Fudge: Fuzzy Ontology Building with Consensuated Fuzzy Datatypes

Ignacio Huitzil[1], Fernando Bobillo[1,2], Juan Gómez-Romero[3], and Umberto Straccia[4]

[1]University of Zaragoza, Zaragoza, Spain
[2]Aragon Institute of Engineering Research (I3A), Zaragoza, Spain
[3]University of Granada, Granada, Spain
[4]ISTI-CNR, Pisa, Italy

Abstract

An important problem in Fuzzy OWL 2 ontology building is the definition of fuzzy membership functions for real-valued fuzzy sets (so-called fuzzy datatypes in Fuzzy OWL 2 terminology). In this paper, we present a tool, called *Fudge*, whose aim is to support the consensual creation of fuzzy datatypes by aggregating the specifications given by a group of experts. Fudge is open-source and currently supports several linguistic aggregation strategies, including the convex combination, linguistic OWA, weighted mean and fuzzy OWA, and easily allows to build others in. We also propose and have implemented two novel linguistic aggregation operators, based on a right recursive form of the convex combination and of the linguistic OWA.

1 Introduction

OWL 2 ontologies are nowadays widely used to represent and share knowledge representation. In summary, an ontology is a formal, explicit specification of a shared conceptualization [45]. They allow the specification of the concepts (viz. classes), individuals, and relationships (viz. roles) that exist in an application domain. As ontologies were born to provide a shared representation, it is quite common, especially in multidisciplinary application domains, that the ontology building phase involves some kind of consensus between the ontology developers. For example, the latitude and the longitude of a mountain could be represented as classes or properties, and an agreement is necessary.

In many real-world applications, there is a need to manage imprecise knowledge. The limitations of classical ontologies in such domains have been discussed in several fora. Fuzzy ontologies, specifically Fuzzy OWL 2 ontologies, based on fuzzy logic theory, are a very natural and established extension to manage imprecise knowledge [27, 42]. In fuzzy ontologies, an application domain is represented by using fuzzy concepts (with partial membership degrees), fuzzy relationships, fuzzy datatypes (described by fuzzy membership functions), and axioms that only hold to some degree of truth in $[0, 1]$.

One of the main features of fuzzy logic is the possibility to use so-called linguistic variables, i.e., variables whose values are words (or sentences) in a language; for example, HighMountain may be the value of a linguistic variable measuring the height of a mountain. Of course, linguistic variables and values are particularly subjective and context dependent. For instance, both Carrauntoohil

(Ireland) and Fuji (Japan) are usually considered as high mountains by local people, although they have very different elevations (1041 m and 3776 m, respectively).

In Fuzzy OWL 2 ontologies such linguistic variables are represented using fuzzy datatypes. To date, the only existing methods to build fuzzy datatypes are agnostic to any information about the domain (for example, computing a uniform partitioning of the domain [42]) or determine the fuzzy membership functions from real data (for example, using a previous clusterization of real data, as in Datil [23]). However, often no real data (or not enough data, or data with not enough quality) are available at the time of the fuzzy ontology building phase but there are some experts available to provide the definitions. To the best of our knowledge, how to reach a consensus in Fuzzy OWL 2 ontology building when several experts are involved has not received attention so far.

In this paper, we present a tool, called *Fudge*,[1] whose aim is to support the consensual creation of Fuzzy OWL 2 datatypes by aggregating the specifications given by a group of experts, e.g. by merging the definitions of HighMountain given by two Irish and Japanese experts. Fudge is open-source and supports currently different linguistic aggregation strategies, including the convex combination, linguistic OWA, weighted mean and fuzzy OWA and easily allows to built-in others. We also propose and have implemented two novel linguistic aggregation operators, based on a left recursive form of the convex combination and of the linguistic OWA. Fudge is also able to deal with missing data from some of the experts and can use fuzzy quantifiers to define a vector of weights.

The remainder of this paper is organized as follows. Firstly, Section 2 relates our work with other previous works. Section 3 provides some background on Fuzzy OWL 2 ontologies and aggregation operators. Then, Section 4 describes our current strategy to build consensual fuzzy datatypes from several experts and discusses two novel different aggregation operators. Next, Section 5 describes our implementation, while Section 6 discusses some uses cases and provides a numerical example. Finally, Section 7 sets out some conclusions and ideas for future work.

2 Related work

Fuzzy ontologies have already be proved to be useful in several applications, such as information retrieval [9, 44, 50], Semantic Web and the Internet [42], ambient intelligence [16], ontology merging [46], matchmaking [37], decision making [33], summarization [25], construction [20], robotics [18], diabetes diagnosis [19], design [35], human resources [43], decision making [41], gait recognition [5], and many others [28, 38]. For a more detailed overview, we refer the reader to [12].

Some works have previously considered aggregation operators in the fuzzy ontology field [7, 10, 26, 49], but the aggregation was restricted to numerical degrees of truth. Furthermore, most of the existing work assumes a unique definition of the fuzzy membership functions that define the fuzzy datatypes. In the following we will focus only on the few exceptions.

In a series of papers, researchers from the University of Granada and coauthors studied the application of fuzzy ontologies to reaching a consensus on decision making scenarios [29, 36]; in particular, they used existing fuzzy ontologies as part of decision making processes. In related papers, they also studied the process of building a fuzzy ontology; for example, [30, 31, 32] discussed how to build a fuzzy ontology in scenarios of multi-granular linguistic information.

However, please note that there are significant differences with respect to our proposal. The approach in [31] does not take account the opinions of different ontology developers to build a fuzzy ontology. On the other hand, the approach in [32] does not focus on fuzzy datatype construction but on assigning a membership degree to individuals of a concept. Finally, the approach in [30] considers users' opinions in social networks and computes the fuzzy membership function from sentiment

[1] Fudge works on desktop computers and mobile devices.

information (numbers of positive, neutral, and negative words). However, rather than computing fuzzy membership functions by aggregating the fuzzy membership functions of each ontology builder, the authors compute the definitions of the fuzzy membership functions directly from all the opinions of the different users. This depends on the quality of the sentiment analysis and is more vulnerable to malicious users.

To the best of our knowledge, none uses the de-facto standard for fuzzy ontology representation, namely Fuzzy OWL 2, supports several aggregation operators, and provides a publicly available and modifiable implementation.

3 Background

This section gives a quick overview on the main ingredients related to fuzzy ontologies and aggregation operators. The confident reader may directly skip to the next section.

3.1 Fuzzy OWL 2 ontologies

In fuzzy ontologies, the elements of a classical ontology are extended in such a way that concepts, relations, datatypes, and axioms are fuzzy. In particular:

- *Fuzzy concepts* and *fuzzy properties* are interpreted as fuzzy sets of individuals and fuzzy binary relations, respectively. For example, YoungHuman denotes the fuzzy set of young people, while hasClimbed denotes a fuzzy relationship between people and mountains that they climbed (maybe only partially).

- *Fuzzy axioms* express statements that are not either true or false but hold to some degree in $[0, 1]$. For example, we can state that john belongs to the concept of YoungHuman with at least degree 0.9, meaning that he is considerably young.

- *Fuzzy datatypes* generalize crisp values by using a fuzzy membership function. In fuzzy ontologies, it is common to restrict the membership functions to trapezoidal (Figure 1 (a)), triangular (Figure 1 (b)), left-shoulder (Figure 1 (c)), and right-shoulder (Figure 1 (d)) membership functions.[2] For example, instead of considering the age of an individual as the crisp value 18, one may consider a fuzzy set about18 defined via a triangular function centered at the value q2=18.

Fuzzy Description Logics (Fuzzy DLs) [4, 42] are the logical formalism behind Fuzzy OWL 2 ontologies [6], in a similar way as crisp Description Logics (DLs) [2] are the underlying logical formalism of OWL 2 [13]. Although there is not a W3C standard fuzzy ontology language, Fuzzy OWL 2 (based on a fuzzy DL) has become a de facto standard/popular choice. Fuzzy OWL 2 extends OWL 2 ontologies with OWL 2 annotations encoding fuzzy information using an XML-like syntax. The key idea of this representation is to start with an OWL 2 ontology created as usual, with a classical ontology editor. Then, it is possible to annotate the elements to represent the features of the fuzzy ontology that OWL 2 cannot directly encode. In particular, it is possible to annotate fuzzy axioms by adding a degree of truth, to represent fuzzy datatypes, and to define specific elements of fuzzy ontologies (such as fuzzy modifiers or aggregated concepts). Fuzzy OWL 2 is supported by the fuzzy ontology reasoner fuzzyDL [8]. For instance, Figure 2 illustrates how to annotate a fuzzy datatype by using the Fuzzy OWL 2 plug-in available for the Protégé ontology editor. (Note that q_1, q_2, q_3 are denoted as a, b, c, respectively.)

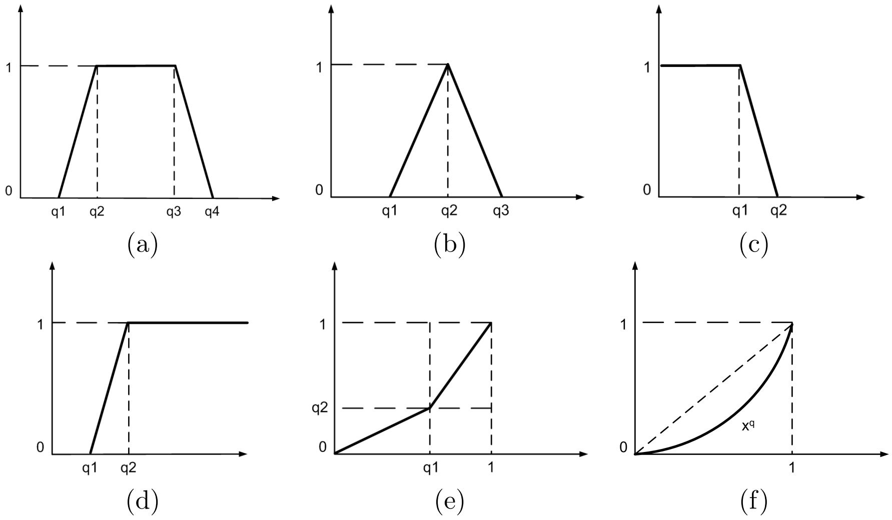
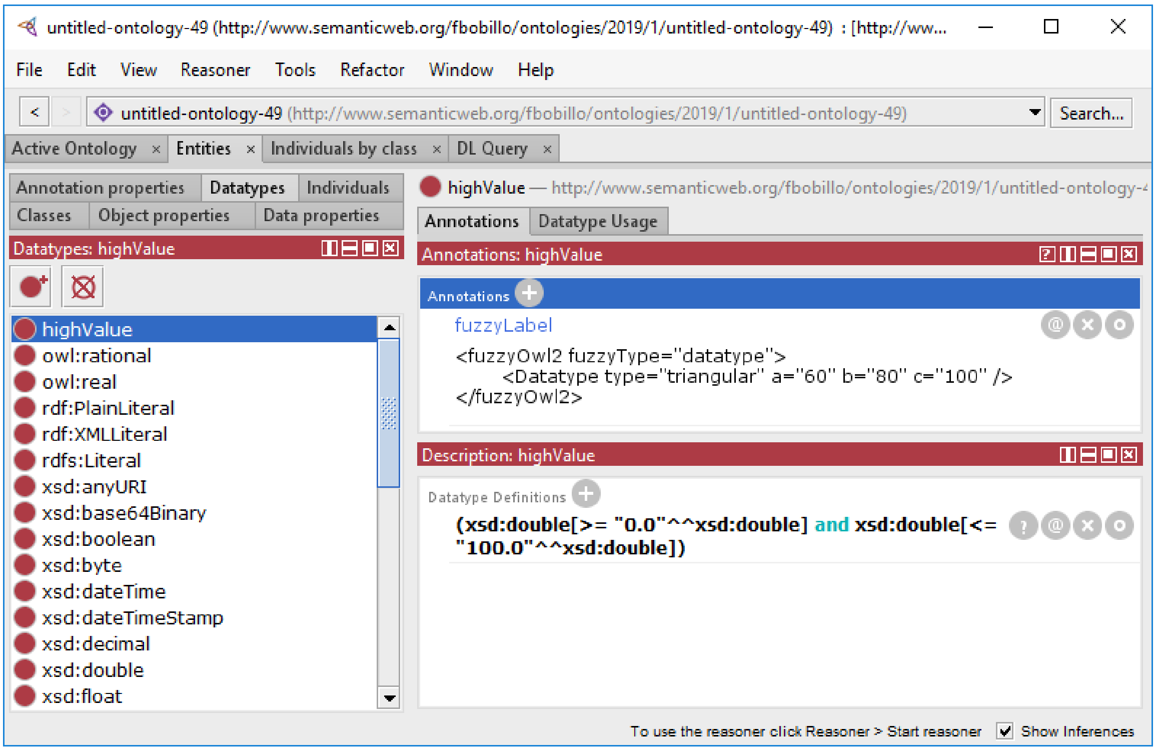


