Automated Cleansing of POI databases

Guy De Tré, Daan Van Britsom, Tom Matthé, and Antoon Bronselaer

Abstract In the context of geographic information systems (GIS), points of interest (POIs) are descriptions that denote geographical locations which might be of interest for some user purposes. Examples are public transport facilities, historical buildings, hotels and restaurants, recreation areas, hospitals etc. Because information gathering with respect to POIs is usually resource consuming, the user community is often involved in this task. In general, POI data originate from different sources (or users) and are therefore vulnerable to imperfections which might have a negative impact on data quality. Different POIs referring to, or describing the same physical geographical location might exist. Such POIs are said to be coreferent POIs. Coreferent POIs must be avoided as they could harm the data(base) quality and integrity. In this chapter, a novel soft computing technique for the (semi-)automated cleansing of POI databases is proposed. The proposed technique consists of two consecutive main steps: the detection of collections of coreferent POIs and the fusion, for each collection, of all coreferent POIs into a single consistent POI that represents all the POIs in the collection. The technique is based on fuzzy set theory, whereas possibility theory is used to cope with the uncertainties in the data. It can be used as a component of (semi-)automated data quality improvement strategies for databases and other information sources.

1 Introduction

Geographic information systems are characterized by a tremendous amount of data, which must be collected, processed and represented in an efficient, user-friendly way. Moreover, some of these data must regularly be actualised as geographic ob-

Guy De Tré, Daan Van Britsom, Tom Matthé, and Antoon Bronselaer
Ghent University, Dept. of Telecommunications and Information Processing,
St.-Pietersnieuwstraat 41, B9000 Ghent (Belgium)
e-mail: {guy.detre, tom.matthe,daan.vanbritsom,antoon.bronselaer@UGent.be}

jects like roads, buildings or borderlines often change. A specific kind of information concerns the description of geographic locations or entities at geographic locations. In general, such information is modelled by objects which are called *points of interest* (POIs). Examples of POIs are objects that describe historical buildings, public services, hotels, restaurants and bars, panoramic views, interesting places to visit, etc. Usually, POIs contain information about location (coordinates) and a short textual description, but also other information such as the category the POI belongs to, multimedia like pictures and video and meta-data like the creator's name, the timestamp of creation, the file size, etc. can be provided.

In practice and due to their specific content, POI databases often contain data that are obtained from different heterogeneous sources, of which some might be maintained by user communities. User communities are often involved in data collection processes in cases where detailed, not commonly known data have to be inserted and maintained. When POIs originate from different sources or are entered by a user community, taking care of data consistency and correctness needs special attention. Indeed, such data are extremely vulnerable to errors, which might among others be due to uncertainty, imprecision, vagueness or missing information.

A problem that seriously harms the overall quality of a geographical information system (GIS) occurs when different POIs, denoting the same geographic entity, are inserted in the system. Such POIs are called *coreferent POIs*: they differ from each other, but all describe the same geographic location or object at a geographic location. Coreferent POIs can introduce uncertainty and inconsistency in the data, result in a storage and data processing overhead and moreover can cause low quality or, even worser, incorrect information retrieval results [26].

It is therefore important and relevant to develop techniques to *detect* coreferent POIs. Once detected, the problem of coreference has to be solved. Two basic approaches can be identified. In the first approach, the existence of coreferent POIs is *prevented* with techniques that, e.g., inform users about POIs that are detected to be in the neighbourhood of a new POI. As such, it is up to the user to check and verify whether the insertion makes sense. In the second approach, which is handled in more detail in this chapter, the responsibility for the correctness of the database is to a considerable extent shifted to the database management system. Coreferent POIs have to be *merged* (or *fused*) by the database management system into one single, consistent POI and the duplicates have to be removed. Perhaps, the simplest merging strategy is to keep one of the coreferent POIs and then remove all the others. As this simple merging strategy often introduces an information loss, more advanced merging techniques are required.

The research described in this chapter contributes to as well automatic detection, as automatic merging of coreferent POIs. The automatic detection of coreferent POIs has been approached as an uncertain Boolean problem. This means that two POIs are either coreferent or not (i.e., a boolean matter), but uncertainty about this decision must be dealt with. In order to determine this uncertainty, the POI structure is decomposed into elementary attributes (i.e., atomic sub objects). In this chapter it has been explicitly assumed that all POIs share the same structure. Thus, the issue of POI schema matching is not taken into account here. For each elementary attribute,

an elementary *evaluator* is allocated. Such an evaluator determines the uncertainty about the coreference of two values of the attribute's domain (which is the set of all allowed values for the attribute). The returned uncertainty is modelled by a possibilistic truth value [35, 18]. Because the POI's coordinates are among the main characteristics of a POI, special attention is paid to the detection of co-location of POIs, i.e., the definition of an appropriate evaluator for geographic coordinates. To obtain the overall uncertainty on the coreference of two POIs, the elementary evaluators are applied and their resulting possibilistic truth values are aggregated. For this aggregation, a variant of the Sugeno integral, as presented in [8] is used. The proposed merging approach for coreferent POIs uses the possibilistic truth values that are returned from the elementary evaluations and the aggregation, to determine how and which parts of two coreferent POIs should be merged to obtain a single deduplicated POI. Different strategies are described in the chapter.

The presented work also contributes to research on data quality issues in information retrieval by studying techniques that allow to automatically improve the data quality of information sources. Although applied in the context of geographic POI databases, the presented techniques can also be used and further extended for coreference detection and handling in other data sources and web sources. An improved data quality on its turn will automatically lead to better database query and information retrieval results. In cases where the data sources are read-only and hence can not be updated, coreference handling can be postponed until the data querying or information retrieval results are retrieved. Coreferent results can then be automatically filtered out and adequately handled before presenting them to the users.

The remainder of the paper is structured as follows. In Section 2, a brief overview of related work is given. Next, in Section 3, some preliminary definitions and notations with respect to objects and POIs are presented. Then, in Section 4 the problem of determining the uncertainty about the coreference of two POIs is dealt with. Herewith special attention is respectively paid to the definition of evaluators for atomic objects (in Subsection 4.1), the determination of the uncertainty about the co-location of two POIs in a two-dimensional space (in Subsection 4.2), and the computation of the overall uncertainty about the coreference of the overall POIs, i.e., the definition of aggregators for complex objects (in Subsection 4.3). Section 5, discusses the problem of merging two coreferent objects. Some general merge functions are described. These general functions allow one to develop a specific merge technique for POIs. The presented techniques for the detection and merging of coreferent POIs are illustrated in Section 6. Finally, in Section 7, some conclusions and indications for further work are given.

2 Related Work

Both the topics of coreference detection and of the merging of coreferent data have already been studied from different perspectives. In the next subsections we briefly give an overview of related work in these areas.

2.1 Coreference Detection

Coreference detection is already being studied since the late '60s, at which time it was commonly described as *record linkage*. A basic work on record linkage is [25]. Both traditional and fuzzy approaches exist.

In traditional approaches, coreference detection is typically done by means of a clustering method. An example is the DBSCAN algorithm [24]. When applying the DBSCAN algorithm to a POI database, clusters of coreferent POIs are expanded by adding similar POIs. Similarity between POIs is often determined by means of some multidimensional similarity measure, which is a weighted linear combination of spatial, linguistic and semantic measures. Spatial similarity is usually measured by calculating the distance between two POIs [34] and map this to inverse values in the interval $[0, 1]$, where 1 denotes an identical location and 0 represents the maximal distance. Linguistic similarity is usually measured by applying the Jaro-Winkler or another string comparison metric [29, 43] and semantic similarity can be computed by comparing the relative positions of the concepts under consideration in a taxonomic ontology structure [37].

In fuzzy approaches, the problem of detecting coreferent POIs is usually addressed by considering that duplicates are due to uncertainty and by explicitly handling this uncertainty by means of fuzzy set theory [47] and its related possibility theory [48, 21] (see, e.g., [40, 20]). Fuzzy ranges are then used to model spatial uncertainty about the co-location of two POIs. In [40], rectangular ranges are used, whereas in [20] context dependent circular ranges are proposed that are based on the scales of the maps in which the POIs are entered. In the remainder of this chapter, fuzzy set theory is used to further enhance spatial similarity measures so that these better cope with imperfections in the descriptions (of the locations) of the POIs. The problem of detecting co-location and merging of co-located data is also somewhat related to issues of conflation in GIS (see, e.g., [27]). Conflation is the complex process of combining information from two digital maps to produce a third map which is better than either of its component sources. In [36] the software agent technology paradigm has been applied as a conflation solution. Agent system techniques are hereby combined with expert system techniques to provide a feasible system architecture for distributed conflation.

2.2 Merging of Coreferent Data

The scientific foundations of POI merging lay in the research on information fusion, which deals with the combination of information provided by independent sources into one piece of information. The challenge hereby is to resolve inconsistencies between the different sources. An interesting aspect of information fusion is its applicability in many different contexts.

In a mathematical context, information fusion has led to the development of numerous aggregation operators such as generalized means [45, 23], t-norms and t-

conorms [22] and uninorms [46]. Aggregation operators fuse information that is represented as an element of a complete lattice $(L, \leq)$. The information typically expresses *facts*, for example the opinion or score of an agent. A flexible spatial data fusion approach based on a generalized ordered weighted averaging operator reflecting the concept of a fuzzy majority is presented in [16, 5]. Next to aggregation operators, a significant body of research deals with the case where deductive knowledge, such as inference rules and (integrity) constraints is used to combine information from different sources. Hereby, each source is considered to be a *propositional belief base* modelled as a first-order theory (see, e.g., [4, 1, 2, 31, 32, 30]). A typical difference between propositional belief bases and aggregation operators, is the presence of non-factual knowledge, such as inference rules and integrity constraints. As a consequence, the interest here is to *combine* all information in a maximal first-order theory. Such a setting occurs, amongst others, in heterogeneous databases [7]. A third type of information fusion deals with the case where each source provides knowledge by means of a possibility distribution (see, e.g., [38, 19]). In this case, it is assumed that the different sources have to cope with imprecision and/or incomplete knowledge and the key question is how uncertainty can be processed when dealing with different sources, that can provide conflicting information. Other approaches include heterogeneous data source fusion based on semantic rules (e.g., [33]) or ontologies (e.g., [6]).

Despite these related research areas, surprisingly the problem of merging coreferent data has not been as deeply investigated as the problem of coreference detection. An interesting overview of information combination operators for data fusion is given in [3]. In [12] the properties of object merging functions are investigated and a general framework for the merging of coreferent objects is proposed. In this paper we investigate and illustrate how this general framework can be applied in the context of POI merging.

3 Some Preliminaries

In this section we give some basic definitions and properties of objects and points of interest (POIs). These definitions form the formal basis for the techniques presented in the remainder of the chapter.

3.1 Basic Concepts on Objects

A fundamental concept in this chapter is that of an object. An *object* is axiomatically defined as a piece of data that describes an entity. A distinction is made between atomic and complex objects. Atomic objects are objects of which the universe is non compound, while complex objects belong to a universe O that is composed of non compound universes, i.e., $O = U_1 \times \cdots \times U_n$. The appropriate universe of

entities is denoted by $\mathscr{E}$ and the link between objects and entities is formalised by a surjective function $\rho : O \rightarrow \mathscr{E}$. Objects that refer to the same entity in $\mathscr{E}$ through ρ are said to be coreferent. Formally:

$$\forall(o_1, o_1) \in O^2 : (o_1 \leftrightarrow o_2) \Leftrightarrow (\rho(o_1) = \rho(o_2)). \tag{1}$$

The universe of an object is always equipped with a label function $l : O \rightarrow \mathscr{L}$, where $\mathscr{L}$ represents the appropriate set of labels. The label of a universe represents the class of entities that objects in the universe are describing. For example, consider $l(\mathbb{R}) =$'latitude', then we know that objects in $\mathbb{R}$ are describing entities of the class 'latitude', i.e., describe the geographic latitude coordinate of a location on the earth's surface.

