMOD/PODS'15: International Conference on Management of Data, Melbourne, Australia, 31 May–4 June 2015. [\[CrossRef\]](#)
29. Available online: <https://lucene.apache.org/core/> (accessed on 28 February 2023).
30. Available online: <https://developer.twitter.com> (accessed on 28 February 2023).

31. Available online: <https://ionicframework.com/> (accessed on 28 February 2023).
32. Available online: <https://www.mongodb.com/> (accessed on 28 February 2023).

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.