Figure 1: (a) Trapezoidal function; (b) Triangular function; (c) Left-shoulder function; (d) Right shoulder function; (e) Linear function; (f) Power function



```
<fuzzyOwl2 fuzzyType="datatype">
      <Datatype type="triangular" a="60" b="80" c="100" />
</fuzzyOwl2>
```

(xsd:double[>= "0.0"^^xsd:double] and xsd:double[<= "100.0"^^xsd:double])

Figure 2: Definition of the fuzzy datatype highValue using Fuzzy OWL 2

We refer the reader to [4, 6, 42] for further insights on Fuzzy OWL 2 and Fuzzy DLs.

3.2 Aggregation operators

Aggregation Operators (AOs) are mathematical functions that are used to combine different pieces of information. There is no standard definition of AO. Usually, given a domain $\mathbb{D}$, an AO is a mapping $@ : \mathbb{D}^K \to \mathbb{D}$, aggregating K values of K different criteria. Often, AOs use also a vector of weights $\boldsymbol{W} = [w_1, \dots, w_K]$ such that $w_i \in [0, 1]$ and $\sum_{i=1}^{K} w_i = 1$.

Let us consider first the most usual case, where $\mathbb{D} = [0, 1]$. Then, we will address the case where $\mathbb{D}$ is a linguistic label. For a more detailed overview on existing AOs, we refer the reader to [3] (for the numerical case) and [51] (for the linguistic case).

Aggregation of numerical degrees of truth Often, we would like to aggregate degrees of satisfaction of K different criteria $x_i \in [0, 1], i \in \{1, \dots, K\}$, that are the membership degrees to K fuzzy sets; the range of the AO is also $[0, 1]$. We will denote these AOs as $@([w_1, \dots, w_K], [x_1, \dots, x_K])$. A typical example of such an AO is the *weighted sum*:

$$\mathbf{WS}([w_1, \dots, w_K], [x_1, \dots, x_K]) = \sum_{i=1}^{K} w_i x_i \ . \tag{1}$$

Another very important family of AOs are the *Ordered Weighted Averaging* (OWA) operators [52]. Formally, an OWA operator of dimension K is an AO such that:

$$\mathbf{OWA}([w_1, \dots, w_K], [x_1, \dots, x_K]) = \sum_{i=1}^{K} w_i x_{\sigma(i)} \tag{2}$$

where $\sigma(i)$ is a permutation such that $x_{\sigma(1)} \geq x_{\sigma(2)} \geq \cdots \geq x_{\sigma(K)}$. A key aspect of these operators is the reordering step: a weight w_i is not associated with the i-th element x_i but with the i-th ordered element $x_{\sigma(i)}$.

Let us recall that the maximum and the minimum can be obtained by using $\mathbf{W} = [1, 0, \dots, 0]$ and $\mathbf{W} = [0, \dots, 0, 1]$, respectively, and that OWA verifies the *internality* property:

$$\min(x_1, \dots, x_n) \leq \mathbf{OWA}([w_1, \dots, w_K], [x_1, \dots, x_K]) \leq \max(x_1, \dots, x_n) \ . \tag{3}$$

An important measure of OWA operators is the *orness* [52], which depends only on the weighting vector:

$$orness([w_1, \dots, w_K]) = \frac{1}{K-1} \sum_{i=1}^{K} (K-i) w_i \ . \tag{4}$$

It is easy to check that $orness([w_1, \dots, w_K]) \in [0, 1]$, $orness([1, 0, \dots, 0]) = 1$ (the value is 1 for the maximum, a t-conorm) and that $orness([0, \dots, 0, 1]) = 0$ (the value is 0 for the minimum, a t-norm). Hence, the orness measures to what extent an AO behaves as a disjunctive operator (a t-conorm).

A common practical problem is how to compute the weights of an OWA operator. Two well-known solutions are the following: *(i)* using quantifier-based aggregation [54] and *(ii)* applying recursive OWA [48], as illustrated next:

[2] The functions in Figure 1 (e)–(f) will be addressed later in this paper, as they will be used to compute weights from fuzzy quantifiers.

- In quantifier-based aggregation, the vector of weights $\boldsymbol{W}$ can be defined using a fuzzy quantifier [54]. A proportional fuzzy quantifier $Q : [0, 1] \to [0, 1]$ is a fuzzy subset such that for each $r \in [0, 1]$, the membership grade $Q(r)$ indicates the degree to which the proportion r satisfies the linguistic quantifier that Q represents. For the sake of illustrative purposes, we will consider here the *Regular Increasing Monotone* (RIM) quantifiers [53], which satisfy the boundary conditions $Q(0) = 0$ and $Q(1) = 1$, and are monotone increasing, i.e., $Q(x_1) \leq Q(x_2)$ when $x_1 \leq x_2$. Essentially, these quantifiers reflect the notion that as the proportion increases, the degree of satisfaction does not decrease. Given a RIM Q, the weights of an OWA weighting vector of dimension K can be computed as ($i \in \{1, ..., K\}$)

$$w_i = Q(\frac{i}{K}) - Q(\frac{i-1}{K}) \ . \tag{5}$$

  Note that indeed $w_i \in [0, 1]$ and $\sum_{i=1}^{K} w_i = 1$.

- Recursive OWA starts from a desired value for the orness of the OWA operator and computes the weights in two recursive ways, a *Left Recursive Form* (LRF) and a *Right Recursive Form* (RRF). LRF is obtained after rewriting Eq. 2 as:

$$\begin{gathered}\mathbf{OWA}([w_1^K, \dots, w_K^K], [x_1, \dots, x_K]) = \\ v_L^K \cdot \mathbf{OWA}([w_1^{K-1}, \dots, w_{K-1}^{K-1}], [x_{\sigma(1)}, \dots, x_{\sigma(K-1)}]) + (1 - v_L^K) \cdot x_{\sigma(K)} \ ,\end{gathered} \tag{6}$$

  where the weights are defined as

$$\begin{aligned} v_L^K &= \frac{(K-1) \cdot \text{orness}}{(K-2) \cdot \text{orness} + 1} \\ w_i^K &= v_L^K \cdot w_i^{K-1}, \ i \in \{1, \dots, K-1\} \\ w_K^K &= 1 - v_L^K \ . \end{aligned} \tag{7}$$

  The base case happens when $K = 2$:

$$\mathbf{OWA}([w_1^2, w_2^2], [x_1, x_2]) = v_L^2 \cdot x_{\sigma(1)} + (1 - v_L^2) \cdot x_{\sigma(2)} \tag{8}$$

  Similarly, RRF is obtained after rewriting Eq. 2 as:

$$\begin{gathered}\mathbf{OWA}([w_1^K, \dots, w_K^K], [x_1, \dots, x_K]) = \\ (1 - v_R^K) \cdot x_{\sigma(1)} + v_R^K \cdot \mathbf{OWA}([w_2^{K-1}, \dots, w_K^{K-1}], [x_{\sigma(2)}, \dots, x_{\sigma(K)}]) \ ,\end{gathered} \tag{9}$$

  where the weights are defined as

$$\begin{aligned} v_R^K &= \frac{(K-1)(1 - \text{orness})}{(K-2)(1 - \text{orness}) + 1} \\ w_i^K &= v_R^K \cdot w_i^{K-1}, \ i \in \{1, \dots, K-1\} \\ w_K^K &= 1 - v_R^K \ . \end{aligned} \tag{10}$$

Aggregation of fuzzy linguistic values Although this case is not as usual as the aggregation of numerical values, some aggregation operators to combine fuzzy linguistic numbers have been proposed [51]. Among them, we will consider here four of them: convex combination (CONV–RRF[3]), linguistic OWA (LOWA–RRF), weighted mean (WMEAN), and fuzzy OWA (FOWA).

To start with, we will assume a vector of weights $\boldsymbol{W}$ and a permutation σ over fuzzy numbers/-values (see, e.g., [24, 40]) such that $d_{\sigma(1)} \geq d_{\sigma(2)} \geq \cdots \geq d_{\sigma(K)}$ (that is, $d_{\sigma(i)}$ denotes the i-th largest fuzzy number).[4] Now,

1. CONV–RRF was proposed by Delgado et al. [15]. It assumes a fixed vector of possible values for the linguistic variables $\mathcal{L} = [l_1, \ldots, l_L]$ such that $l_i < l_j$ if $i < j$. CONV–RRF has a recursive definition. Let us start with the base case where we want to aggregate $K = 2$ fuzzy values $d_i \in \mathcal{L}$. The CONV–RRF of a vector $[d_1, d_2]$ given a vector of weights $\boldsymbol{W} = [w_1, w_2]$, is defined as
$$\mathbf{CONV^{RRF}}([w_1, w_2], [d_1, d_2]) = l_c \tag{11}$$
where $d_{\sigma(1)} = l_j, d_{\sigma(2)} = l_i, c = i + round(w_{\sigma(1)} \cdot (j - i))$. Note that the permutation is applied both to the weights and to the values to be aggregated, so each w_i is associated to the value d_i. Now, if $K > 2$, then:
$$\begin{gathered}\mathbf{CONV^{RRF}}([w_1, \ldots, w_K], [d_1 \ldots, d_K]) = \\ \mathbf{CONV^{RRF}}\Big([w_{\pi(1)}, 1 - w_{\pi(1)}], \Big[d_{\pi(1)}, \mathbf{CONV^{RRF}}([\beta_2, \ldots, \beta_k], [d_{\pi(2)}, \ldots, d_{\pi(K)}])\Big]\Big) \ ,\end{gathered} \tag{12}$$
where $\beta_h = w_{\pi(h)} \ / \ \sum_{j=2}^{K} w_{\pi(j)}, h \in \{2, \ldots, K\}$.