In addition, complex objects are equipped with a tree structure in the sense that there exist logical groups of labels that belong together. For example, in objects that describe geographic entities, the universes with label 'street', 'house number', and 'postal code' form a logical group, i.e., the address. Formally, for a complex universe O and with the understanding that $\mathscr{P}(U)$ denotes the power set of U (i.e., the set of all subsets of U, including the empty set and U itself), there exists a function:

$$\lambda : \mathscr{P}(\{l(U_i)\}_{i=1\ldots n}) \rightarrow \{0,1\}. \tag{2}$$

such that λ indicates for each group of labels, whether these labels form a logical group or not. As the structure that corresponds to λ must be a tree structure, some constraints must be satisfied. The labels themselves must represent leaf nodes and the root node is given by the set of all labels, which means that:

$$\forall i \in \{1,\ldots,n\} : \lambda(\{l(U_i)\}) = 1 \tag{3}$$

$$\lambda(\{l(U_1),\ldots,l(U_n)\}) = 1. \tag{4}$$

Also, the parent child relation must be respected. In terms of λ, this means that for two arbitrary sets of labels, the following constraint must be satisfied:

$$(\lambda(A) = \lambda(B) = 1) \Rightarrow (A \subseteq B \vee B \subseteq A \vee A \cap B = \emptyset) \tag{5}$$

which states that two logical groups A and B are either connected through the ancestor relation or are disjoint.

3.2 Basic concepts on POIs

Reconsider the universe of entities $\mathscr{E}$. A *point of interest* (or POI) is axiomatically understood as a piece of data that describes a geographic entity in the real world that is modelled by $\mathscr{E}$. A POI is hence a special kind of complex object which is commonly used to describe an interesting location (or an entity at an interesting location).

By applying the function ρ that has been introduced in the previous subsection we obtain that two POIs POI_1 and POI_2 are coreferent, i.e., $POI_1 \leftrightarrow POI_2$ iff

$$(POI_1 \leftrightarrow POI_2) \Leftrightarrow (\rho(POI_1) = \rho(POI_2)). \tag{6}$$

Note that with the previous assumptions, we aim to keep the automated cleansing approach as general as possible and thus applicable to any data(base) model. The only requirements are that the data(base) model should support the modelling of complex objects which belong to a compound universe $O = U_1 \times \cdots \times U_n$ and for which there exists a label function l. The universe O is moreover equipped with a tree structure that is modelled by a function λ, which specifies logical groups of labels that belong together.

Example 1. An example of a compound universe that can be used to model POIs is

$$O_{POI} = U_1 \times U_2 \times U_3 \times U_4 \times U_5 \times U_6$$

where

$$\begin{aligned} U_1 &= S \\ U_2 &= S \\ U_3 &= [-90, 90] \\ U_5 &= [-180, 180] \\ U_5 &= S \\ U_6 &= C. \end{aligned}$$

Herewith, S is the set of all character strings and C is an enumerated list of allowed POI types. The label function l is specified as follows

$$\begin{aligned} &\forall u \in U_1 : l(u) = \text{identifier} \\ &\forall u \in U_2 : l(u) = \text{name} \\ &\forall u \in U_3 : l(u) = \text{latitude} \\ &\forall u \in U_4 : l(u) = \text{longitude} \\ &\forall u \in U_5 : l(u) = \text{description} \\ &\forall u \in U_6 : l(u) = \text{type} \\ &\forall o \in O_{POI} : l(u) = \text{POI}. \end{aligned}$$

The tree structure that is specified on O_{POI} is given by the function λ which is specified as follows (all subsets of labels that are not explicitly mentioned map to 0):

$$\lambda(\{l(U_1)\}) = \lambda(\{identifier\}) = 1$$
$$\lambda(\{l(U_2)\}) = \lambda(\{name\}) = 1$$
$$\lambda(\{l(U_3)\}) = \lambda(\{latitude\}) = 1$$
$$\lambda(\{l(U_4)\}) = \lambda(\{longitude\}) = 1$$
$$\lambda(\{l(U_5)\}) = \lambda(\{description\}) = 1$$
$$\lambda(\{l(U_6)\}) = \lambda(\{type\}) = 1$$
$$\lambda(\{identifier, name, latitude, longitude, description, type\}) = 1$$
$$\lambda(\{l(U_3), l(U_4)\}) = \lambda(\{latitude, longitude\}) = 1.$$

The tree structure corresponding to these mappings is presented in Figure 1. The semantics of the POIs under consideration can then be described as follows. The element of U_1 is the unique identifier of the POI, the element of U_2 is the name of the POI. The elements of U_3 and U_4 are connected to each other and together represent the location of the POI, which is given by a latitude and a longitude. Both latitudes and longitudes are expressed in decimal degrees (where 0.000001 degrees corresponds to 0.111 metre). The element of U_5 is a free description, provided by the user and modelled by full text. Finally, the element of U_6 is the type (or category) of the POI. It is assumed that this type is chosen from a given list. □

Because each POI is an element of a universe $O = U_1 \times \cdots \times U_n$, it can be denoted by a n-tuple $(u_1, \ldots, u_n)$, where $u_i \in U_i$, $i = 1, \ldots, n$.

Example 2. Reconsider the POI structure as introduced in Example 1. The following four 6-tuples are illustrations of POIs.

$(POI_1$, 'Friday market', 51.056934, 3.727112, 'Friday Market, Ghent', 'Market'$)$
$(POI_2$, 'St-Bavo', 51.053036, 3.727015, 'St-Bavo's Cathedral, Ghent', 'Church'$)$
$(POI_3$, 'Ghent cathedral', 51.053177, 3.726382, 'St-Bavo Cathedral', 'Cathedral'$)$
$(POI_4$, 'St-Bavo', 51.033333, 3.700000, 'St-Bavo – Ghent', 'Cathedral'$)$.

POI_2, POI_3 and POI_4 are examples of coreferent POIs. All four POIs have a different location. □

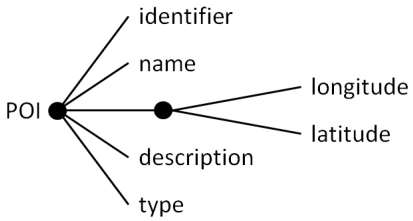

Fig. 1 Tree structure on labels corresponding to the mappings of Example 1.

4 Detection of Coreferent POIs

In this section, the problem of determining the uncertainty about the coreference of two POIs is dealt with. A possibilistic solution for finding coreferent objects consists of finding functions that express the uncertainty of coreference by means of *possibilistic truth values* [35, 42, 17, 18], which are possibility distributions over the Boolean domain $\mathbb{B} = \{T, F\}$. Thus, for a given Boolean proposition p, the possibilistic truth value (or PTV) $\tilde{p}$:

$$\tilde{p} = \left\{(T, \mu_{\tilde{p}}(T)), (F, \mu_{\tilde{p}}(F))\right\} \tag{7}$$

expresses the possibility that p is true (T) and the possibility that p is false (F). The domain of all possibilistic truth values is denoted $\mathscr{F}(\mathbb{B})$, i.e., the power set of normalised fuzzy sets over $\mathbb{B}$. In what follows, we shall adopt the couple shorthand notation for possibilistic truth values, i.e., $\tilde{p} = (\mu_{\tilde{p}}(T), \mu_{\tilde{p}}(F))$. Let us define the order relation $\geq$ on the set $\mathscr{F}(\mathbb{B})$ as follows:

$$\tilde{p} \geq \tilde{q} \Leftrightarrow \begin{cases} \mu_{\tilde{p}}(F) \leq \mu_{\tilde{q}}(F), \textbf{ if } \mu_{\tilde{p}}(T) = \mu_{\tilde{q}}(T) = 1 \\ \mu_{\tilde{q}}(T) \leq \mu_{\tilde{p}}(T), \textbf{ else} \end{cases} \tag{8}$$

An evaluator is a function that estimates a possibilistic truth value in order to express uncertainty about coreference [9].

Given a universe of objects O, an *evaluator* over O is defined as a function E_O:

$$E_O : O^2 \rightarrow \mathscr{F}(\mathbb{B}) \tag{9}$$

An evaluator compares two objects and yields a possibilistic truth value that expresses both the possibility that the objects are coreferent and the possibility that the objects are not coreferent. An evaluator is

- *Reflexive* if and only if:

$$\forall (o_1, o_2) \in O^2 : (o_1 = o_2) \Rightarrow (E_O(o_1, o_2) = (1, 0)) \tag{10}$$

- *Strong reflexive* if and only if:

$$\forall (o_1, o_2) \in O^2 : (o_1 = o_2) \Leftrightarrow (E_O(o_1, o_2) = (1, 0)) \tag{11}$$

- *Commutative* if and only if:

$$\forall (o_1, o_2) \in O^2 : E_O(o_1, o_2) = E_O(o_2, o_1) \tag{12}$$

In what follows, evaluators are always assumed to be commutative and at least reflexive. Finally, an evaluator is called *transitive* if and only if, for every triplet $(o_1, o_2, o_3) \in O^3$:

$$1-\mu_{E_O(o_1,o_3)}(F) \geq \min\left(1-\mu_{E_O(o_1,o_2)}(F), 1-\mu_{E_O(o_2,o_3)}(F)\right)$$
$$1-\mu_{E_O(o_1,o_3)}(T) \geq \min\left(1-\mu_{E_O(o_1,o_2)}(F), 1-\mu_{E_O(o_2,o_3)}(T)\right)$$
$$1-\mu_{E_O(o_1,o_3)}(T) \geq \min\left(1-\mu_{E_O(o_1,o_2)}(T), 1-\mu_{E_O(o_2,o_3)}(F)\right).$$

In the next subsections we successively describe evaluators for atomic objects, evaluators for determining co-location and evaluators for complex objects.

4.1 Elementary Evaluators for Atomic Objects

When it comes to the evaluation of atomic objects (i.e., objects with a non compound universe), some existing approaches are useful in the detection of coreferent POIs. More specifically, the comparison of character strings and numerical data has already been studied extensively and is the basis for the development of general purpose evaluators for character strings and numerical data. Such evaluators are briefly introduced in the next subsections.

4.1.1 Evaluators for Character Strings

First, *syntactical* evaluators have been proposed. These evaluators allow for the comparison of two character strings, taking into account the occurrence of spelling errors, abbreviations, ... [10, 13]. Hereby, strings are decomposed into a multiset of substrings. These multisets are then compared such that similarities between elements are taken into account [9]. The evaluators are called 'syntactical', because they decide upon coreference of two objects by comparing the syntactical construction of objects. Syntactical evaluators for strings are for example well suited for comparison of POI names and descriptions.

Secondly, *semantical* evaluators have been proposed [11]. As opposed to syntactical evaluators, semantical evaluators reject the idea that a decision of coreference must be based on a syntactical similarity between two objects. Instead, it accepts the fact that the existence of some (semantical) relationship between two objects can be sufficient to decide that these objects are coreferent. Examples of such relationships are the synonym relationship, the specification/generalization relationship, ... In [11], an approach is proposed for the dynamical discovery of (semantical) relationships between objects. In the case of POIs, semantical evaluators are well suited for the comparison of POI types.

4.1.2 Evaluators for Numerical Data

Evaluators for character strings can also be used for coreference detection of numerical data too. Indeed, coreferent numerical values refer to the same number, but can differ from each other due to typing errors or uncertainty. A typical example are telephone numbers or bank account numbers. In such cases, depending on the context in which the numbers are used, either syntactical and/or semantical evaluators can be applied for coreference detection.