2. LOWA–RRF is a variant of CONV–RRF using a reordering step as in standard OWA [21]:
$$\begin{gathered}\mathbf{LOWA^{RRF}}([w_1, \ldots, w_K], [d_1 \ldots, d_K]) = \\ \mathbf{CONV^{RRF}}([w_1, \ldots, w_K], [d_{\sigma(1)}, \ldots, d_{\sigma(K)}]) \ .\end{gathered} \tag{13}$$

3. WMEAN is an extension of the classical weighted mean to deal with trapezoidal fuzzy numbers [17]. Specifically, given K trapezoidal fuzzy numbers $d_i = \mathbf{trap}(q_1^i, q_2^i, q_3^i, q_4^i)$ and a vector of K weights $\mathbf{W}$, the result is:
$$\begin{gathered}\mathbf{WMEAN}([w_1, \ldots, w_K], [d_1, \ldots, d_K]) = \\ \mathbf{trap}\left(\sum_{i=1}^{K} w_i q_1^i, \sum_{i=1}^{K} w_i q_2^i, \sum_{i=1}^{K} w_i q_3^i, \sum_{i=1}^{K} w_i q_4^i\right) \ .\end{gathered} \tag{14}$$
In general, the result of the aggregation is not any of the original aggregated values, and the weights describe the importance of each expert.

4. FOWA is a variant of WMEAN with a reordering step [11], so that w_1 is associated to the largest fuzzy number. Given a permutation σ such that $d_{\sigma(i)}$ denotes the i-th largest trapezoidal fuzzy number, FOWA is defined as:
$$\begin{gathered}\mathbf{FOWA}([w_1, \ldots, w_K], [d_1, \ldots, d_K]) = \\ \mathbf{trap}\left(\sum_{i=1}^{K} w_i q_1^{\sigma(i)}, \sum_{i=1}^{K} w_i q_2^{\sigma(i)}, \sum_{i=1}^{K} w_i q_3^{\sigma(i)}, \sum_{i=1}^{K} w_i q_4^{\sigma(i)}\right) \ .\end{gathered} \tag{15}$$

[3] We write CONV–RRF and LOWA–RRF to distinguish them from the new operators CONV–LRF and LOWA–LRF that we will define in Section 4.

[4] Of course, this requires an order relation between fuzzy numbers that we will discuss later.

Ordering fuzzy linguistic labels In CONV–RRF, LOWA–RRF, and FOWA, we need to define an ordering between fuzzy linguistic labels. Many approaches can be found in the literature (see, e.g., [24, 40]), but we will rely here on Delgado et al. [14] transformation function $\mathbf{transform}(d)$ from a linguistic domain (trapezoidal fuzzy numbers d) to a numerical domain, so that $d_1 \geq d_2$ iff $\mathbf{transform}(d_1) \geq \mathbf{transform}(d_2)$:

$$\mathbf{transform}(\mathbf{trap}(q_1, q_2, q_3, q_4)) = \frac{8(q_3 + q_2)H + (q_4 + q_1)H + 8(H + q_3 q_4 - q_1 q_2)}{24H} , \qquad (16)$$

where $H = q_4 + q_3 - q_2 - q_1$ and $q_1 = q_2 = q_3 = q_4$ is assumed not to hold.

4 Consensual aggregation of fuzzy datatypes

In this section we start by illustrating how to use aggregation operators to build consensual fuzzy datatypes. Then, we discuss two novel aggregation operators, all of them supported in the Fudge tool.

4.1 Using aggregation operators to build fuzzy datatypes

At first, let us define formally the problem we will address. We assume that there is a group of experts $E_1, E_2, \ldots, E_N$ providing the definitions of the membership functions $F_1, F_2, \ldots, F_N$ characterizing several fuzzy datatypes from a fuzzy ontology. d_{ij} denotes the definition of the datatype F_i according to expert E_j. d_{ij} is assumed to be a linguistic value. Furthermore, there could be missing data, i.e., expert E_j might not provide his/her definition of some datatype F_i. Therefore, for each datatype F_i we have a number of definitions denoted K, with $K \leq N$. Our objective is to define each F_i as a consensus of the definitions $\langle d_{i1}, d_{i2}, \ldots, d_{iK} \rangle$. We will sometimes omit the subscript i when the particular F_i is not important.

We assume that all the definitions are given using trapezoidal fuzzy numbers of the form $\mathbf{trap}(q_1, q_2, q_3, q_4)$, as they are those supported by Fuzzy OWL 2. Note that triangular (denoted **tri**), right-shoulder (denoted **right**), and left-shoulder (denoted **left**) functions can be represented as trapezoidal fuzzy numbers, provided that right-shoulder and left-shoulder functions are defined over a fixed range $[r_1, r_2]$. For example, a left-shoulder function $\mathbf{left}(q_1, q_2)$ can be represented as $\mathbf{trap}(r_1, r_1, q_1, q_2)$.

Now, for each fuzzy datatype F_i, we compute $@(\mathbf{W}, [d_{i1}, d_{i2}, \ldots, d_{iK}])$ as a consensual definition, for some aggregation operator @ taking as input a vector of numerical weights $\mathbf{W}$ and a vector of trapezoidal fuzzy numbers, returning as output a trapezoidal fuzzy number, and satisfying internality. Possible choices for the aggregation operator include CONV, LOWA, WMEAN, and FOWA.

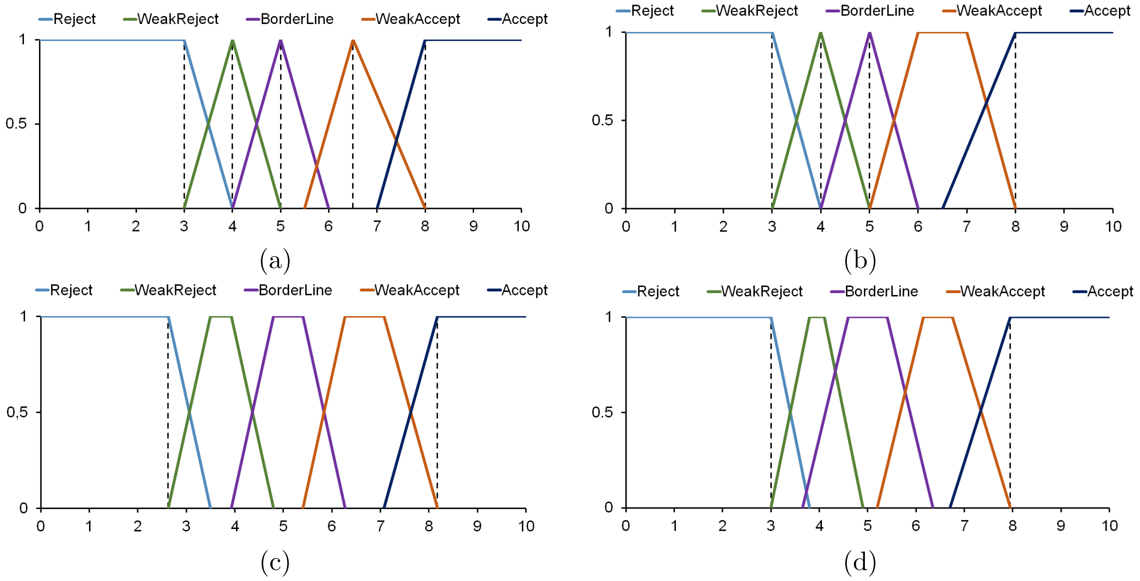
An advantage of such a consensus process is that the individual opinions of the ontology builders are only used to build the final consensual values, thus respecting the privacy of the experts by hiding their individual opinions.

Example 1 *For the sake of illustrative purposes, let us consider the problem of paper reviewing. We assume that we want to build a general fuzzy ontology with the relevant definitions (e.g., a hierarchy of publication types, the steps of the reviewing process, the different roles that take part in the process, etc.) to reuse it in other applications, to enable interoperability, or to detect inconsistencies automatically.*

We assume that there are 5 possible decisions for a submission (Reject, WeakReject, Borderline, WeakAccept, Accept) and that we need to define their definitions by aggregating the definitions given *by* 4 *experts (the coauthors of our work). For example, these categories could correspond to reject, reject and encourage re-submission, major revision, minor revision, and accept as it is, respectively. Table* 1 *shows the definition of the fuzzy datatypes given by all the experts.*

Table 2 *shows instead the consensual aggregation for several aggregation functions. Note that for CONVEX RRF and WMEAN, we used a vector of weights* $\boldsymbol{W} = [0.2, 0.25, 0.25, 0.3]$ *taking into account the experience (years in academia) of the experts, while for LOWA–RRF and FOWA we used a fuzzy quantifier* $\textbf{right}(0.3, 0.8)$, *leading to a vector of weights* $\boldsymbol{W} = [0, 0.4, 0.5, 0.1]$ *for FOWA.*

The resulting datatypes for each aggregation strategy are illustrated in Figure 3 *(a)–(d). It is worth to note that all methods result in a left-shoulder and a right-shoulder function, but there are differences for the rest of functions: CONV–RRF result in* 3 *triangular and* 0 *trapezoidal functions, LOWA result in* 2 *triangular and* 1 *trapezoidal functions, and both WMEAN and FOWA result in* 3 *trapezoidal and* 0 *triangular functions. Note also that CONV–RRF and LOWA–RRF result in* 4 *datatypes with the definitions given by Expert 3; CONV–RRF and LOWA–RRF coincide in* 3 *definitions.*



(a)

(b)

(c)

(d)

Figure 3: Consensual datatypes using (a) CONV; (b) LOWA; (c) WMEAN; and (d) FOWA

Now, let us discuss how to evaluate a given submission. We assume that each submission has a numerical score from 0 *to* 10 *that combines the evaluation of several criteria (e.g., originality, technical soundness, significance, presentation, and relevance) given by different reviewers.*[5] *For a paper* p_0 *with a score of* 7.5*, the membership degrees to each category are shown in Table* 3.

Therefore, the optimal decision depends on the aggregation strategy: for WMEAN, it is WeakAccept*, while for the other ones the optimal choice is* Accept.

Computing the vector of weights In LOWA or FOWA, we propose two different strategies to obtain the vector of weights. Namely:

[5] This involves the aggregation of numerical values, which is a well-known problem and out of our scope.

Alternative	Expert 1	Expert 2	Expert 3	Expert 4
Reject	$\mathbf{left}(3.5, 4.5)$	$\mathbf{left}(3.5, 4)$	$\mathbf{left}(3, 4)$	$\mathbf{left}(1, 2)$
WeakReject	$\mathbf{tri}(3.5, 4.5, 5.5)$	$\mathbf{trap}(3.5, 4, 4.5, 5)$	$\mathbf{tri}(3, 4, 5)$	$\mathbf{trap}(1, 2, 3, 4)$
Borderline	$\mathbf{tri}(4.5, 5.5, 6.5)$	$\mathbf{tri}(4.5, 5, 5.5)$	$\mathbf{tri}(4, 5, 6)$	$\mathbf{trap}(3, 4, 6, 7)$
WeakAccept	$\mathbf{tri}(5.5, 6.5, 8)$	$\mathbf{trap}(5, 5.5, 6.5, 7.5)$	$\mathbf{trap}(5, 6, 7, 8)$	$\mathbf{trap}(6, 7, 8, 9)$
Accept	$\mathbf{right}(6.5, 8)$	$\mathbf{right}(6.5, 7.5)$	$\mathbf{right}(7, 8)$	$\mathbf{right}(8, 9)$

Table 1: Individual definitions of the decisions given by the experts

Alternative	CONV–RRF	LOWA–RRF	WMEAN	FOWA
Reject	$\mathbf{left}(3, 4)$	$\mathbf{left}(3, 4)$	$\mathbf{left}(2.63, 3.5)$	$\mathbf{left}(3, 3.8)$
WeakReject	$\mathbf{tri}(3, 4, 5)$	$\mathbf{tri}(3, 4, 5)$	$\mathbf{trap}(2.63, 3.5, 3.93, 4.8)$	$\mathbf{trap}(3, 3.8, 4.1, 4.9)$
Borderline	$\mathbf{tri}(4, 5, 6)$	$\mathbf{tri}(4, 5, 6)$	$\mathbf{trap}(3.93, 4.8, 5.4, 6.28)$	$\mathbf{trap}(3.65, 4.6, 5.4, 6.35)$
WeakAccept	$\mathbf{tri}(5.5, 6.5, 8)$	$\mathbf{trap}(5, 6, 7, 8)$	$\mathbf{trap}(5.4, 6.28, 7.075, 8.18)$	$\mathbf{trap}(5.2, 6.15, 6.75, 7.95)$
Accept	$\mathbf{right}(7, 8)$	$\mathbf{right}(6.5, 8)$	$\mathbf{right}(7.08, 8.175)$	$\mathbf{right}(6.7, 7.95)$

Table 2: Consensual definitions of the decisions for 4 aggregation strategies

- Quantifier-guided aggregation, using Eq. 5 as in standard OWA [54]. In this case, we propose to use right-shoulder (Figure 1 (d)), linear (Figure 1 (e)), and power (Figure 1 (f)) functions as RIM quantifiers.
- A recursive procedure to compute a vector of weights with a given orness, using either Eq. 7 (to combine the lowest value and the aggregation of the other ones) or Eq. 10 (to combine the highest value and the aggregation of the other ones) [48].

Dealing with incomplete data It could be the case that some of the experts do not provide his/her definition of some datatype. In this case, unavailable opinions are not taken into account during the aggregation, so a new vector of weights $\boldsymbol{W}$ is computed. Specifically, in CONV–RRF and WMEAN, we can normalize each weight dividing by the sum of the weights of the available experts. In LOWA–RRF or FOWA, we can use the previously described strategies (quantifier-guided aggregation or a recursive procedure starting from the orness) to get a vector with a smaller size.

Example 2 *Assume that there are 4 experts E_1, E_2, E_3, and E_4, but E_3 does not provide a definition for some datatype. To aggregate the other definitions using CONV–RRF or WMEAN, the initial vector of weights $[w_1, w_2, w_3, w_4]$ can be updated as*

$$[\frac{w_1}{w_1 + w_2 + w_4}, \frac{w_2}{w_1 + w_2 + w_4}, \frac{w_4}{w_1 + w_2 + w_4}] .$$

To aggregate the available definitions using LOWA-RRF or FOWA, let us firstly assume that

Alternative	CONV–RRF	LOWA–RRF	WMEAN	FOWA
Reject	0	0	0	0
WeakReject	0	0	0	0
Borderline	0	0	0	0
WeakAccept	0.33	0.5	**0.61**	0.38
Accept	**0.5**	**0.67**	0.39	**0.64**

Table 3: Degree of satisfaction of each decision for a paper with score of 7.5

the vector of weights was computed using a quantifier $Q = \textbf{right}(0.3, 0.8)$. *Then, the initial vector* $[0, 0.4, 0.5, 0.1]$ *is replaced with* $[0.067, 0.667, 0.267]$.

Now let us assume that the vector of weights was computed from a desired orness 0.6 *using a left recursive procedure. Then, the initial vector* $[0.368, 0.245, 0.205, 0.182]$ *is replaced with* $[0.45, 0.3, 0.25]$.

4.2 Some properties of the linguistic aggregation operators

The first thing to observe is that CONV–RRF (and hence LOWA–RRF) does not care about the concrete definitions (e.g., if a trapezoidal function has some value of q_1 or another); only the relative ordering matters.

In our case, when using CONV–RRF, rather than assuming a global vector of linguistic labels $\mathcal{L}$ (such as VeryLow, Low, Neutral, High, and VeryHigh), for each fuzzy datatype F_i we assume a different vector $\mathcal{L}_i = [l_{i1}, l_{i2}, \dots]$. On the one hand, it does not make sense to use the same label VeryHigh to define a mountain or a temperature. On the other hand, the linguistic labels are not defined a priori, so we can only rely on the definitions given by the experts. Furthermore, the same of labels can be different for each fuzzy datatype F_i, as some experts might not provide a definition. Thus, for each F_i, we define $\mathcal{L}_i = [d_{i\sigma(K)}, \dots, d_{i\sigma(1)}]$. Therefore, the result of the aggregation is always one of the aggregated values, so a single expert (or a group of them giving the same value) is the single provider of the consensual definition. The same holds in LOWA as well.

It is worth to note that a pair of labels $l_{ij}, l_{jk} \in F_i$ may have the same definition. That is, although it is probably not very common in practice, two experts may use the very same trapezoidal membership function to define a label. This case was not originally considered in [15], where the authors assumed that $k > j$. However, we consider this case because the combining the same definition is not trivial, as Example 3 shows.

Example 3 *Assume that there are* 5 *experts and that* E_1 *and* E_2 *provide the same definition, so we have* $\mathcal{L} = [\textsf{Exp1\&2}, \textsf{Exp3}, \textsf{Exp4}, \textsf{Exp5}]$ *(for ease of presentation, we assume that the i-th expert provides the i-th largest value). Given a weighting vector* $[0.15, 0.15, 0.1, 0.5,\ 0, 1]$*, one may verify that the consensual definition using CONV–RRF is that of expert* E_3*:*

$$\mathbf{CONV^{RRF}}\Big([0.15, 0.15, 0.1, 0.5, 0, 1], [\textsf{Exp1\&2}, \textsf{Exp1\&2}, \textsf{Exp3}, \textsf{Exp4}, \textsf{Exp5}]\big) = \textsf{Exp3}$$

Note that if we group the opinion of the two first experts and assign to this new value the sum of their weights, the result of CONV–RRF is different:

$$\mathbf{CONV^{RRF}}\Big([0.3, 0.1, 0.5, 0, 1], [\textsf{Exp1\&2}, \textsf{Exp3}, \textsf{Exp4}, \textsf{Exp5}]\big) = \textsf{Exp4}$$

Table 4 summarizes some key features of the four methods, namely if the output is always one of the inputs or not, and if weights are assigned to a specific expert or not.

Criterion	CONV–RRF	LOWA–RRF	WMEAN	FOWA
Output is always one of the input datatypes	Yes	Yes	No	No
Weights are assigned to a specific expert	Yes	No	Yes	No

Table 4: Comparison between the four aggregation strategies

4.3 Left Recursive Form of CONV and LOWA

In this section, we will propose some new linguistic aggregation operators. Inspired by the rewriting of classical OWA in two recursive forms LRF and RRF (see Eqs. 6 and 9), we may view the standard definition of CONV (Eqs. 11–12) as a right recursive form. From that, we propose a left recursive form (CONV–LRF).

Definition 1 *The* Left Recursive Form of the convex combination *(CONV–LLF) of $K \geq 2$ linguistic labels given a weighting vector $[w_1, \ldots, w_K]$ is defined as follows:*

- *if $K = 2$, then*

$$\mathbf{CONV^{LRF}}([w_1, \ldots, w_K], [d_1 \ldots, d_K]) \quad = \quad \mathbf{CONV^{RRF}}([w_1, \ldots, w_K], [d_1 \ldots, d_K])$$

- *if $K > 2$, then:*

$$\mathbf{CONV^{LRF}}([w_1, \ldots, w_K], [d_1 \ldots, d_K]) = \mathbf{CONV^{LRF}}\Big([1 - w_{\pi(K)}, w_{\pi(K)}], \Big[\mathbf{CONV^{LRF}}([\beta_1, \ldots, \beta_{k-1}], [d_{\pi(1)}, \ldots, d_{\pi(K-1)}]), d_{\pi(K)}\Big]\Big) \ , \tag{17}$$

where $\beta_h = w_{\pi(h)} \ / \ \sum_{j=1}^{K-1} w_{\pi(j)}, h \in \{1, \ldots, K-1\}$.

Example 4 *4 experts provide definitions* $[\mathsf{\textit{VeryHigh}}, \mathsf{\textit{High}}, \mathsf{\textit{Low}}, \mathsf{\textit{VeryLow}}]$ *and there is a weighting vector* $[0.45, 0.05, 0.1, 0.4]$*. Let us firstly aggregate using CONV–RRF:*

$\mathbf{CONV^{RRF}}([0.45, 0.05, 0.1, 0.4], [\textit{VeryHigh}, \textit{High}, \textit{Low}, \ \textit{VeryLow}]) =$
$\mathbf{CONV^{RRF}}\Big(\Big[0.45, 0.55\Big], \Big[\textit{VeryHigh},$
$\quad \mathbf{CONV^{RRF}}([0.09, 0.91], [\textit{High}, \mathbf{CONV^{RRF}}([0.2, 0.8], [\textit{Low}, \textit{VeryLow}])\Big]\Big) =$
$\mathbf{CONV^{RRF}}\Big(\Big[0.45, 0.55\Big], \Big[\textit{VeryHigh}, \mathbf{CONV^{RRF}}([0.09, 0.91], [\textit{High}, \textit{VeryLow}])\Big]\Big) =$
$\mathbf{CONV^{RRF}}\Big(\Big[0.45, 0.55\Big], \Big[\textit{VeryHigh}, \textit{VeryLow}\Big]\Big) = \textit{Low}$

Now, let us compute CONV–LRF, obtaining a different result:
$\mathbf{CONV^{LRF}}([0.45, 0.05, 0.1, 0.4], [\textit{VeryHigh}, \textit{High}, \textit{Low}, \ \textit{VeryLow}]) =$
$\mathbf{CONV^{LRF}}\Big(\Big[0.6, 0.4\Big], \Big[\mathbf{CONV^{LRF}}([0.83, 0.17],$
$\quad [\mathbf{CONV^{LRF}}([0.9, 0.1], [\textit{VeryHigh}, \textit{High}]), \textit{Low}]), \textit{VeryLow}\Big]\Big) =$
$\mathbf{CONV^{LRF}}\Big(\Big[0.6, 0.4\Big], \Big[\mathbf{CONV^{LRF}}([0.83, 0.17], [\textit{VeryHigh}, \textit{Low}]), \textit{VeryLow}\Big]\Big) =$
$\mathbf{CONV^{LRF}}\Big(\Big[0.6, 0.4\Big], \Big[\textit{VeryHigh}, \textit{VeryLow}\Big]\Big) = \textit{High}$

To understand the differences between CONV–LRF and CONV–RRF, we computed the results of the aggregation of four different values with different vectors of weights. Specifically, we evaluated each possible arrangement of degrees $w_i = k \cdot 0.01, \sum_{i=1}^{4} w_i = 1, k \in \{1, 2, \ldots, 100\}, i \in \{1, 2, 3, 4\}$. Out of the 156849 possibilities, CONV–LRF and CONV–RRF give the same value in 114259 cases (72.85 %). In 39538 cases (25.2 %), CONV–LRF returns a higher value, whereas in 3052 cases (1.95 %) CONV–RRF returns a higher value. Therefore, CONV–LRF seems to have a slightly higher orness than CONV–RRF. If the weights associated to the higher value or to the smaller value were greater than 0.83, both approaches coincided. This was also the case when the other weights were greater than 0.74.

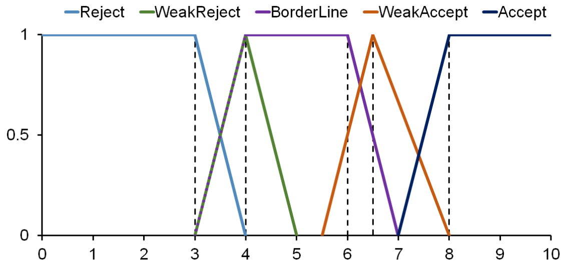


Figure 4: Consensual datatypes using CONV–LRF

CONV–LRF can also be used to define a new version of the linguistic OWA based on the left recursive form of the convex combination.