Correferency of numerical data can also be due to imprecision. In such a case the difference between two numbers can be used as the basis for evaluation. If two numbers a and b are close enough, i.e., if $|a-b| \leq \varepsilon$, then a and b can be considered as being coreferent, else they are not considered to be coreferent. Hereby, ε acts as a threshold value and depends on the application under consideration. An example on how the value of ε can be determined is given in the next subsection.

4.2 Evaluators for Co-Location

Next to these general purpose evaluators described in the previous subsection, the case of POIs requires some case-specific evaluators for the comparison of locations. More details on these evaluators are discussed below.

Perhaps the most important aspect of a POI is its registered geographic location. POI's are considered to be zero dimensional objects, whereas geographic entities in the real world are generally two or three dimensional objects and hence can be denoted by multiple locations. Consider for example all locations of the surface of a bridge, park or lake or all locations in a building. To construct the POI, one of these locations has to be chosen as the representative location (or point). The location of a POI is hence, due to its nature, already very vulnerable to imprecision what is one of the main causes for coreferency. Beside of this inherent imprecision, coreferent POIs can also be assigned to different locations due to uncertainty or a lack of information.

In the remainder of this subsection a soft technique for estimating the uncertainty about the co-location of two POIs is presented. First, a basic technique commonly used in fuzzy geographic applications is presented. Secondly, this basic technique is further enhanced in order to explicitly cope with the scale at which the POI is entered by the user.

4.2.1 Basic Technique

The geographic location of a POI is usually modelled in a two-dimensional space by means of a latitude lat and longitude lon, as has been illustrated in Example 1. Consider two POIs POI_1 and POI_2 with locations (lat_1, lon_1) and (lat_2, lon_2) respectively. In geographic applications, the distance (in metres) between the two lo-

cations is usually approximately computed by

$$d(POI_1, POI_2) = 2R\arcsin(h) \tag{13}$$

where $R = 6367000$ is the radius of the earth in metres and

$$h = \min\left(1, \sqrt{\sin^2\left(\frac{lat_2^r - lat_1^r}{2}\right) + \cos(lat_1^r)\cos(lat_2^r)\sin^2\left(\frac{lon_2^r - lon_1^r}{2}\right)}\right)$$

with $lat_j^r = \frac{\pi}{180} lat_j$ and $lon_j^r = \frac{\pi}{180} lon_j$, for $j = 1,2$, being the conversions in radians of lat_j and lon_j [39]. The higher the precision of the measurement of the latitude and longitude, the higher the precision of this distance.

From a theoretical point of view, POIs are considered to be geographic locations. Hence, two POIs are considered to be co-located if their distance equals zero. In practice however, one has to deal with imperfect positioning specifications of locations. Therefore, it is more realistic to consider two POIs as being co-located if they refer to the same area and are thus *close enough*. In traditional approaches *'close enough'* is usually modelled by a threshold $\varepsilon > 0$, such that two POIs POI_1 and POI_2 are ε-close if and only if

$$d(POI_1, POI_2) \leq \varepsilon. \tag{14}$$

The problem with such a single threshold is that it puts a hard constraint on the distance, which implies an 'all or nothing' approach: depending on the choice for ε, two POIs will be considered as being co-located or not. If an inadequate threshold value is chosen, this will yield in a bad decision. A single threshold neither offers the flexibility to use different criteria in different contexts.

Fuzzy sets [47] have been used to soften the aforementioned hard constraint. In general, a fuzzy set with a membership function $\mu_{\varepsilon-close}$, as presented in Figure 2, is used to model '*close enough*'. This membership function is defined by

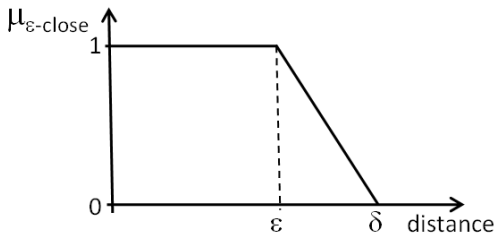

Fig. 2 Fuzzy set with membership function $\mu_{\varepsilon-close}$ for representing 'close enough'.

$$\mu_{\varepsilon-close} : [0,+\infty] \rightarrow [0,1]$$
$$d \mapsto \begin{cases} 1 & \text{, if } d \leq \varepsilon \\ \dfrac{\delta - d}{\delta - \varepsilon} & \text{, if } \varepsilon < d \leq \delta \\ 0 & \text{, if } d > \delta. \end{cases} \quad (15)$$

The extent to which two POIs POI_1 and POI_2 are considered to be co-located is then given by $\mu_{\varepsilon-close}(d(POI_1, POI_2))$. Hence, for distances below ε, $\mu_{\varepsilon-close}$ denotes co-location, for distances larger than δ no co-location is assumed, whereas for distances between ε and δ, there is a gradual transition from co-location to no co-location. Other membership function definitions can be used.

4.2.2 Enhanced Technique

A practical problem with fuzzy approaches as described above, is that the membership function has to reflect reality as adequate as possible. This implies that adequate values for ε and δ must be chosen. Values that are too stringent (too small) will result in false negatives, i.e., some POIs will falsely be identified as not being co-located, whereas values that are too soft (too large) will result in false positives, i.e., some POIs will falsely be identified as being co-located. In this subsection, it is considered that different POIs can originate from different sources or users. Such a situation often occurs in practical cases where data of different origins have to be collected and combined. Under this consideration, it makes sense to study how the parameters ε and δ are influenced by the *context* in which the POI has been originally registered. Eq. (15) can then be further enhanced in order to better reflect the imperfection and the context of the placement of the POI.

In practice, the exact coordinates of the location of a POI will not always be known. In such a case, the location of the POI has to be approximated. When user communities are involved in the construction and maintenance of a POI database, users might be asked to denote the position of the POI on a map. User communities are often involved when the content of the database changes regularly, which is for example the case with locations of speed control devices, locations that denote dangerous road conditions, and locations that denote interesting points to visit during walking or cycling activities.

If POI locations are entered via geographic maps the quality of the data will to some extent depend on the context the user is working in. Next, we focus on two aspects of this work context, namely scale and precision, and show how explicitly coping with these can help to improve Eq. (15).

If users work with maps on computer screens or screens of mobile devices when entering or maintaining (locations of) POIs, they work with a representation of (a part of) the real world that is drawn at a specific *scale* $(1 : s)$, which means, e.g., that 1 cm on the scale corresponds to s cm in reality. For example, a map of Europe on a computer screen can be drawn at scale $(1 : 15000000)$, a map of Belgium at scale $(1 : 1000000)$ and a map of Ghent at scale $(1 : 125000)$. It is clear that the

precision with which a user can place a POI on a map depends on the scale of the map. Denoting a POI that represents the Eiffel tower on a map of Europe will be less precise than on a map of France, which on its turn will be less precise than on a map of Paris. On the other hand, depending on his or her knowledge about the location of the new POI the user can zoom-in or zoom-out on the map to enter the POI at the map with the most appropriate detail for the user. Considering the different scales used in the different sources or used by different users, a scale $(1 : s_{min})$ corresponding to the most detailed level and a scale $(1 : s_{max})$ corresponding to the least detailed level can be determined. Hence, all occurring scales $(1 : s)$ will be within the range $(1 : s_{min}) \leq (1 : s) \leq (1 : s_{max})$.

Another aspect to take into account is the *precision* with which the user can denote the location of a POI on the screen. Usually, when working at an appropriate scale $(1 : s)$, the user will be able to place a point on the screen with a precision of a couple of centimetres, i.e., the exact location of the point will be within a circle with the denoted point as centre and radius d_s. This radius can be considered to be a parameter that depends on the scale $(1 : s)$ and the user's abilities for accurately denoting the POI on the screen. Therefore, in practical applications, d_s could be adjustable by the user or by a user feedback mechanism.

The scales $(1 : s)$, $s_{min} \leq s \leq s_{max}$, and corresponding radiuses d_s can now be used to further enhance the definition of the membership function $\mu_{\varepsilon-close}$ that is used in Eq. (15).

Estimating the Value of ε

In order to better approach reality, ε should reflect the maximum distance for which two POIs are indistinguishable and hence must be considered as being co-located.

If no further information about the geographical area of the POI is available, then the POI is positioned at the location that is entered by the user and modelled by its latitude and longitude. Two POIs are then indistinguishable if they are mapped to the same latitude and longitude. The maximum precision can be approximated by the dot pitch of the screen and be used to estimate the value of ε. The dot pitch d_p of a screen is defined as the diagonal distance between two pixels on the screen and usually has a standard value of 0.28mm. Considering the minimum scale $(1 : s_{min})$, the value of ε can then be approximated by

$$\varepsilon = d_p s_{min}. \tag{16}$$

If information about the geographical area of the POI is given, then the length l of the diagonal of the minimum bounding rectangle that surrounds this area can be used to approximate ε. Indeed, all POIs that are placed in the rectangle can reasonably be considered as being co-located. If the POI location of POI_1 and POI_2 is respectively entered at a scale $(1 : s_1)$ and $(1 : s_2)$, the value of ε can be approximated by

$$\varepsilon = \max(\frac{l}{2}s_1, \frac{l}{2}s_2) \tag{17}$$

where the maximum operator is used to take the roughest, largest approximation (which is due to the least precise scale) in cases where both POIs were entered at a different scale.

Estimating the Value of δ

Taking into account the scale $(1 : s_1)$ and precision d_{s_1} with which a user entered POI_1 and the scale $(1 : s_2)$ and precision d_{s_2} with which POI_2 was entered, the value of δ can be defined by

$$\delta = \varepsilon + \max(s_1 d_{s_1}, s_2 d_{s_2}) \tag{18}$$

where the maximum operator is again used to take the roughest approximation in cases where both POIs were entered at a different scale. With this definition the precisions d_{s_1} and d_{s_2} are handled in a pessimistic way. Alternative definitions for δ are possible.

4.2.3 Evaluator for Co-Location

The membership function $\mu_{\varepsilon-close}$ can now be used to define an evaluator E_{loc} for the determination of co-location. Such an evaluator should satisfy Eq. (9) and hence result in a PTV, expressing the uncertainty about the colocation of two locations of POIs.

A proposal for a simple definition for E_{loc} is

$$\begin{aligned} E_{loc} : ([-90,90] \times [-180,180])^2 &\to \mathscr{F}(\mathbb{B}) \\ ((lat_1, lon_1), (lat_2, lon_2)) &\mapsto (\mu_{\tilde{p}}(T), \mu_{\tilde{p}}(F)) \end{aligned} \tag{19}$$

where the membership grades $\mu_{\tilde{p}}(T)$ and $\mu_{\tilde{p}}(F)$ are defined by

$$\mu_{\tilde{p}}(T) = \frac{\mu_{\varepsilon-close}(d)}{\max(\mu_{\varepsilon-close}(d), 1 - \mu_{\varepsilon-close}(d))} \tag{20}$$

$$\mu_{\tilde{p}}(F) = \frac{1 - \mu_{\varepsilon-close}(d)}{\max(\mu_{\varepsilon-close}(d), 1 - \mu_{\varepsilon-close}(d))}. \tag{21}$$

where the distance $d = d((lat_1, lon_1), (lat_2, lon_2))$ is computed using Eq. (13) and the membership function $\mu_{\varepsilon-close}$ is defined by Eq. (15) with the parameter values ε and δ being estimated as described above. An example of the use of the evaluator E_{loc} is given in Section 6.

The evaluator E_{loc} can be used as a component of a technique to determine whether two POIs are coreferent or not. The resulting PTVs as obtained by Eq. (20) and (21), then denote a measure for the uncertainty about the co-location or spatial similarity of the POIs.

4.3 Evaluators for Complex Objects

Once atomic objects have been compared, a comparison of complex objects can be performed by aggregating the results of atomic comparisons. For that purpose, an extension of the Sugeno integral to the domain of PTVs has been proposed [8].