Definition 2 *The* Left Recursive Form of the linguistic LOWA *(LOWA–LLF) of $K \geq 2$ linguistic labels given a weighting vector $[w_1, \ldots, w_K]$ is defined as follows:*

$$\begin{gathered}\mathbf{LOWA^{LRF}}([w_1, \ldots, w_K], [d_1 \ldots, d_K]) = \\ \mathbf{CONV^{LRF}}([w_1, \ldots, w_K], [d_{\sigma(1)}, \ldots, d_{\sigma(K)}]) \ .\end{gathered} \tag{18}$$

Example 5 *Let us revisit Example 1, considering CONV–LRF and LOWA–LRF but not their right recursive forms. It turns out that LOWA–LRF produces exactly the same output, and that CONV–LRF only differs in the consensuated definition of Borderline, which is now given by* **trap**$(3, 4, 6, 7)$*, i.e., the definition given by Expert 4. The resulting datatypes for CONV–LRF are illustrated in Figure 4.*

Using CONV–LRF, the membership degree of paper p_0 to Borderline is also 0, as it happens using CONV–RRF. Both CONV–LRF and LOWA–LRF return 2 triangular and 1 trapezoidal functions (CONV–RRF returns 3 triangular functions), CONV–LRF includes 3 datatypes defined by Expert 3 (CONV–RRF includes 4); and CONV–LRF and LOWA–LRF coincide in 2 definitions (the right recursive forms coincide in 3).

CONV–LRF and LOWA–LRF behave as their right recursive form counterparts with respect to the features analyzed in Table 4.

5 Implementation: Fudge

We have developed an implementation of the consensual aggregation of fuzzy datatypes described in the previous section. Our tool is called *Fudge* (FUzzy Datatypes from a Group of Experts) and is available online[6]. The application has two versions (for desktop computers and for mobile devices) and uses OWL API [22] to manage (fuzzy) OWL 2 ontologies represented in Fuzzy OWL 2 language. Some snapshots of the tool can be found in A.

Fudge receives a folder containing the input ontology files, and it imports all the .owl files in the folder. We assume that each of the input files includes a Fuzzy OWL 2 ontology —specifically, an OWL 2 ontology where datatypes can have an OWL 2 annotation describing the parameters of the

[6] http://webdiis.unizar.es/~ihvdis/Fudge.html

fuzzy function. As an output, Fudge creates a new ontology with the axioms included in the input files, except the declarations of the datatypes, which are unique. That is, if two or more files have a datatype with the same name, it only adds a consensual one to the output ontology.

In theory, all input files should contain the same axioms (ontology schema and individuals), and only the datatype annotations may be different. In practice, it could happen that not all ontologies contain the same axioms. In such cases, there are several possible choices: adding to the output ontology axioms that are in all input ontologies, adding axioms that are in some of the ontologies, adding axioms that are in most of the ontologies, etc. Among them, we chose to add the axioms included in the input ontology with a larger number of logical axioms.

Fudge considers as a name of an entity its full URI (e.g., http://sid.cps.unizar.es/engines.owl#HighTemperature) rather than its fragment identifier (e.g., HighTemperature), as two experts could use the same fragment to denote two different entities (e.g., in the car domain, temperature of an engine and temperature of oil).

Note that some datatype may not be annotated in some of the input files. In such cases, only the existing annotations are taken into account, and a vector of weights of the appropriate size is computed, as already discussed in the previous section.

The declaration of the fuzzy datatypes must conform the specification of Fuzzy OWL 2, including an annotation (with the type of the membership function and the values of the parameters) and a range restriction to an interval $[r_1, r_2]$. The current implementation is restricted to trapezoidal, triangular, left-shoulder, and right-shoulder functions, accordingly to the Fuzzy OWL 2 specification.

So far, the supported aggregation operators are CONV–LRF, CONV–RRF, WMEAN, LOWA–LRF, LOWA–RRF, and FOWA. It is worth to stress that the application has been designed to ensure that adding more aggregation operators is very easy. Indeed, it is enough to *(i)* add a new class extending an existing one, *(ii)* implement a method computing the aggregation of K trapezoidal functions, and *(iii)* update the graphical interface by adding another item to a list of aggregation operators.

The following fragment of code illustrates the creation of the new class:

```
public class NewAO extends AggregationOperator
{
   @Override
   public TrapezoidalFuzzyNumber aggregate(
      ArrayList<TrapezoidalFuzzyNumber> values, Double[] weights)
      {
         ...
   }
}
```

To obtain the weights for LOWA–LRF, LOWA–RRF, and FOWA one may use quantifier-base aggregation (using right-shoulder, linear, and power functions) or two recursive procedures starting from a given orness.

5.1 Desktop computers

The versions for desktop computers is written in Java. A simple user interface allows to select the input ontologies, the type of consensus (aggregation operator) and the necessary parameters: a vector of weights for CONV–LRF, CONV–RRF, and WMEAN, and the type of fuzzy quantifier and its parameters for LOWA–LRF, LOWA–RRF, and FOWA. Figure 6 shows the main tab of the user interface. Initially, the second and the third tab are disabled.

If the user selects CONV–LRF, CONV–RRF, or WMEAN as the aggregation operator, the fourth tab is enabled, as shown in Figure 7. In this case, Fudge checks that all values are positive and normalized, and that the sum is equal to 1.

If the user selects LOWA–LRF, LOWA–RRF, or FOWA as an aggregation operator, the second and the third tab become enabled. The second tab allows to obtain the weights from a fuzzy quantifier. Figures 8–10 show how to select the type of fuzzy quantifier (right-shoulder, linear, and power functions) and their parameters. The user can see a general picture of the selected fuzzy quantifier and a customized picture with the values of the selected parameters (Figure 11). Fudge checks that all values are correct; for instance, in a right-shoulder function, $q_2 \geq q_1$. The third tab allows to use a recursive procedure to obtain the weights. Figure 12 shows that the user can select the type of recursive (left or right) and an orness value in $[0, 1]$.

5.2 Mobile devices

The version for mobile devices is written in Android. Since the version for desktop computers was written in Java, the adaption was relatively easy and only the graphical interface needed to be changed.

Figure 13 (a) shows the main activity, where the user can select the input fuzzy ontologies and the aggregation operator. Depending on the aggregation operator, s/he will be able to select different ways to compute the weights. For example, Figure 13 (b) shows how to obtain the weights for LOWA–LRF, LOWA–RRF, and FOWA from a quantifier; a customized picture of the quantifier is shown in Figure 13 (c). Figure 14 (a) shows to obtain the weights using the recursive procedure, an example of vector is shown in Figure 14 (b). Furthermore, Figure 14 (c) shows how to manually set the vector of weights for CONV–LRF, CONV–RRF, and WMEAN.

6 Use case: location-based services

In this section we provide a use case illustrating the usefulness of having consensual fuzzy datatypes in the field of location-based services by discussing how to extend the SHERLOCK system [55].

SHERLOCK (System for Heterogeneous mobilE Requests by Leveraging Ontological and Contextual Knowledge) is a general and flexible system that provides Location-Based Services (LBSs) using semantic technologies and mobile agents. The output of the system depends indeed on the location of the user, but also on his/her context, understood in a more general sense. SHERLOCK is a knowledge-based system that uses ontologies to represent the knowledge of an application domain (objects and interesting areas in a scenario) and the user context. Then, an ontology reasoner is used to discover implicit knowledge which is taken it account to answer user requests. The system can be used from both fixed and mobile devices. It has been applied to different application scenarios, such as recommendation of transportation means, helping firefighting, camera selection for live broadcasting of sport events, or emergency management.

Let us focus on the recommendation of transportation means (see Figure 5). Sherlock uses some background knowledge, stating for example that PublicTransport is a subclass of TransportationMean, and that Cab and Tram are subclasses of TransportationMean. Such general schema is populated with concrete instances. For example, cabLicense4815162342 is an instance of Cab tramLine1 is an instances of Tram. By using an ontology reasoner, both instances can be inferred to belong to the class TransportationMean. Each transportation mean has some properties that can be used to compute the best alternative, such location, direction, speed, or price.

So far, it is possible to ask SHERLOCK to retrieve transportation means that satisfy some strict requirements specified with thresholds, e.g., cheaper than a given price, closer than a given distance, or faster than a given time. However, while this can be acceptable in some applications, it seems

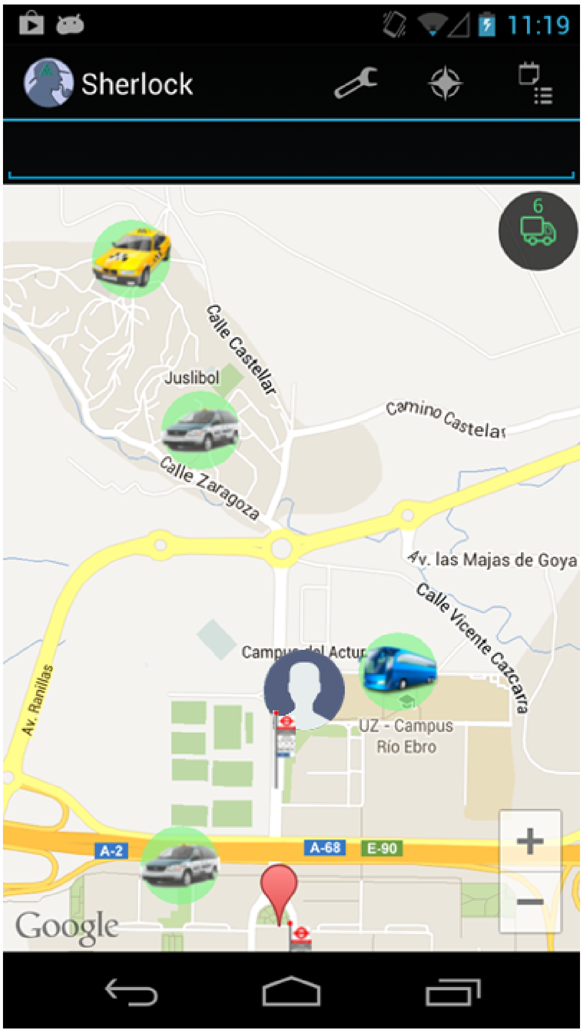


Figure 5: Snapshot of SHERLOCK: recommendation of transportation means

interesting to extend the system with flexible queries, so for instance closeness and user satisfaction with the price are a matter of degree. To do so, a possible solution is to use fuzzy ontologies with fuzzy datatypes representing the possible values of the linguistic variables (such as location or price as fuzzy membership functions. For example, a fuzzy price LessThan10Euro can be defined using a function **left**$(10, 15)$, so that a price of 11 € belongs to LessThan10Euro with degree 0.9. However, providing a definition for those fuzzy membership functions is not a trivial step for a non-expert and domain knowledgeable user.

As an option, we may use Fudge to compute the values of the fuzzy datatypes. We can ask some domain experts, or or other users, to provide their individual definitions for the fuzzy membership functions, and then use any of the discussed linguistic aggregation operators to compute a consensual definition for each fuzzy datatype. For instance, we can assign a weight to each expert/user (for instance, according to his/her professional experience in the field) and use CONV–LRF, CONV–RRF, or WMEAN. We may also determine the vector of weights from a fuzzy quantifier or an orness value and use LOWA–LRF, LOWA–RRF, or FOWA. [7]

7 Conclusions and ideas for future work

In this paper, we have addressed the problem of building fuzzy OWL 2 datatypes via a consensual approach among definitions provided by a group of experts, i.e., to combine the definitions provided by a group of experts into a single one using linguistic aggregation operators. Our proposal supports all fuzzy membership function types that are available in the Fuzzy OWL 2 language (a de-facto standard to represent fuzzy ontologies). Let us recall that, so far, aggregation operators were used in fuzzy ontologies to combine numerical values in $[0, 1]$ only, but in this work we allow to use aggregation of fuzzy membership functions as well. In addition we have also proposed two novel aggregation operators to be used in the process, namely a left recursive form of the convex combination (CONV–LRF) and of the linguistic OWA (LOWA–LRF). An empirical analysis shows that CONV–LRF and the standard convex combination (CONV–RRF) usually coincide, but CONV–LRF has a slightly higher orness degree.

We also proposed to use some existing aggregation strategies: namely, CONV–RRF, standard linguistic OWA (LOWA–RRF), weighted mean (WMEAN) and fuzzy OWA (FOWA). We discussed some features of the aggregation strategies, e.g. the possibility to assign a weight to a specific expert, and the possibility to obtain as an output a value which was not provided by any expert. Furthermore, we show how to obtain the weights from fuzzy quantifiers or an orness value in problems with incomplete data —in which not all experts provide a definition for every fuzzy datatype.

More importantly, our approach is implemented in a publicly available tool called Fudge. We have discussed the main features of Fudge, such as two user interfaces (one for desktop computers and another one for Android devices) and the generation of a consensual OWL 2 ontology from several Fuzzy OWL 2 input files.

We also discuss a concrete use case: how to extend a real ontology-based system providing location-based services to support flexible queries, by aggregation the definitions of some experts to build a consensual fuzzy ontology. In general, any of the existing fuzzy ontology-based application can benefit from our proposal to build them.

Concerning future work, a first idea is to wrap our implementation as a Protégé plug-in [34] in order to better integrate the tool into the (fuzzy) ontology development process. This way, it would be possible to directly create a fuzzy ontology, refine it with the opinion of other experts,

[7] This may also make it possible to evaluate how well the tool may perform in practice, which we leave for further research.

and query the aggregated ontology from Protégé. Right now, these steps can be carried out, but an intermediate step involving Fudge is needed. The plug-in may also good for gathering data from user usage to the end of an experimental validation.

The modular design of the application makes it very easy to consider more fuzzy operators. In particular, we would like to incorporate more general linguistic aggregation operators, e.g., not verifying internality or assuming a vector of non-numerical weights [51]. More general linguistic operators include the linguistic weighted OWA [47] or the fuzzy triangular ordered weighted arithmetic operators (based on a t-norm and a t-conorm) [39]. We might also consider supporting different quantifiers, or alternative rankings between fuzzy numbers.

Last but not least, it would be also possible to study other scenarios in which the experts are allowed to use already existing linguistic terms, possibly different to the labels used by other experts. In this case, we plan to reuse existing work on defining reasoning-preserving mappings between local linguistic terms [1].

Acknowledgment

I. Huitzil was partially funded by Universidad de Zaragoza - Santander Universidades (Ayudas de Movilidad para Latinoamericanos - Estudios de Doctorado). I. Huitzil and F. Bobillo were partially supported by the projects TIN2016-78011-C4-3-R and JIUZ-2018-TEC-02. J. Gómez-Romero was partially supported by Universidad de Granada (P9-2014-ING).

We are particularly grateful to Professor Miguel Delgado for the many things he has taught us along the years and is still teaching us. In particular, he always encourages us to find a good balance between formal theoretical foundations and useful practical implementations, as we tried to do in this contribution, where we successfully found an application of some of his early contributions.

References

References

[1] J. Agustí-Cullell, F. Esteva, P. García, L. Godo, R. López de Mántaras, and C. Sierra. Logical multi-valued logics in modular expert systems. *Journal of Experimental and Theoretical Artificial Intelligence*, 6(3):303–321, 1994.

[2] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2 edition, 2007.

[3] G. Beliakov, H. Bustince, and T. Calvo. *A Practical Guide to Averaging Functions*, volume 329 of *Studies in Fuzziness and Soft Computing*. Springer, 2016.

[4] F. Bobillo, M. Cerami, F. Esteva, À. García-Cerdaña, R. Peñaloza, and U. Straccia. Fuzzy description logics. In P. Cintula, C. Fermüller, and C. Noguera, editors, *Handbook of Mathematical Fuzzy Logic Volume III*, volume 58 of *Studies in Logic, Mathematical Logic and Foundations*, chapter XVI, pages 1105–1181. College Publications, 2015.

[5] F. Bobillo, L. Dranca, and J. Bernad. A fuzzy ontology-based system for gait recognition using Kinect sensor. In *Proceedings of the 11th International Conference on Scalable Uncertainty*

Management (SUM 2017), volume 10564 of *Lecture Notes in Computer Science*, pages 397–404. Springer, October 2017.

[6] F. Bobillo and U. Straccia. Fuzzy ontology representation using OWL 2. *International Journal of Approximate Reasoning*, 52(7):1073–1094, 2011.

[7] F. Bobillo and U. Straccia. Aggregation operators for fuzzy ontologies. *Applied Soft Computing*, 13(9):3816–3830, 2013.

[8] F. Bobillo and U. Straccia. The fuzzy ontology reasoner fuzzyDL. *Knowledge-Based Systems*, 95:12–34, 2016.

[9] S. Calegari and E. Sanchez. Object-fuzzy concept network: An enrichment of ontologies in semantic information retrieval. *Journal of the American Society for Information Science and Technology*, 59(13):2171–2185, 2008.

[10] C. Carlsson, M. Brunelli, and J. Mezei. Decision making with a fuzzy ontology. *Soft Computing*, 16(7):1143–1152, 2012.

[11] S. J. Chen and S. M. Chen. A new method for handling multicriteria fuzzy decision-making problems using FN-IOWA operators. *Cybernetics and Systems*, 34(2):109–137, 2003.

[12] V. Cross and S. Chen. Fuzzy ontologies: State of the art revisited. In *Proceedings of the 37th Conference of the North American Fuzzy Information Processing Society (NAFIPS 2018)*, volume 831 of *Communications in Computer and Information Science*, pages 230–242. Springer, 2018.

[13] B. Cuenca-Grau, I. Horrocks, B. Motik, B. Parsia, P. F. Patel-Schneider, and U. Sattler. OWL 2: The next step for OWL. *Journal of Web Semantics*, 6(4):309–322, 2008.

[14] M. Delgado, F. Herrera, E. Herrera-Viedma, and L. Martínez. Combining numerical and linguistic information in group decision making. *Journal of Information Sciences*, 107:177–194, 1998.

[15] M. Delgado, J. L. Verdegay, and M. A. Vila. On aggregation operations of linguistic labels. *International Journal of Intelligent Systems*, 8(3):351–370, 1993.

[16] N. Díaz-Rodríguez, M. Pegalajar-Cuéllar, J. Lilius, and M. Delgado. A fuzzy ontology for semantic modelling and recognition of human behaviour. *Knowledge-Based Systems*, 66:46–60, 2014.

[17] W. M. Dong and F. S. Wong. Fuzzy weighted averages and implementation of the extension principle. *Fuzzy Sets and Systems*, 21(2):183–199, 1987.

[18] M. Eich, R. Hartanto, S. Kasperski, S. Natarajan, and J. Wollenberg. Towards coordinated multirobot missions for lunar sample collection in an unknown environment. *Journal of Field Robotics*, 31(1):35–74, 2014.

[19] S. El-Sappagh, M. Elmogy, and A. M. Riad. A fuzzy-ontology-oriented case-based reasoning framework for semantic diabetes diagnosis. *Artificial Intelligence in Medicine*, 65(3):179–208, 2015.

[20] J. Gómez-Romero, F. Bobillo, M. Ros, M. Molina-Solana, M. D. Ruiz, and M. J. Martín-Bautista. A fuzzy extension of the semantic building information model. *Automation in Construction*, 57:202–212, 2015.

[21] F. Herrera, E. Herrera-Viedma, and J. L. Verdegay. Direct approach processes in group decision making using linguistic OWA operators. *Fuzzy Sets and Systems*, 79(2):175–190, 1996.

[22] M. Horridge and S. Bechhofer. The OWL API: A Java API for OWL ontologies. *Semantic Web*, 2(1):11–21, 2011.

[23] I. Huitzil, U. Straccia, N. Díaz-Rodríguez, and F. Bobillo. Datil: Learning fuzzy ontology datatypes. In *Proceedings of the 17th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2018), Part II*, volume 854 of *Communications in Computer and Information Science*, pages 100–112. Springer, 2018.

[24] A. J. Kamble and T. Venkatesh. Some results on fuzzy numbers. *Annals of Pure and Applied Mathematics*, 7(2):90–97, 2014.

[25] C.-S. Lee, Z.-W. Jian, and L.-K. Huang. A fuzzy ontology and its application to news summarization. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 35(5):859–880, 2005.

[26] C.-S. Lee, M. H. Wang, and H. Hagras. A type-2 fuzzy ontology and its application to personal diabetic-diet recommendation. *IEEE Transactions on Fuzzy Systems*, 18(2):374–395, 2010.

[27] T. Lukasiewicz and U. Straccia. Managing uncertainty and vagueness in description logics for the semantic web. *Journal of Web Semantics*, 6(4):291–308, 2008.

[28] C. Martínez-Cruz, A. van der Heide, D. Sánchez, and G. Triviño. An approximation to the computational theory of perceptions using ontologies. *Expert Systems with Applications*, 39(10), 2012.

[29] J. A. Morente-Molinera, G. Kou, R. G. Crespo, J. M. Corchado, and E. Herrera-Viedma. Solving multi-criteria group decision making problems under environments with a high number of alternatives using fuzzy ontologies and multi-granular linguistic modelling methods. *Knowledge-Based Systems*, 137:54–64, 2017.

[30] J. A. Morente-Molinera, G. Kou, C. Pang, F. J. Cabrerizo, and E. Herrera-Viedma. An automatic procedure to create fuzzy ontologies from users' opinions using sentiment analysis procedures and multi-granular fuzzy linguistic modelling methods. *Information Sciences*, 476:222–238, 2019.

[31] J. A. Morente-Molinera, I. J. Pérez, M. R. Ureña, and E. Herrera-Viedma. Building and managing fuzzy ontologies with heterogeneous linguistic information. *Knowledge-Based Systems*, 88:154–164, 2015.

[32] J. A. Morente-Molinera, I. J. Pérez, M. R. Ureña, and E. Herrera-Viedma. Creating knowledge databases for storing and sharing people knowledge automatically using group decision making and fuzzy ontologies. *Information Sciences*, 328:418–434, 2016.

[33] J. A. Morente-Molinera, R. Wikström, E. Herrera-Viedma, and C. Carlsson. A linguistic mobile decision support system based on fuzzy ontology to facilitate knowledge mobilization. *Decision Support Systems*, 81:66–75, 2016.

[34] M. A. Musen. The protégé project: a look back and a look forward. *AI Matters*, 1(4):4–12, 2015.

[35] T. D. Noia, M. Mongiello, F. Nocera, and U. Straccia. A fuzzy ontology-based approach for tool-supported decision making in architectural design. *Knowledge and Information Systems*, 58(1):83–112, 2019.

[36] I. J. Pérez, R. Wikström, J. Mezei, C. Carlsson, and E. Herrera-Viedma. A new consensus model for group decision making using fuzzy ontology. *Soft Computing*, 17(9):1617–1627, 2013.