This integral uses two fuzzy measures (γ^T and γ^F). The measure γ^T (resp. γ^F) provides the conditional necessity that two complex objects are (not) coreferent, given that some set of attributes are (not) coreferent. In the case of POIs, $\gamma^T(\{\text{'name'}, \text{'type'}\})$ is a number in the unit interval that represents the necessity that two POIs are coreferent, provided that their names and types are coreferent. Similarly, $\gamma^F(\{\text{'name'}, \text{'type'}\})$ is a number in the unit interval that represents the necessity that two POIs are not coreferent, provided that their names and types are not coreferent. As required by the definition of fuzzy measures, γ^T and γ^F are normalised between $\emptyset$ and $\mathscr{L}$ and are monotonic.

It is noted that the fuzzy measures can be used to take structural information of objects into account. For example, it can be reflected in γ^T and γ^F that street, zip code and city together constitute an address by introducing dependencies between these atomic objects. This can be easily automated by usage of the function λ as introduced by Eq. (2).

The Sugeno integral introduced in [8] combines conditional necessity (γ^T and γ^F) with marginal necessity (the PTVs obtained from atomic comparison) into one PTV that reflects the uncertainty about the fact that two complex objects are coreferent. The inference used for this combination is purely possibilistic in nature and is therefore a valid and well suited aggregation method for PTVs in the case of coreference.

With the understanding that $\tilde{P}$ denotes a finite set of PTVs $\tilde{P} = \{\tilde{p}_1, \ldots, \tilde{p}_n\}$, the Sugeno integral of $\tilde{P}$ with respect to γ^T and γ^F is defined by

$$S_{\gamma^{T,F}}(\tilde{P}) : \mathscr{F}(\mathbb{B})^n \to \mathscr{F}(\mathbb{B}) : \tilde{P} \mapsto \tilde{p} \tag{22}$$

so that

$$\begin{aligned}
\mu_{\tilde{p}}(T) &= Pos_{\tilde{p}}(T) \\
&= 1 - Nec_{\tilde{p}}(F) \\
&= 1 - \bigvee_{i=1}^{n} Nec\left(\tilde{P}_{(i)^F} = F\right) \wedge \gamma^F\left(\tilde{P}_{(i)^F}\right) \\
&= 1 - \bigvee_{i=1}^{n} \left(\min_{\tilde{p} \in \tilde{P}_{(i)^F}} (1 - \mu_{\tilde{p}}(T)) \right) \wedge \gamma^F\left(\tilde{P}_{(i)^F}\right)
\end{aligned}$$

and

$$\begin{aligned}\mu_{\tilde{p}}(F) &= Pos_{\tilde{p}}(F)\\ &= 1 - Nec_{\tilde{p}}(T)\\ &= 1 - \bigvee_{i=1}^{n} Nec\left(\tilde{P}_{(i)^T} = T\right) \wedge \gamma^T\left(\tilde{P}_{(i)^T}\right)\\ &= 1 - \bigvee_{i=1}^{n} \left(\min_{\tilde{p}\in\tilde{P}_{(i)^T}} (1-\mu_{\tilde{p}}(F))\right) \wedge \gamma^T\left(\tilde{P}_{(i)^T}\right)\end{aligned}$$

where $\cdot_{()^T}$ and $\cdot_{()^F}$ are permutations on the elements of $\tilde{P}$. With the understanding that $\tilde{p}_{(i)^T}$ (resp. $\tilde{p}_{(i)^F}$) denotes the ith element of the permutation $\cdot_{()^T}$ (resp. $\cdot_{()^F}$) and that $\leq$ is the order relation for PTVs as defined by Eq. 8, the permutations $\cdot_{()^T}$ and $\cdot_{()^F}$ are defined as follows:

$$\forall i \in \{1,\ldots,n-1\} : \tilde{p}_{(i+1)^T} \leq \tilde{p}_{(i)^T}. \tag{23}$$

In other words $\cdot_{()^T}$ is a permutation that orders the elements of $\tilde{P}$ according to largest PTV first. Furthermore the permutation $\cdot_{()^F}$ on the elements of $\tilde{P}$ is defined by

$$\forall i \in \{1,\ldots,n-1\} : \tilde{p}_{(i)^F} \leq \tilde{p}_{(i+1)^F}. \tag{24}$$

This is the reciproque permutation of $\cdot_{()^T}$.

More details about (the use of) the Sugeno integral can be found in [8].

Because a POI is considered to be a special kind of a complex object, the evaluators for complex objects can be used to determine the PTV expressing the overall uncertainty that two POIs are coreferent or not. This will be illustrated in Section 6.

5 Merging of Coreferent POIs

Once coreferent POIs are detected, their duplicate information should be removed and their non-duplicate information should be merged. The challenges hereby are to avoid information loss and to resolve the inconsistencies that might exist among the different coreferent data.

A general *merge function* for coreferent objects of a universe O has been formally defined by

$$\varpi_O : \mathscr{M}(O) \to O \tag{25}$$

where $\mathscr{M}(O)$ denotes the set of all multisets drawn from the universe O [13, 15]. The merge function thus takes a multiset of objects and produces one single object as a result. As proposed by Yager [44], a multiset M over O is hereby characterized by a counting function $\omega_M : O \to \mathbb{N}$. For $v \in O$, $\omega_M(v)$ then represents the number of times that v occurs in M.

In the next Subsections 5.1 and 5.2, specific merge functions for atomic and complex objects will be defined. These functions will then be further fine-tuned for

the purpose of POI merging in Subsection 5.3. More information on the properties of the proposed functions is given in [13].

5.1 Merge Functions for Atomic Objects

Let us first introduce merge functions ϖ_U where U is a non compound universe. Recall that the context in which ϖ_U is to be used, is that of coreference. As such, we can assume that upon merge time, an evaluator E_U is available. Let M be a multiset of coreferent objects that are identified by a coreference detection framework. Then, for each object $u \in M$, $|M| = \sum_{u \in U} \omega_M(u)$ PTVs can be calculated by comparing u with all objects in M. Due to reflexivity of E_U, the PTV $(1,0)$ occurs at least $\omega_M(u)$ times. As such, for each object $u \in M$ a collection of PTVs is obtained where each $\tilde{p}$ indicates the uncertainty about the proposition that two objects are coreferent. In [28], a method is proposed to construct a possibility distribution $\pi_{\mathbb{N}}$ (a fuzzy integer) from a collection of PTVs. Hereby, $\pi_{\mathbb{N}}(k)$ indicates the possibility that exactly k propositions are true. Hence, for each element $u \in M$, a possibility distribution $\pi^u_{\mathbb{N}}$ can be constructed, where $\pi^u_{\mathbb{N}}(n)$ represents the possibility that 'exactly n values in M are coreferent with u'.

The method described in [28] has been used for the construction of a confidence based merge function as it allows to express the uncertainty about the number of coreferent objects according to a given evaluator E_U. It works as follows. Let P be a set of independent Boolean propositions and let $\tilde{P}$ be the multiset of corresponding PTVs which results from the evaluation of the proposition in P. Then, the quantity of true propositions in P is modelled by the possibility distribution $\pi_{\mathbb{N}}$ such that:

$$\begin{aligned}\pi_{\mathbb{N}}(k) = \min\big(&\sup\big\{\alpha \in [0,1] | |\{p \in P | \mu_{\tilde{p}}(T) \geq \alpha\}| \geq k\big\}, \\ &\sup\big\{\alpha \in [0,1] | |\{p \in P | \mu_{\tilde{p}}(F) < \alpha\}| \geq k\big\}\big). \end{aligned} \tag{26}$$

Eq. (26) states that the possibility $\pi_{\mathbb{N}}(k)$ is the minimum of the possibility that at least k propositions are true and the possibility that at most $|P| - k$ propositions are false. The possibility $\pi_{\mathbb{N}}(k)$ can be efficiently calculated by adopting the following notations. For a multiset $\tilde{P}$, let $\tilde{p}_{(i)}$ denote the i^{th} largest PTV with respect to the order relation defined in Eq. (8). The following then holds:

$$\pi_{\mathbb{N}}(k) = \begin{cases} \mu_{\tilde{p}_{(k)}}(F) & , \textbf{ if } k = 0 \\ \mu_{\tilde{p}_{(k)}}(T) & , \textbf{ if } k = |M| \\ \min\Big(\mu_{\tilde{p}_{(k)}}(T), \mu_{\tilde{p}_{(k+1)}}(F)\Big) & , \textbf{ else} \end{cases} \tag{27}$$

Figure 3 shows two example multisets, each consisting of five PTVs $(\mu_{\tilde{p}}(T), \mu_{\tilde{p}}(F))$, where $\circ$ denotes the possibility $\mu_{\tilde{p}}(T)$ of T and $\times$ denotes the possibility $\mu_{\tilde{p}}(F)$ of F. The derived possibility distributions $\pi_{\mathbb{N}}$, computed using Eq. (27), are shown below the PTVs. Note that the membership functions of the derived fuzzy integers $\pi_{\mathbb{N}}$ are always *convex*.

Applying this method allows us to express the number of coreferent objects, *according to* the evaluator E_U. Hence, although we already know that objects in M are coreferent, the distributions $\pi_{\mathbb{N}}$ express the uncertainty about this statement, at least, according to the evaluator E_U. Based on these observations, a merge function can be defined, considering that the result of the merging should be the object which has the highest number of coreferent objects according to E_U. We then obtain a merging technique where the uncertainty model of E_U is used to choose the best representative.

For this purpose, a method for comparing fuzzy integers is required. Many methods have been proposed. The most common technique is to *defuzzify* the fuzzy integer, for example by means of the center of gravity [22]. Fuzzy integers are then compared by comparing the results of defuzzification. The method that we shall adopt here, is not based on defuzzification, but is rather possibilistic in nature. We propose two order relations for fuzzy integers, one constructed from the viewpoint of possibility and one constructed from the viewpoint of necessity.

For two fuzzy integers, $\tilde{n}$ and $\tilde{m}$, the sup-order relation $\prec_{\text{sup}}$ is defined by

$$\tilde{n} \prec_{sup} \tilde{m} \Leftrightarrow \sup \tilde{n}_\alpha < \sup \tilde{m}_\alpha. \tag{28}$$

Hereby, $\tilde{n}_\alpha$ is the α-cut of $\tilde{n}$ where α is chosen such that:

$$\alpha = \sup\{x | \sup \tilde{n}_x \neq \sup \tilde{m}_x\}.$$

Also, for two fuzzy integers, $\tilde{n}$ and $\tilde{m}$, the inf-order relation $\prec_{\text{inf}}$ is defined by

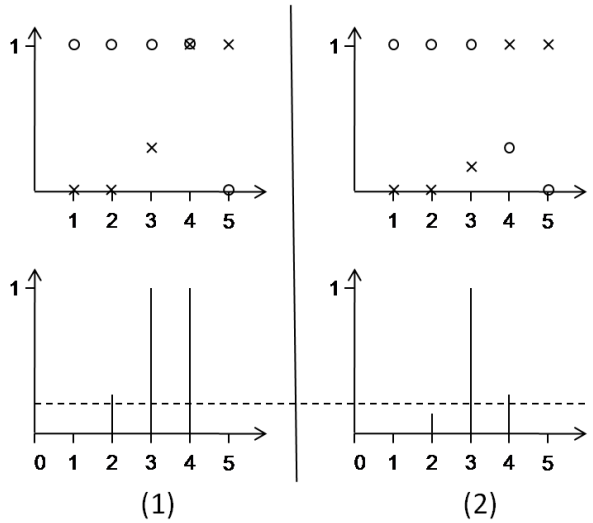

Fig. 3 Two example sets of five PTVs $(\mu_{\tilde{p}}(T), \mu_{\tilde{p}}(F))$ where $\circ$ and $\times$ respectively denote $\mu_{\tilde{p}}(T)$ and $\mu_{\tilde{p}}(F)$ (top) and their corresponding derived fuzzy integer (bottom).

$$\tilde{n} \prec_{\text{inf}} \tilde{m} \Leftrightarrow \inf \tilde{n}_\alpha < \inf \tilde{m}_\alpha. \tag{29}$$

Hereby, $\tilde{n}_\alpha$ is the α-cut of $\tilde{n}$ where α is chosen such that:

$$\alpha = \sup\{x | \inf \tilde{n}_x \neq \inf \tilde{m}_x\}.$$

The sup-order of fuzzy integers searches for the highest α, such that the α-cuts have a different supremum and then chooses the fuzzy number for which the α-cut has the higher supremum. It can be seen that this method is equivalent to first searching the fuzzy integers that have the maximal k, say $k_{\max}$, for which $\pi_{\mathbb{N}}(k_{\max}) = 1$. If multiple such fuzzy integers exist, the decision is obtained by applying the leximax-operator on the sequence $\pi_{\mathbb{N}}(k_{\max}+1), \ldots, \pi_{\mathbb{N}}(|M|)$. The dual is true for $\prec_{\text{inf}}$. Note that both $\prec_{\text{sup}}$ and $\prec_{\text{inf}}$ are partial orders. If multiple fuzzy numbers are equivalent, a random choice is made. Note that two non-equal convex fuzzy integers are always comparable by either $\prec_{\text{inf}}$ or $\prec_{\text{sup}}$.