[37] A. Ragone, U. Straccia, F. Bobillo, T. D. Noia, and E. D. Sciascio. Fuzzy bilateral matchmaking in e-marketplaces. In *Proceedings of the 12th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2008), Part III*, volume 5179 of *Lecture Notes in Computer Science*, pages 293–301. Springer-Verlag, 2008.

[38] J. A. Rodger. A fuzzy linguistic ontology payoff method for aerospace real options valuation. *Expert Systems with Applications*, 40(8), 2013.

[39] U. F. Simo and H. Gwét. Fuzzy triangular aggregation operators. *International Journal of Mathematics and Mathematical Sciences*, 2018, Article ID 9209524, 13 pages, 2018.

[40] L. Stefanini, L. Sorini, and M. L. Guerra. Fuzzy numbers and fuzzy arithmetic. In *Handbook of Granular Computing*. Wiley, 2008.

[41] U. Straccia. Multi-criteria decision making in fuzzy description logics: A first step. In *Proceedings of the 13th International Conference on Knowledge-Based & Intelligent Information & Engineering Systems (KES 2009)*, volume 5711 of *Lecture Notes in Artificial Intelligence*, pages 79–87. Springer-Verlag, 2009.

[42] U. Straccia. *Foundations of Fuzzy Logic and Semantic Web Languages*. CRC Studies in Informatics Series. Chapman & Hall, 2013.

[43] U. Straccia, E. Tinelli, S. Colucci, T. D. Noia, and E. D. Sciascio. Semantic-based top-k retrieval for competence management. In *Proceedings of the 18th International Symposium on Foundations of Intelligent Systems (ISMIS 2009)*, volume 5722 of *Lecture Notes in Computer Science*, pages 473–482. Springer, 2009.

[44] U. Straccia and G. Visco. DL-Media: An ontology mediated multimedia information retrieval system. In *Proceedings of the 4th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2008)*, volume 423 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[45] R. Studer, V. R. Benjamins, and D. Fensel. Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, 25(1-2):161–197, 1998.

[46] K. Todorov, C. Hudelot, A. Popescu, and P. Geibel. Fuzzy ontology alignment using background knowledge. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 22(1):75–112, 2014.

[47] V. Torra. The weighted OWA operator. *International Journal of Intelligent Systems*, 12(2):153–166, 1997.

[48] L. Troiano and R. R. Yager. Recursive and iterative OWA operators. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 13(6):579–600, 2005.

[49] P. Vojtáš. $\mathcal{EL}$ description logic with aggregation of user preference concepts. In M. Duží, H. Jaakkola, Y. Kiyoki, and H. Kangassalo, editors, *Information modelling and Knowledge Bases*, volume XVIII, pages 154–165. IOS Press, 2007.

[50] M. Wallace. Ontologies and soft computing in flexible querying. *Control and Cybernetics*, 38(2):481–507, 2009.

[51] Z. Xu. *Linguistic Decision Making: Theory and Methods*. Springer, 2012.

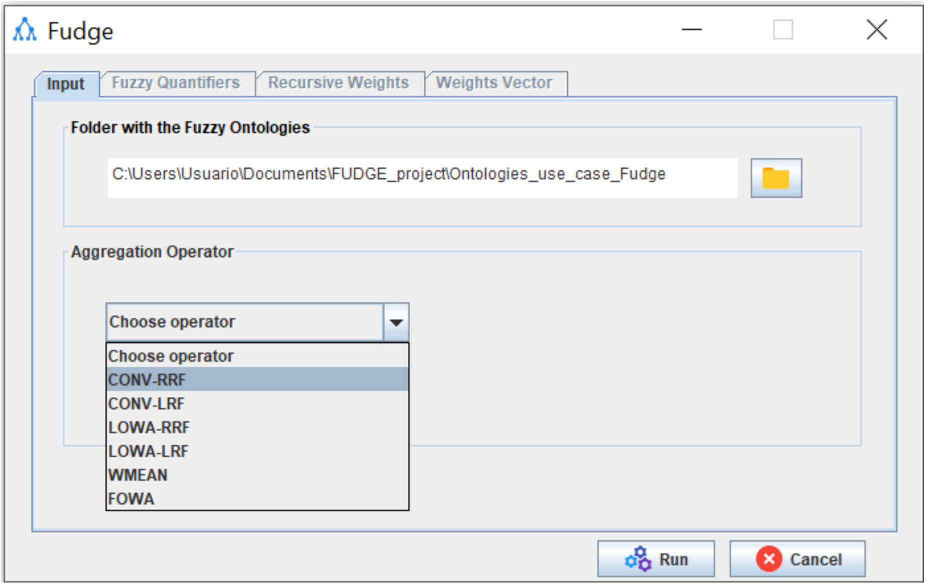
[52] R. R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Transactions on Systems, Man and Cybernetics*, 18(1):183–190, 1988.

[53] R. R. Yager. Connectives and quantifiers in fuzzy sets. *Fuzzy Sets and Systems*, 40(1):39–75, 1991.

[54] R. R. Yager. Quantifier guided aggregation using OWA operators. *International Journal of Intelligent Systems*, 11(1):49–73, 1996.

[55] R. Yus, E. Mena, S. Ilarri, and A. Illarramendi. SHERLOCK: Semantic management of location-based services in wireless environments. *Pervasive and Mobile Computing*, 15:87–99, 2014.

A Snapshots of Fudge



Folder with the Fuzzy Ontologies

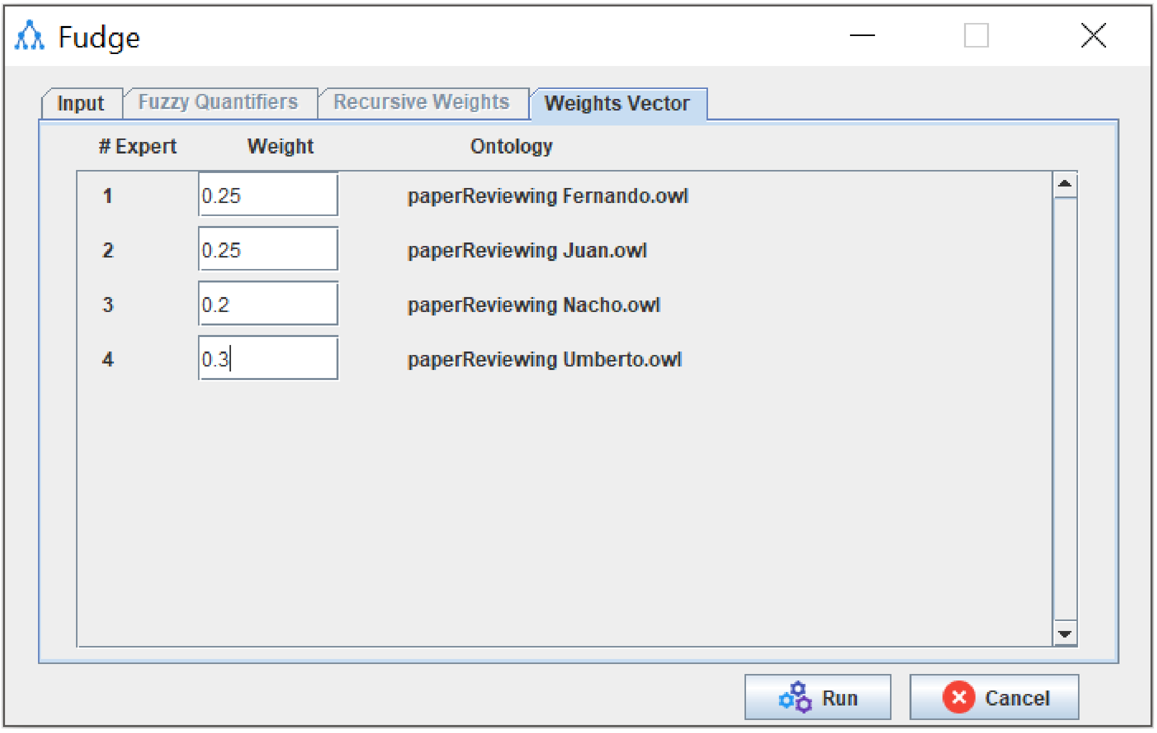
C:\Users\Usuario\Documents\FUDGE_project\Ontologies_use_case_Fudge

Aggregation Operator

Choose operator

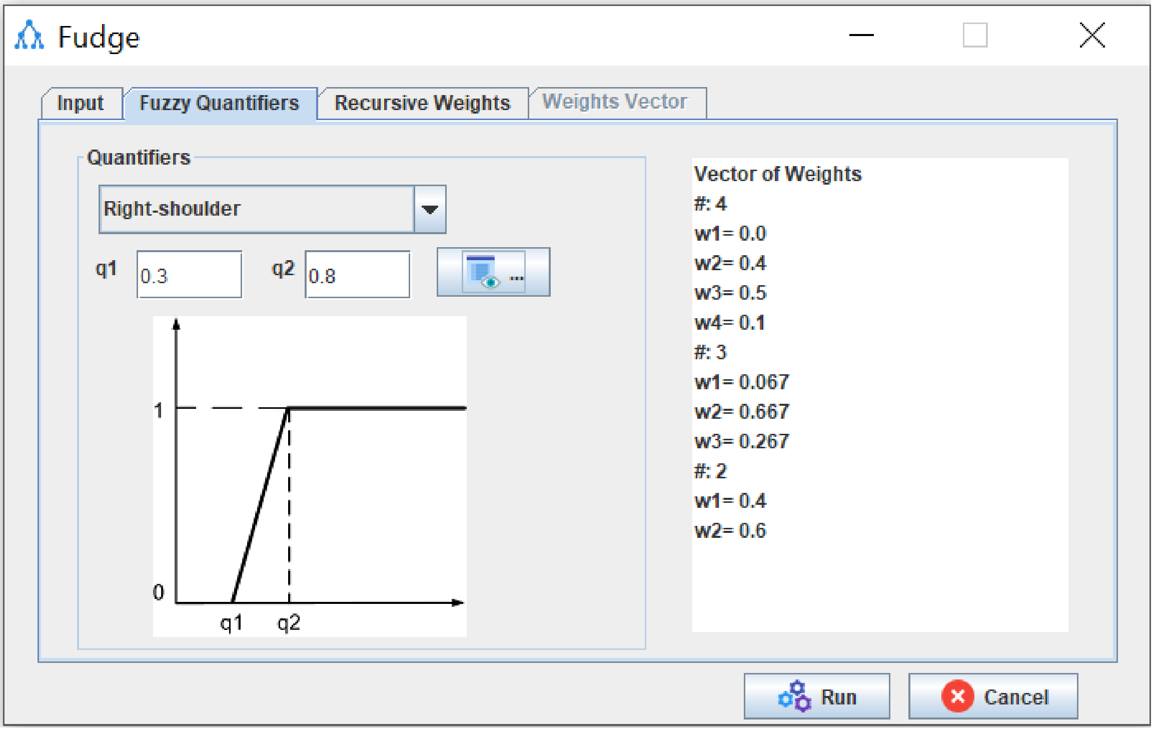
Choose operator
CONV-RRF
CONV-LRF
LOWA-RRF
LOWA-LRF
WMEAN
FOWA

Figure 6: Snapshot of Fudge: selection of input files and aggregation operator



# Expert	Weight	Ontology
1	0.25	paperReviewing Fernando.owl
2	0.25	paperReviewing Juan.owl
3	0.2	paperReviewing Nacho.owl
4	0.3	paperReviewing Umberto.owl