Consider the fuzzy integers shown in Figure 3. The order relation $\prec_{\text{sup}}$ denotes the leftmost fuzzy integer as the largest, because the 1-cut of the leftmost fuzzy integer has a higher supremum (4) than the rightmost (3). However, the order relation $\prec_{\text{inf}}$ denotes the rightmost fuzzy integer as the largest, because the 0.2-cut (denoted by the dashed line) of the leftmost fuzzy number has a lower infimum (2) than the 0.2-cut of the rightmost fuzzy integer (3).

Using the order of fuzzy integers, it is possible to define a merge function ϖ_U, which is driven by an evaluator E_U for atomic universes U. For example, using the order relation $\prec_{\text{sup}}$, the confidence-based merge function ϖ_U for coreferent objects of an atomic universe U has been defined by

$$\varpi_U(M) = \arg\max_{u \in M} \pi^u_{\mathbb{N}} \tag{30}$$

where $\pi^u_{\mathbb{N}}$ is a possibility distribution, representing a fuzzy integer, that is obtained from the multiset $\tilde{P}_u$ of PTVs for which

$$\forall u' \in M : \omega_{\tilde{P}_u}(E_U(u,u')) = \omega_M(u').$$

As such, $\varpi_U(M)$ selects the object $u \in M$ that has the largest corresponding fuzzy number $\pi^u_{\mathbb{N}}$ according to the order relation $\prec_{\text{sup}}$. Selecting the largest fuzzy number hereby reflects that the object with the largest confidence has been chosen as the result of the merging. Illustrations of such merge functions for atomic objects are given in Subsection 6.2.

5.2 Merge Functions for Complex Objects

In order to merge coreferent objects of a complex, composite universum O, a *composite merge function* is used. A possible strategy in doing so is to consider an

evaluator E_O and to construct merge functions for complex universes as explained in the previous subsection.

Another way of defining composite merge functions is to combine the *projection* operator on the compound universe O with merge functions for the atomic universes. Doing so, yields the following definition.

Consider a complex universe $O = U_1 \times \cdots \times U_n$. A *composite merge function* ϖ_O over O is defined by

$$\varpi_O : \mathscr{M}(O) \to O \tag{31}$$

where

$$\varpi_O(M) = (\varpi_{U_1}(\mathrm{Proj}_1(M)), \ldots, \varpi_{U_n}(\mathrm{Proj}_n(M)))$$

with $\mathrm{Proj}_i(M) \in \mathscr{M}(U_i)$ such that

$$\omega_{\mathrm{Proj}_i(M)}(u) = \sum_{o \in M \wedge o_i = u} \omega_M(o).$$

5.3 Merging of Coreferent POIs

The general merge strategies presented in the previous subsections can be used to develop a merge technique for coreferent POIs. Because POIs are complex objects (as specified in Subsection 3.2), a composite merge function as defined by Eq. (31) can be used to merge coreferent POIs.

This approach is motivated by the fact that we prefer to keep only the best (partial) information from each coreferent POI in the resulting merged POI. Hence, we do not prefer to select and preserve one of the existing POIs as the result of the merging operation. With this strategy, we explicitly opt to cleanse the (regular) POI database without introducing uncertain data in it. Indeed, alternatively one might also choose to work with a 'fuzzy' POI database in which uncertainty about the possible values of the POI attributes is explicitly stored. As such the informative richness of the many sources provided by the user communities can be better maintained. However, such an approach will result in POIs that are more difficult to interpret and to process. For that reason this approach is not further considered within the scope of the work presented in this chapter.

In order to specify the composite merge function in accordance with Eq. (31), merge functions for atomic objects, handling locational, descriptive and categorical data must be provided. The use and selection of such merge functions is discussed in the next subsections.

5.3.1 Merging of Locational Data

In POIs, locational data is usually specified by means of a latitude and a longitude value, each of them being modelled by an atomic object, respectively taken from the

atomic universa $[-90,90]$ and $[-180,180]$ (cf. Example 1). Because POIs are often descriptions of geographic areas (buildings, parks, lakes, etc.) latitude and longitude data are often imprecise. A good merge function for latitude and longitude data has to reduce this imprecision as good as possible. Hence, an aggregation function like arithmetic mean could be a good candidate. Two situations are distinguished:

- If we have no information about the scale $1:s$ of the map on which the POIs are entered by the user (or if no map is used to enter POIs), then the latitude value lat (resp. longitude value lon) of the merged POI, resulting from the merging of the coreferent POIs $POI_1,\ldots,POI_n$ is obtained by taking the arithmetic mean of the latitudes lat_i, $i=1,\ldots,n$ (resp. longitudes lon_i) of all coreferent POIs, i.e.,

$$lat = \frac{\sum_{i=1,n} lat_i}{n}, \quad lon = \frac{\sum_{i=1,n} lon_i}{n}. \tag{32}$$

  Alternatively, to eliminate the impact of outliers, the median of the latitudes (resp. longitudes) can be taken as merge result.
- If the POI locations have been inserted by users using maps, then we have scale information and only the latitudes (resp. longitudes) of the POIs at the most detailed scale are considered in the computation of the arithmetic means, i.e.,

$$lat = \frac{\sum_{\substack{i=1,n \\ s_i=s_{min}}} lat_i}{\sum_{\substack{i=1,n \\ s_i=s_{min}}} 1}, \quad lon = \frac{\sum_{\substack{i=1,n \\ s_i=s_{min}}} lon_i}{\sum_{\substack{i=1,n \\ s_i=s_{min}}} 1} \tag{33}$$

  where $1:s_i$ is the scale at which POI_i is entered and $s_{min} = \min\{s_i | i=1,2,\ldots,n\}$.

This approach guarantees that only POIs that are entered at the scale with the highest precision among the scales that are used for the coreferent POIs under consideration are involved in the merge operation.

5.3.2 Merging of Descriptive Data

For descriptive, atomic POI components, a confidence-based merge function, as defined by Eq. (30) can be used. This is motivated by the assumption that the description for which the possible quantity of coreferent descriptions is maximised, is a good candidate for the merge result. Indeed, because for this description we have the highest confidence that it is coreferent with most of the other descriptions.

On the one hand, by selecting only one description from the descriptions of the coreferent POIs, the risk for an inconsistent description in the merged POI is minimised as one could assume that users most likely provide consistent descriptions. However, on the other hand, by neglecting the descriptions of the non-selected POIs, information not present in the description of the selected POI might be lost. A solution for this is to a apply a multi-document summarising technique to the descriptions of all coreferent POIs. Such summarising techniques have been described in [41, 15].

5.3.3 Merging of Categorical Data

For atomic POI components that contain categorical data, a confidence-based merge function, as defined by Eq. (30) can also be used. This is motivated by the assumption that the category for which the possible quantity of coreferent descriptions is maximised, is a good candidate for the merge result.

The underlying assumption at this point is that if different category labels are used in the coreferent POIs, these are most likely the result of user mistakes. Hence, keeping only the label for which the confidence is the highest might be a good merge strategy.

Alternatively, if the category labels are organised in a hierarchical structure, reflecting category-subcategory relationships, then the most common ancestor of the category labels in the coreferent POIs, might be taken as the merge result. In such a case, there is less chance for mistakes, but specific category label information might get lost.

6 An Illustrative Example

To illustrate the corefence detection and merging of POIs as described in the previous sections, the POIs of Example 2 are reconsidered. First we deal with coreference detection in Subsection 6.1, next in Subsection 6.2 the merging is illustrated.

6.1 Illustration of Coreference Detection

As illustrated in Example 1, the POIs under consideration are objects of a complex universe $O_{POI} = U_1 \times U_2 \times U_3 \times U_4 \times U_5 \times U_6$ that consists of six non compound universa $U_1, \ldots, U_6$ of which only the five universa $U_2, \ldots, U_6$ are relevant with respect to corefence detection. Indeed, the univere U_1 is used to model the identifier of a POI which by definition should be unique and which is either provided by the user or generated by the system. Hence it is assumed that the semantics of the identifier do not contribute to the coreference detection process. In the next example, we illustrate POI coreference detection on the basis of the universa $U_2, \ldots, U_6$.

Example 3. Consider the four POIs of Example 2 and assume that all of them have been entered by users using a map interface. POI_1, POI_2 and POI_3 are entered at scale 1 : 10000 which corresponds to a street map of Ghent, whereas POI_4 is entered at scale 1 : 1000000 which corresponds to a map of Belgium. The latitude, longitude, scale, radius of screen precision, and parameter value for ε of these POIs (cf. Subsection 4.2.2) are summarised in Table 1. The minimum scale supported is assumed to be 1 : 10000. For all POIs, the same precision $d_s = 0.01m$ is used. This precision is assumed to be provided by the user (or could alternatively be set by default in the system).

Table 1 Information about the POIs used in Example 3.

POI	lat	lon	$1:s$	d_s	$\varepsilon = d_p s_{min}$
POI_1	51.056934	3.727112	1:10000	0.01m	2.8m
POI_2	51.053036	3.727015	1:10000	0.01m	2.8m
POI_3	51.053177	3.726382	1:10000	0.01m	2.8m
POI_4	51.033333	3.700000	1:1000000	0.01m	2.8m

We now present the calculation of the uncertainty of coreference for objects of each of the constituting relevant universa $U_2, \ldots, U_6$.

- **Coreference detection for objects of the universum U_2.** This universum is used to model the name of the POI. As explained before, the uncertainty of coreference for names is preferably determined by means of a syntactical evaluator E_{name}. By using the evaluators described in [10, 12], the PTVs $(\mu_{\tilde{p}}(T), \mu_{\tilde{p}}(F))$ in Table 2 are obtained. These PTVs express the uncertainty about the coreference of the names of the POIs under consideration, i.e., POI_1, POI_2, POI_3 and POI_4. From these results it can be seen that the names of POI_2 and POI_4 are certainly coreferent because reflexivity of the evaluator requires that equal object value are certain to be coreferent. In addition, other POI names are certainly not coreferent, due to a lack of sufficient syntactical similarities between names.

Table 2 Uncertainty about the coreference of the names of POI_x and POI_y.

POI_x	POI_y	$E_{name}(POI_x, POI_y)$
POI_1	POI_2	(0,1)
POI_1	POI_3	(0,1)
POI_1	POI_4	(0,1)
POI_2	POI_3	(0,1)
POI_2	POI_4	(1,0)
POI_3	POI_4	(0,1)

- **Coreference detection for objects of the universa U_3 and U_4.** Universa U_3 and U_4 are respectively used to model the latitude and longitude of a POI. In order to apply the techniques presented in Subsection 4.2 both the latitude and longitude of a POI have to be considered together. An evaluator $e_{location}$, which uses Eq. 19, is applied to compute the PTV that reflects the (un)certainty about the co-location of two POIs. Table 3 gives an overview of the results obtained from the application of the evaluator $E_{location}$ for the POIs under consideration. The third column gives the distances between the POIs as computed by using Eq. (13). The fourth column contains the values for the parameter δ as computed by using Eq. (18). Whereas the last column represents the resulting PTVs $(\mu_{\tilde{p}}(T), \mu_{\tilde{p}}(F))$ denot-

ing the (un)certainty about the co-location of the POIs as obtained by applying Eq. 19.

Table 3 Uncertainty about the co-location of POI_x and POI_y.

POI_x	POI_y	$d(POI_x, POI_y)$	$\delta = \varepsilon + \max(s_1 d_{s_1}, s_2 d_{s_2})$	$E_{location}(POI_x, POI_y)$
POI_1	POI_2	433.2m	102.8m	(0,1)
POI_1	POI_3	420.6m	102.8m	(0,1)
POI_1	POI_4	3235.2m	10002.8m	(1,0.48)
POI_2	POI_3	46.9m	102.8m	(1,0.79)
POI_2	POI_4	2890.8m	10002.8m	(1,0.41)
POI_3	POI_4	2874.1m	10002.8m	(1,0.40)

These results reflect that POI_1 is not co-located with POI_2 and POI_3, which is reflected by the PTV $(0,1)$. Remind that it has been assumed in the example that POI_4 is entered at scale 1 : 1000000, which is less precise than scale 1 : 10000. This makes that there is no certainty about the co-location of POI_4 with POI_1, POI_2 and POI_3 what is respectively reflected in the PTVs $(1,0.48)$, $(1,0.41)$ and $(1,0.40)$. Due to their possibilistic interpretation each of these PTVs expresses that it is either completely possible that there is co-location ($\mu_{\tilde{p}}(T) = 1$) or that it is either to a lower extent possible that there is no co-location ($\mu_{\tilde{p}}(F)$ resp. being equal to 0.48, 0.41 and 0.40). Likewise, the PTV $(1,0.79)$ expresses that it is either completely possible ($\mu_{\tilde{p}}(T) = 1$) that POI_2 and POI_3 are co-located, or that it is either possible to a lower extent $\mu_{\tilde{p}}(F) = 0.79$ that these are not co-located. This rather high value of 0.79 is due to the pessimistic assumption of ε being only 2.8m, where Saint-Bavo cathedral has a diagonal of about 110m. Alternatively, using Eq. (17), we obtain that $\varepsilon = 55m$ and applying Eq. (18) yields $\delta = 155m$. So, using this alternative approach, the resulting PTV becomes $\{(T,1)\}$, what corresponds to true and illustrates the efficiency of Eq. (17).

- **Coreference detection for objects of the universum U_5.** Universum U_5 is used to model the description of the POI. Similarly as for the name, the coreference detection for the description of a POI is preferably done using a syntactical evaluator E_{descr}. The PTVs in Table 4 show the uncertainty about coreference of the descriptions of the POIs under consideration and are obtained by applying the evaluators that have been described in [10, 12]. From these results it follows that the POI description of POI_1 is certainly not coreferent with that of POI_3 (PTV $(0,1)$). There is also higher confidence that this description is not coreferent with that of POI_2 ($\mu_{\tilde{p}}(F) = 1$ in PTV $(0.5,1)$) and POI_4 ($\mu_{\tilde{p}}(F) = 1$ in PTV $(0.3,1)$) than there is confidence that the description of POI_1 is coreferent with the description of POI_2 ($\mu_{\tilde{p}}(T) = 0.5$ in PTV $(0.5,1)$) and the description of POI_4 ($\mu_{\tilde{p}}(T) = 0.3$ in PTV $(0.3,1)$). Furthermore, there is higher confidence that the descriptions of POI_2, POI_3 and POI_4 are coreferent (PTVs $(1,0.1)$) than there is

Table 4 Uncertainty about the coreference of the descriptions of POI_x and POI_y.

POI_x	POI_y	$E_{descr}(POI_x, POI_y)$
POI_1	POI_2	(0.5,1)
POI_1	POI_3	(0,1)
POI_1	POI_4	(0.3,1)
POI_2	POI_3	(1,0.1)
POI_2	POI_4	(1,0.1)
POI_3	POI_4	(1,0.1)

confidence that these descriptions are not coreferent.

- **Coreference detection for objects of the universum U_6.** Universum U_6 is used to model the category class of the POI. As opposed to the POI name and description, the type of the POIs is compared in a semantical manner. Therefore, a binary relation R between POI types is constructed dynamically as described in [11]. Then, based on this binary relation, uncertainty about category values can be inferred using the semantic evaluator $E_{category}$. Table 5 presents the results of these computations.

Table 5 Uncertainty about the coreference of the category values of POI_x and POI_y.

POI_x	POI_y	$E_{category}(POI_x, POI_y)$
POI_1	POI_2	(0,1)
POI_1	POI_3	(0,1)
POI_1	POI_4	(0,1)
POI_2	POI_3	(1,0.5)
POI_2	POI_4	(1,0.5)
POI_3	POI_4	(1,0)

As can be seen, because of the PTV $(0,1)$ the category value of POI_1 ('Market') is not coreferent with the category values of POI_2 ('Church'), POI_3 ('Cathedral') and POI_4 ('Cathedral'). The category values of POI_3 and POI_4 are the same ('Cathedral') and therefore coreferent, what is reflected by the PTV $(1,0)$. Furthermore, the category value of POI_3 and POI_4 ('Cathedral') is connected through an 'is-a' relation with the category value of POI_2 ('Church'). This connection is reflected in the binary relation R (not shown in the chapter) and resulted in a PTV $(1,0.5)$ describing that there is higher confidence that the value 'Church' is related to the value 'Cathedral' ($\mu_{\tilde{p}}(T) = 1$ in PTV $(1,0.5)$) than there is confidence that both values are not coreferent ($\mu_{\tilde{p}}(F) = 0.5$ in PTV $(1,0.5)$).

Finally, given the above uncertainties about the coreference for all the objects of the universa $U_2, \ldots, U_6$ (i.e., marginal possibilities), the uncertainty about the coref-

erence of POIs can be calculated using a complex evaluator E_{POI}. For that purpose, an aggregation technique based on the Sugeno integral is used. As has been proposed in [8], such an approach requires two necessity measures γ^T and γ^F. These fuzzy measure γ^T (resp. γ^F) evaluates subsets of POI attributes and expresses the necessity that coreference of the values of the attributes in the set implies coreference (resp. does not imply coreference) of the POIs containing those values. The necessity measures used in this example are given as shown in Table 6. The given measures reflect that marginal knowledge about less than three attributes is considered to provide us with no necessity at all about the coreference of the POIs. However, marginal knowledge of three or more attributes allows us to infer necessity about (non) coreference. Note that the fuzzy measures satisfy the normalisation constraint:

$$\forall L \in \mathscr{L} : \min\left(\gamma^T(L), \gamma^F\left(\overline{L}\right)\right) = 0. \tag{34}$$

Table 6 The given necessity measures γ^T and γ^F used in the Sugeno integral in order to reflect how conditional knowledge about the values of POI attribute subsets leads to knowledge about the coreferece of the POIs.

$L \subseteq \mathscr{L}$	$\gamma^T(L)$	$\gamma^F(L)$
$\emptyset$	0	0
{name}	0	0
{location}	0	0
{description}	0	0
{type}	0	0
{name, location}	0	0
{name, description}	0	0
{name, type}	0	0
{location, description}	0	0
{location, type}	0	0
{description, type}	0	0
{name, location, description}	0.9	1
{name, location, type}	0.6	1
{name, description, type}	0	1
{location, description, type}	0.8	1
{name, location, description, type}	1	1

Combining the conditional necessity as given in Table 6 with the marginal necessities that can be derived from the marginal PTVs from Tables 2, 3, 4 and 5 is then done using the Sugeno integral for PTVs, which is defined by Eq. (22). Applying the Sugeno integral with the PTVs from Tables 2, 3, 4 and 5 leads to the aggregated PTVs shown in Table 7. More details about (the use of) the Sugeno integral are given in [8]. □

Table 7 Overall uncertainty about the coreference of POI_x and POI_y.

POI_x	POI_y	$E_{POI}(POI_x, POI_y)$
POI_1	POI_2	(0,1)
POI_1	POI_3	(0,1)
POI_1	POI_4	(0.3,1)
POI_2	POI_3	(1,0.79)
POI_2	POI_4	(1,0.41)
POI_3	POI_4	(1,0.40)

6.2 Illustration of Merging

Reconsider the POIs of Example 2. Based on the coreference detection results presented in Table 7, we can safely conclude that POI_2, POI_3 and POI_4 are coreferent (to some extent). In the next example we illustrate the merging of these three coreferent POIs.

Example 4. By applying the techniques described in Section 5, the merging of coreferent POIs is done in two steps. In the first step, merge functions for the relevant universa $U_2, \ldots, U_6$ are specified and applied.

- **Merging of objects of the universum U_2.** Objects of the universe U_2 represent POI names. The names of the coreferent POIs POI_2, POI_3 and POI_4 are respectively, 'St-Bavo', 'Ghent cathedral' and 'St-Bavo'. Reconsider the PTVs obtained from the coreference detection of POI names given in Table 2. By applying Eq. (30), it is obtained that the name with the largest possible quantity of coreferent names is 'St-Bavo'.
  Indeed, for each coreferent POI POI_i, the corresponding fuzzy number $\pi_{\mathbb{N}}^{POI_i}$ is obtained as follows:
  - For POI_2:

$$E_{name}(POI_2, POI_2) = (1,0)$$
$$E_{name}(POI_2, POI_3) = (0,1)$$
$$E_{name}(POI_2, POI_4) = (1,0).$$

    This allows us to construct the multiset $\tilde{P} = \{(1,0),(1,0),(0,1),(1,0)\}$ where the first POI $(1,0)$ is added to obtain a correct modelling for $\pi_{\mathbb{N}}^{POI_i}(0)$. Applying Eq. (8) yields the ordered list of POIs

$$[(1,0),(1,0),(1,0),(0,1)].$$

    Applying Eq. (27) then yields

$$\pi_{\mathbb{N}}^{POI_2}(0) = 0$$
$$\pi_{\mathbb{N}}^{POI_2}(1) = \min(1,0) = 0$$
$$\pi_{\mathbb{N}}^{POI_2}(2) = \min(1,1) = 1$$
$$\pi_{\mathbb{N}}^{POI_2}(3) = 0.$$

– For POI_3:

$$E_{name}(POI_3, POI_3) = (1,0)$$
$$E_{name}(POI_3, POI_2) = (0,1)$$
$$E_{name}(POI_3, POI_4) = (0,1).$$

This allows us to construct the extended multiset $\tilde{P} = \{(1,0),(1,0),(0,1),(0,1)\}$ and the ordered list of POIs

$$[(1,0),(1,0),(0,1),(0,1)].$$

Applying Eq. (27) then yields

$$\pi_{\mathbb{N}}^{POI_3}(0) = 0$$
$$\pi_{\mathbb{N}}^{POI_3}(1) = \min(1,1) = 1$$
$$\pi_{\mathbb{N}}^{POI_3}(2) = \min(0,1) = 0$$
$$\pi_{\mathbb{N}}^{POI_3}(3) = 0.$$

– For POI_4:

$$E_{name}(POI_4, POI_4) = (1,0)$$
$$E_{name}(POI_4, POI_2) = (1,0)$$
$$E_{name}(POI_4, POI_3) = (0,1).$$

This yields the extended multiset $\tilde{P} = \{(1,0),(1,0),(1,0),(0,1)\}$ and the ordered list of POIs

$$[(1,0),(1,0),(1,0),(0,1)].$$

Applying Eq. (27) then yields

$$\pi_{\mathbb{N}}^{POI_4}(0) = 0$$
$$\pi_{\mathbb{N}}^{POI_4}(1) = \min(1,0) = 0$$
$$\pi_{\mathbb{N}}^{POI_4}(2) = \min(1,1) = 1$$
$$\pi_{\mathbb{N}}^{POI_4}(3) = 0.$$

Applying Eq. (28) results in

$$\pi_{\mathbb{N}}^{POI_3} \prec_{\text{sup}} \pi_{\mathbb{N}}^{POI_2} \text{ and } \pi_{\mathbb{N}}^{POI_3} \prec_{\text{sup}} \pi_{\mathbb{N}}^{POI_4}$$

Such that using Eq. (30) returns

$$\varpi_{name}(\{POI_2, POI_3, POI_4\}) = \arg \max_{u \in \{POI_2, POI_3, POI_4\}} \pi_{\mathbb{N}}^{u} = POI_2 \text{ or } POI_4.$$

Hence the name with the largest possible quantity of coreferent names is the name of POI_2 or POI_4, which is in both cases 'St-Bavo'. So, the merged value for the POI name is 'St-Bavo'. As a side effect of this merge technique the (less specific) information 'Ghent cathedral' is lost.

- **Merging of objects of the universa U_3 and U_4.** Universa U_3 and U_4 together model the location of a POI. These universa were handled together in the coreference detection process. Recall from Table 1 that POI_2 and POI_3 have been entered at scale $1:10000$, whereas POI_4 has been entered at a less detailed map scale $1:1000000$. Because we have scale information and not all coreferent POIs have been entered at the same scale, Eq. (33) can be used to compute the latitude and longitude value of the merged POI. Hereby, only the information related to the most detailed scale, i.e., the data from POIs POI_2 and POI_3, are considered. Using the data given in Table 1, this yields

$$lat = \frac{51.053036 + 51.053177}{2} = 51.053106$$

and

$$lon = \frac{3.727015 + 3.726382}{2} = 3.726699.$$

The differences between the latitude and longitude of POI_2 and POI_3 are inherent to the fact that both POIs are representing (the geographical area of) St.-Bavo cathedral, which has a diagonal of about 110m, at a scale with a precision of 0.01m.

- **Merging of objects of the universum U_5.** Objects of the universe U_5 represent POI descriptions. The descriptions of the coreferent POIs POI_2, POI_3 and POI_4 are respectively, 'St-Bavo's Cathedral, Ghent', 'St-Bavo Cathedral' and 'St-Bavo – Ghent'. The same technique as previously used for POI names can be applied. Hence, Eq. (30) can now be applied with the PTVs obtained from the coreference detection of POI descriptions given in Table 4.
  For each coreferent POI POI_i, the corresponding fuzzy number $\pi_{\mathbb{N}}^{POI_i}$ is obtained as follows:
  - For POI_2:

$$E_{descr}(POI_2, POI_2) = (1, 0)$$
$$E_{descr}(POI_2, POI_3) = (1, 0.1)$$
$$E_{descr}(POI_2, POI_4) = (1, 0.1).$$

This allows us to construct the extended multiset $\tilde{P} = \{(1,0),(1,0),(1,0.1),(1,0.1)\}$ and the ordered list of POIs

$$[(1,0),(1,0),(1,0.1),(1,0.1)].$$

Applying Eq. (27) then yields

$$\begin{aligned}
&\pi_{\mathbb{N}}^{POI_2}(0) = 0\\
&\pi_{\mathbb{N}}^{POI_2}(1) = \min(1,0.1) = 0.1\\
&\pi_{\mathbb{N}}^{POI_2}(2) = \min(1,0.1) = 0.1\\
&\pi_{\mathbb{N}}^{POI_2}(3) = 1.
\end{aligned}$$

– For POI_3:

$$\begin{aligned}
&E_{descr}(POI_3, POI_3) = (1,0)\\
&E_{descr}(POI_3, POI_2) = (1,0.1)\\
&E_{descr}(POI_3, POI_4) = (1,0.1).
\end{aligned}$$

This yields the extended multiset $\tilde{P} = \{(1,0),(1,0),(1,0.1),(1,0.1)\}$ and the ordered list of POIs

$$[(1,0),(1,0),(1,0.1),(1,0.1)].$$

Applying Eq. (27) then yields

$$\begin{aligned}
&\pi_{\mathbb{N}}^{POI_3}(0) = 0\\
&\pi_{\mathbb{N}}^{POI_3}(1) = \min(1,0.1) = 0.1\\
&\pi_{\mathbb{N}}^{POI_3}(2) = \min(1,0.1) = 0.1\\
&\pi_{\mathbb{N}}^{POI_3}(3) = 1.
\end{aligned}$$

– For POI_4:

$$\begin{aligned}
&E_{descr}(POI_4, POI_4) = (1,0)\\
&E_{descr}(POI_4, POI_2) = (1,0.1)\\
&E_{descr}(POI_4, POI_3) = (1,0.1).
\end{aligned}$$

This yields the extended multiset $\tilde{P} = \{(1,0),(1,0),(1,0.1),(1,0.1)\}$ and the ordered list of POIs

$$[(1,0),(1,0),(1,0.1),(1,0.1)].$$

Applying Eq. (27) then yields

$$\pi_{\mathbb{N}}^{POI_4}(0) = 0$$
$$\pi_{\mathbb{N}}^{POI_4}(1) = \min(1, 0.1) = 0.1$$
$$\pi_{\mathbb{N}}^{POI_4}(2) = \min(1, 0.1) = 0.1$$
$$\pi_{\mathbb{N}}^{POI_4}(3) = 1.$$

Using Eq. (30) returns

$$\varpi_{descr}(\{POI_2, POI_3, POI_4\}) = \arg \max_{u \in \{POI_2, POI_3, POI_4\}} \pi_{\mathbb{N}}^{u} = POI_2 \text{ or } POI_3 \text{ or } POI_4.$$

Hence, all three descriptions qualify as the description with the largest possible quantity of coreferent descriptions. A choice has to be made. Considering the fact that we want to minimise information loss, the description which consists of most characters will be chosen in such a case. So, the merged value for description becomes 'St-Bavo's Cathedral, Ghent'.

- **Merging of objects of the universum U_6.** Objects of the universe U_6 represent the categorical data about the POI. For categorical data, the same confidence-based merge technique as used before is applied. The POI categories in the coreferent POIs POI_2, POI_3 and POI_4 are respectively, 'Church', 'Cathedral' and 'Cathedral'. Using the PTVs obtained from the coreference detection of POI (category) types given in Table 5 yields that the type with the largest possible quantity of coreferent types is 'Cathedral'. This follows from the following computations.
  For each coreferent POI POI_i, the corresponding fuzzy number $\pi_{\mathbb{N}}^{POI_i}$ is obtained as follows:
  - For POI_2:

    $$E_{category}(POI_2, POI_2) = (1, 0)$$
    $$E_{category}(POI_2, POI_3) = (1, 0.5)$$
    $$E_{category}(POI_2, POI_4) = (1, 0.5).$$

    This allows us to construct the extended multiset $\tilde{P} = \{(1,0), (1,0), (1,0.5), (1,0.5)\}$ and the ordered list of POIs

    $$[(1,0), (1,0), (1,0.5), (1,0.5)].$$

    Applying Eq. (27) then yields

    $$\pi_{\mathbb{N}}^{POI_2}(0) = 0$$
    $$\pi_{\mathbb{N}}^{POI_2}(1) = \min(1, 0.5) = 0.5$$
    $$\pi_{\mathbb{N}}^{POI_2}(2) = \min(1, 0.5) = 0.5$$
    $$\pi_{\mathbb{N}}^{POI_2}(3) = 1.$$

– For POI_3:

$$E_{category}(POI_3, POI_3) = (1,0)$$
$$E_{category}(POI_3, POI_2) = (1,0.5)$$
$$E_{category}(POI_3, POI_4) = (1,0).$$

This allows us to construct the extended multiset $\tilde{P} = \{(1,0),(1,0),(1,0.5),(1,0)\}$ and the ordered list of POIs

$$[(1,0),(1,0),(1,0),(1,0.5)].$$

Applying Eq. (27) then yields

$$\pi_{\mathbb{N}}^{POI_3}(0) = 0$$
$$\pi_{\mathbb{N}}^{POI_3}(1) = \min(1,0) = 0$$
$$\pi_{\mathbb{N}}^{POI_3}(2) = \min(1,0.5) = 0.5$$
$$\pi_{\mathbb{N}}^{POI_3}(3) = 1.$$

– For POI_4:

$$E_{category}(POI_4, POI_4) = (1,0)$$
$$E_{category}(POI_4, POI_2) = (1,0.5)$$
$$E_{category}(POI_4, POI_3) = (1,0).$$

This allows us to construct the extended multiset $\tilde{P} = \{(1,0),(1,0),(1,0.5),(1,0)\}$ and the ordered list of POIs

$$[(1,0),(1,0),(1,0),(1,0.5)].$$

Applying Eq. (27) then yields

$$\pi_{\mathbb{N}}^{POI_4}(0) = 0$$
$$\pi_{\mathbb{N}}^{POI_4}(1) = \min(1,0) = 0$$
$$\pi_{\mathbb{N}}^{POI_4}(2) = \min(1,0.5) = 0.5$$
$$\pi_{\mathbb{N}}^{POI_4}(3) = 1.$$

Such that using Eq. (30) returns

$$\varpi_{category}(\{POI_2, POI_3, POI_4\}) = \arg \max_{u \in \{POI_2, POI_3, POI_4\}} \pi_{\mathbb{N}}^{u} = POI_3 \text{ or } POI_4.$$

Thus, the type with the largest possible quantity of coreferent types is the type of POI_3 or POI_4, which is in both cases ‘Cathedral’. Hence, the incorrect category value ‘Church’ is neglected by the merge strategy.

In the second step, the results of the previous merge operations are combined using the composite merge function given by Eq. (31). The resulting merged POI then finally becomes

$$(POI_{merge}, \text{'St-Bavo'}, 51.053106, 3.726699, \text{'St-Bavo's Cathedral, Ghent'}, \text{'Cathedral'}).$$

This POI gives a consistent description of St-Bavo's cathedral. □

The case study presented above is limited, though chosen for exemplifying the presented coreference detection and merging mechanisms. Other, more specific and statistically relevant tests, covering more extended data sets, have been performed and published in [14]. These tests proof the efficiency of the presented methods in terms of precision and recall as compared to the other methods presented in the literature.

7 Conclusions and Further Work

7.1 Contribution

In this chapter, a novel soft computing approach to cleanse POI databases is described. In essence, this approach consists of two parts. In the first part, the uncertainty about the potential coreference of two POIs is estimated and subsets of potentially coreferent POIs are identified (two POIs are considered to be coreferent if they describe the same geographical location or object at a geographical location). In the second part, coreferent POIs are merged into a new POI which acts as a representation of all information present in the coreferent POIs.

At the basis of the approach is the concept of evaluators for coreference detection. Such an evaluator takes two objects as input and returns an estimation of the (un)certainty that these objects are coreferent, expressed by means of a possibilistic truth value (PTV). Evaluators have been proposed for atomic objects, co-location detection and complex objects.

The specific evaluators for co-location detection are especially suited for cases where latitude and longitude coordinates of POIs are entered by users using a map interface, which is often the case with POI databases that are maintained by a user community. The evaluators allow to explicitly cope with the context (scale and precision) with which the locational data have been entered. Fuzzy ranges are used to determine in a flexible way whether two POI locations can be considered to be close enough to conclude that they are co-located.

Coreferent POIs are merged using merge functions. A merge function takes a finite number of objects as input and returns a (new) object that acts as a representation of the input objects. Merge functions have been proposed for atomic objects and complex objects. The presented merge functions for atomic objects are based on

an evaluator. Complex objects are merged using a composite merge function. Typical for composite merge functions is that they do not preserve any of the coreferent POIs, but combine the best (most confident) parts of each of them to construct a novel, merged POI.

7.2 Context

The presented work contributes to research on data quality issues in information retrieval. On the one hand it offers automatic data cleansing techniques which could be developed further and generalised in order to improve data quality in information sources. Information retrieval processes could benefit from an improved data quality and provide better results as the data quality will be propagated in data processing results.

On the other hand such data cleansing techniques can also be applied to cleanse the results of information retrieval operations that run on unclean data. Coreference detection techniques can be used to detect coreferent results, which in their turn eventually can be merged using merging techniques.

Moreover, the computed uncertainty measures obtained from the coreference detection can be communicated to the users as an indication of the quality of the retrieval results.

The presented approach is based on soft computing techniques and allows to reflect human reasoning with respect to coreference detection and object merging in an adequate way. This leads to more justifiable results as compared to those obtained by using existing approaches. This is the main advantage of the proposed approach. Statistically relevant experiments on different data sets reported in [14] reveal that the proposed techniques for coreference detection overall perform better in terms of precision and recall than the related techniques that were mentioned in Section 2. More extended tests to validate the performance of the proposed merge techniques are required and are currently under development. Note that such tests are more difficult to implement as the ground truth for object merging is much more difficult to obtain.

7.3 Further Work

Further research is required and planned. The techniques presented in this chapter have been specifically developed for the cleansing of POI databases. An important aspect that will be further investigated is the generalisation of the approach so that it will become applicable for the cleansing of other, more general databases. For that purpose, among others, it is worth investigating whether other aggregation techniques like, e.g., the technique used in logic scoring of preference (LSP) which is based on the generalized conjunction/disjunction (GCD) function [23], of-

fer better aggregation facilities for coreference detection than the approach based on the Sugeno integral. Furthermore, the desired mathematical properties of merge functions should be better understood and new families of merging functions able to model different kinds of desired behavior should be developed. For example, in some cases it might be preferable to keep as much information as possible in the resulting merged object. In such cases, rather than selecting the most confident part, the merging function should concatenate, summarise or combine all available data in an intelligent way.

Another aspect to investigate further concerns the optimization of the object comparison technique. Optimization is possible as not all pairs in a set of objects must necessarily be checked to detect all coreferent objects. Moreover, not all components of a complex object must necessarily in all cases be evaluated to come to a conclusion regarding coreference.

References

1. Baral C., Kraus S., Minker J. (1991) *Combining multiple knowledge bases*. IEEE Transactions on Knowledge and Data Engineering **3**(2):208–220.
2. Baral C., Kraus S., Minker J., Subrahmanian V. (1992) *Combining knowledge bases consisting of first-order theories*. Computational Intelligence **8**(1):45–71.
3. Bloch I. (1996) *Information Combination Operators for Data Fusion: A Comparative Review with Classification*. IEEE Transactions on Systems, Man and Cybernetics, part A, **26**(1):52–67.
4. Borgida A., Imielinski T. (1984) *Decision making in committees – a framework for dealing with inconsistency and non-monotonicity*. In: Proc. of the Workshop of Nonmonotonic reasoning, pp. 21–32.
5. Bordogna G., Pagani M., Pasi G. (2010) *Imperfect Multisource Spatial Data Fusion Based on a Local Consensual Dynamics*. In: Kacprzyk J., Petry F.E., Yazici A. (eds.) Uncertainty Approaches for Spatial Data Modeling and Processing: A Decision Support Perspective. Studies in Computational Intelligence **271**, Springer-Verlag, Heidelberg, Germany, pp. 79–94.
6. Boury-Brisset A.-C. (2003) *Ontology-based approach for information fusion*. In: Proc. of the 6th International Conference on Information Fusion, pp. 522–529.
7. Bright M., Hurson A., Pakzad S. (1992) *A taxonomy and current issues in multidatabase systems*. Computer **25**(3):50–59.
8. Bronselaer A., Hallez A., De Tré G. (2009) *Extensions of Fuzzy Measures and Sugeno Integral for Possibilistic Truth Values*. International Journal of Intelligent Systems **24**(2):97–117.
9. Bronselaer A., Hallez A., De Tré G. (2009) *A possibilistic view on set and multiset comparison*. Control and Cybernetics **38**(2):341–366.
10. Bronselaer A., De Tré G. (2009) *A possibilistic approach to string comparison*. IEEE Transactions on Fuzzy systems **17**(1):208–223.
11. Bronselaer A., De Tré G. (2009) *Semantical evaluators*. In: Proc. of the IFSA 2009 International Conference, Lisbon, Portugal, pp. 663–668.
12. Bronselaer A., De Tré G. (2010) *Properties of possibilistic string comparison*. IEEE Transactions on Fuzzy systems **18**(2):312–325.
13. Bronselaer A., De Tré G. (2010) *Aspects of object merging*. In: Proc. of the NAFIPS 2010 International Conference, Toronto, Canada, pp. 27–32.
14. Bronselaer A. (2010) *Coreferency of atomic and complex objects* (in Dutch). PhD thesis, Ghent University, December 2010, Ghent, Belgium.

15. Bronselaer A., Van Britsom D., De Tré G. (2012) *A framework for multiset merging*. Fuzzy Sets and Systems **191**:1–20.
16. Carrara P., Bordogna G., Boschetti M., Brivio P.A., Nelson A., Stroppiana D. (2008) *A flexible multi-source spatial-data fusion system for environmental status assessment at continental scale*. International Journal of Geographical Information Science **22**(7):781-799.
17. De Cooman G. (1993) *Evaluatieverzamelingen en - afbeeldingen. Een ordetheoretische benadering van vaagheid en onzekerheid* (in Dutch). Ph.D. dissertation, Ghent University, Belgium.
18. De Cooman G. (1995) *Towards a possibilistic logic*. In: Ruan D. (ed.) Fuzzy set theory and advanced mathematical applications, Kluwer Academic, Boston, USA, pp. 89–133.
19. Destercke S., Dubois D., Chojnacki E. (2009) *Possibilistic information fusion using maximal coherent subsets*. IEEE Transactions on Fuzzy Systems **17**(1):79–92.
20. De Tré G., Bronselaer A., Matthé T., Van de Weghe N., De Maeyer P. (2010) *Consistently Handling Geographical User Data: Context-Dependent Detection of Co-located POIs*. In: Proc. of the Information Processing and Management of Uncertainty in Knowledge-Based Systems conference (IPMU2010), Communications in Computer and Information Science **81**:85–94.
21. Dubois D., Prade H. (1988) *Possibility Theory*. Plenum Press, New York, USA.
22. Dubois D., Prade H. (eds.) (2000) *Fundamentals of Fuzzy Sets*. Kluwer Academic Publishers, Boston, USA.
23. Dujmović J., Larsen H.L. (2007) *Generalized conjunction/disjunction*. International Journal of Approximate Reasoning **46**(3):423–446.
24. Ester M., Kriegel H.P., Sander J., Xu X. (1996) *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*. In: Proc. of the 2nd International Conference on Knowledge Discovery and Data Mining, AAAI Press, pp. 226-231.
25. Fellegi I., Sunter, A. (1969) *A Theory for Record Linkage*. American Statistical Association Journal **64**(328):1183–1210.
26. Federal Geographic Data Committee (1998) *Content standard for digital geospatial metadata*. FGDC-STD-001-1998, Washington D.C., USA.
27. Foley H., Petry F. (2000) *Fuzzy Knowledge-Based System for Performing Conflation in Geographical Information Systems*. In: Proc. of the IEA/AIE 2000 Conference, Lecture Notes in Artificial Intelligence **1201**:260–271.
28. Hallez A., De Tré G., Verstraete J., Matthé T. (2004) *Application of fuzzy quantifiers on possibilistic truth values*. In: Proc. of the Eurofuse Workshop on Data and Knowledge Engineering, Warsaw, Poland, pp. 252–254.
29. Jaro M. (1976) *Unimatch: A record linkage system: Users manual*. US Bureau of the Census, Technical Report.
30. Konieczny S., Pérez R. (2002) *Merging information under constraints: a logical framework*. Journal of Logic and Computation **12**(1):111–120.
31. Lin J., Mendelzon A. (1994) *Knowledge base merging by majority*. In: Pareshi R., Fronhöfer B. (eds.) Dynamic Worlds: From the Frame Problem to Knowledge Management, Kluwer Academic, Boston, USA, pp. 195–218.
32. Lin J., Mendelzon A. (1998) *Merging databases under constraints*. International Journal of Cooperative Information Systems **7**(1):55–76.
33. Nachouki G., Ouafafou M. (2008) *Multi-data source fusion*. Information Fusion **9**(4):523–537.
34. National Geospatial Intelligence Agency (NGA) (2004) *Geodetic System 1984: Its Definitions and Relationships with Local Geodetic Systems*. NIMA Technical Report 8350.2.
35. Prade G. (1982) *Possibility sets, fuzzy sets and their relation to Lukasiewicz logic*. In: Proc. of the International Symposium on Multiple-Valued Logic, pp. 223–227.
36. Rahimi S., Cobb M., Ali D., Paprzycki M., Petry F.E. (2002) *A Knowledge-Based Multi-Agent System for Geospatial Data Conflation*. Journal of Geographic Information and Decision Analysis **6**(2):67–81.

37. Rodríguez M.A., Egenhofer M.J. (2004) *Comparing Geospatial Entity Classes: An Asymmetric and Context-Dependent Similarity Measure*. International Journal of Geographical Information Science **18**:229–256.
38. Sandri S., Dubois D., Kalfsbeek H. (1995) *Elicitation, assessment and pooling of expert judgements using possibility theory*. IEEE Transactions on Fuzzy Systems **3**(3):313–335.
39. Sinnott R.W. (1984) *Virtues of the Haversine*. Sky and Telescope **68**(2):159.
40. Torres R., Keller G.R., Kreinovich V., Longpré L., Starks S.A. (2004) *Eliminating Duplicates under Interval and Fuzzy Uncertainty: An Asymptotically Optimal Algorithm and Its Geospatial Applications*. Reliable Computing **10**(5):401–422.
41. Van Britsom D., Bronselaer A., De Tré G. (2011) *Automatically Generating Multi-Document Summarizations*. In: Proc. of the 11th International Conference on Intelligent Systems Design and Applications, Cordoba, Spain, pp. 142–147.
42. Van Schooten A. (1988) *Ontwerp en implementatie van een model voor de representatie en manipulatie van onzekerheid en imprecisie in databanken en expert systemen* (in Dutch). Ph.D. dissertation, Ghent University, Belgium.
43. Winkler W.E. (1999) *The State of Record Linkage and Current Research Problems*. R99/04, Statistics of Income Division, U.S. Census Bureau.
44. Yager R. (1986) *On the theory of bags*. International Journal of General Systems **13**(1):23–27.
45. Yager R. (1988) *On ordered weighted averaging aggregation operators in multicriteria decision making*. IEEE Transactions on Systems, Man and Cybernetics **18**(1):183–190.
46. Yager R., Rybalov A. (1996) *Uninorm aggregation operators*. Fuzzy Sets and Systems **80**(1):111–120.
47. Zadeh L.A. (1965) *Fuzzy Sets*. Information and Control **8**(3):338–353.
48. Zadeh L.A. (1978) *Fuzzy sets as a basis for a theory of possibility*. Fuzzy Sets and Systems **1**:3–28.