Figure 7: Snapshot of Fudge: vector of weights



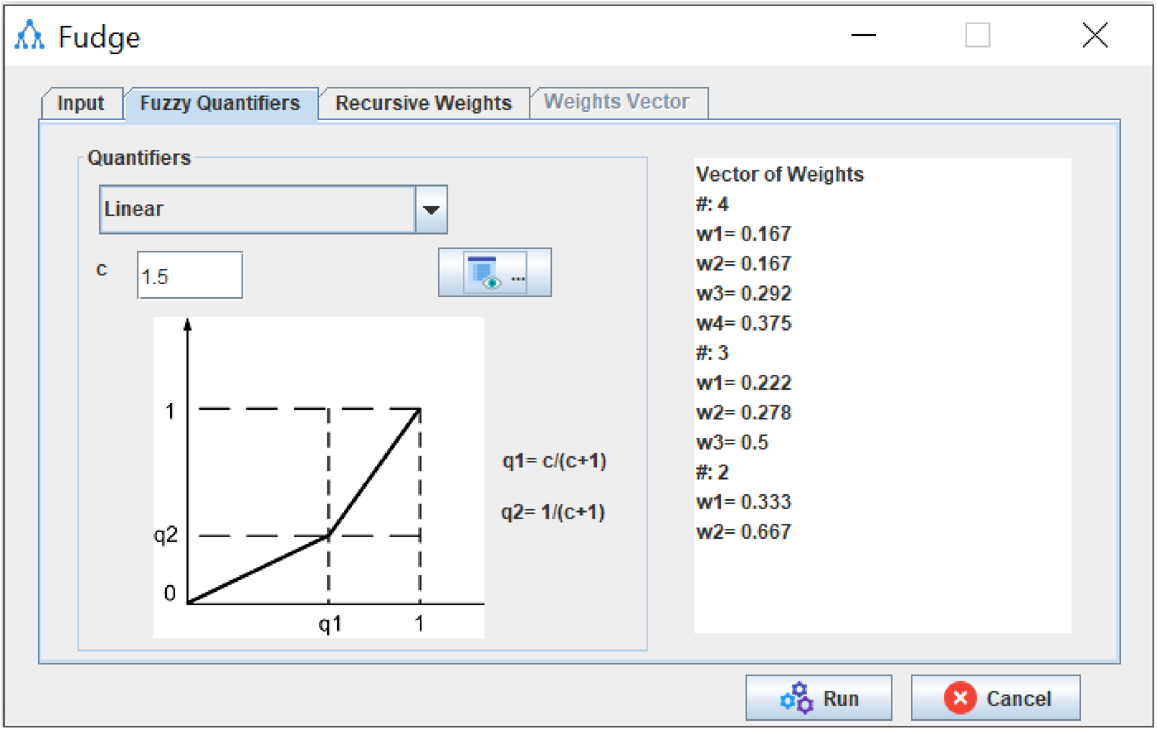
Quantifiers

Right-shoulder

q1 0.3 q2 0.8

```
Vector of Weights
#: 4
w1= 0.0
w2= 0.4
w3= 0.5
w4= 0.1
#: 3
w1= 0.067
w2= 0.667
w3= 0.267
#: 2
w1= 0.4
w2= 0.6
```

Figure 8: Snapshots of Fudge: selection of right-shoulder quantifier



Quantifiers

Linear

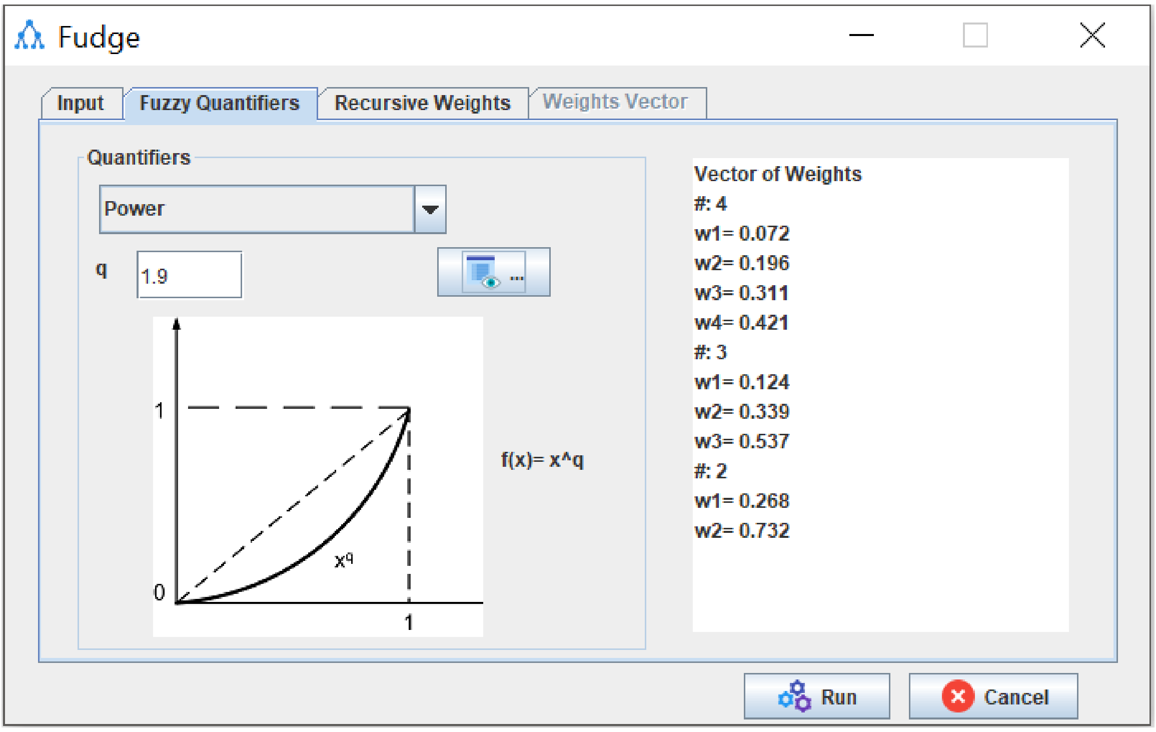
c 1.5

q1= c/(c+1)

q2= 1/(c+1)

```
Vector of Weights
#: 4
w1= 0.167
w2= 0.167
w3= 0.292
w4= 0.375
#: 3
w1= 0.222
w2= 0.278
w3= 0.5
#: 2
w1= 0.333
w2= 0.667
```

Figure 9: Snapshots of Fudge: selection of linear quantifier



Quantifiers

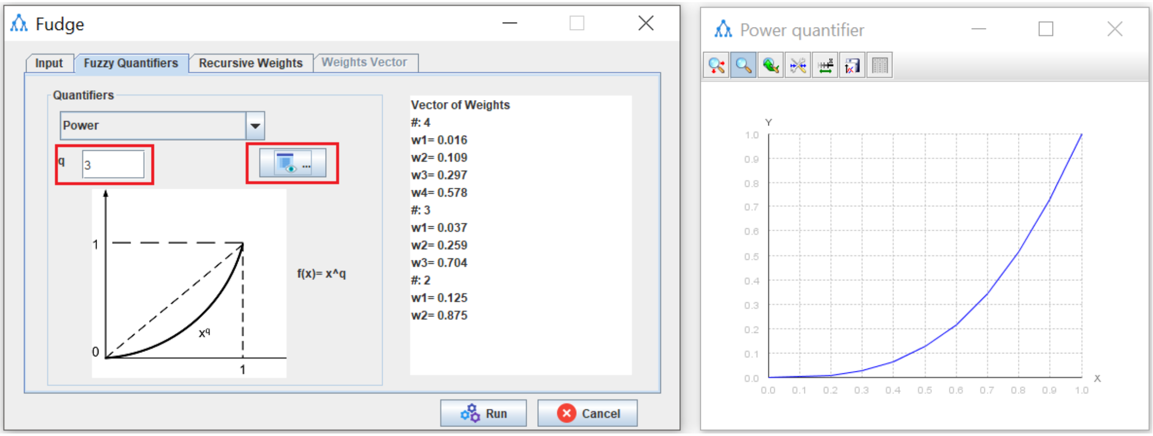
Power

q 1.9

f(x)= x^q

Vector of Weights
#: 4
w1= 0.072
w2= 0.196
w3= 0.311
w4= 0.421
#: 3
w1= 0.124
w2= 0.339
w3= 0.537
#: 2
w1= 0.268
w2= 0.732

Figure 10: Snapshots of Fudge: selection of power quantifier



Quantifiers

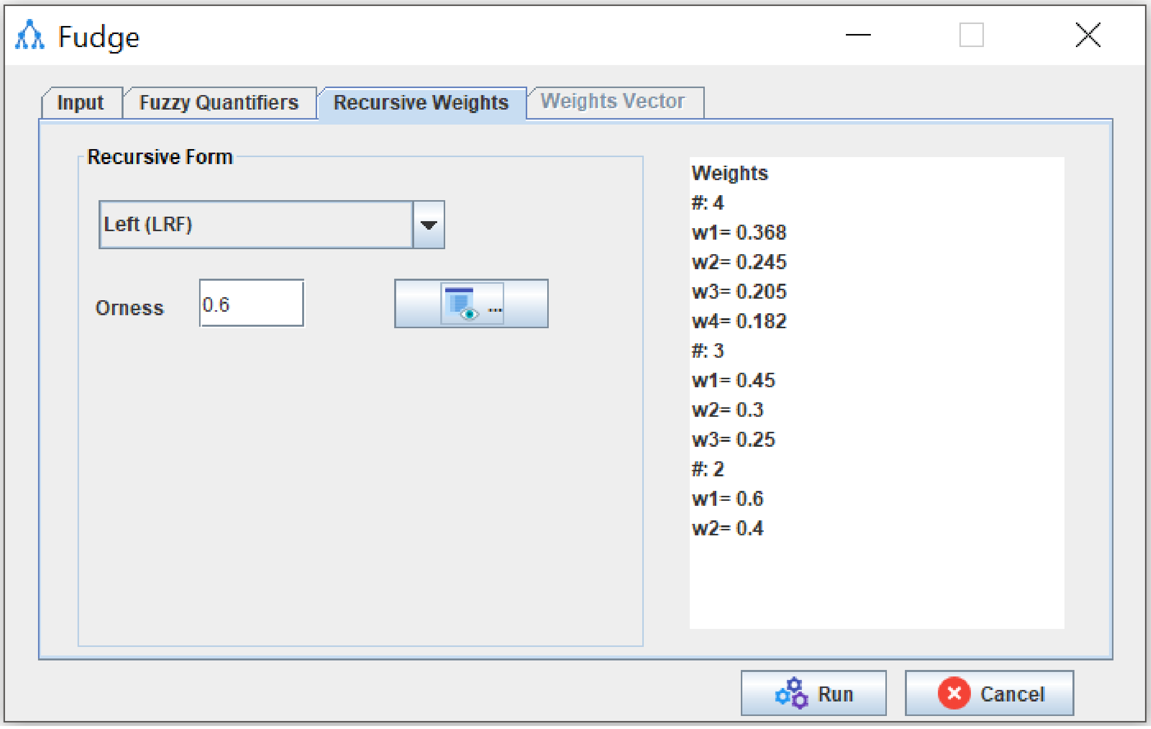
Power

q 3

f(x)= x^q

Vector of Weights
#: 4
w1= 0.016
w2= 0.109
w3= 0.297
w4= 0.578
#: 3
w1= 0.037
w2= 0.259
w3= 0.704
#: 2
w1= 0.125
w2= 0.875

Figure 11: Snapshots of Fudge: customized picture for a power quantifier



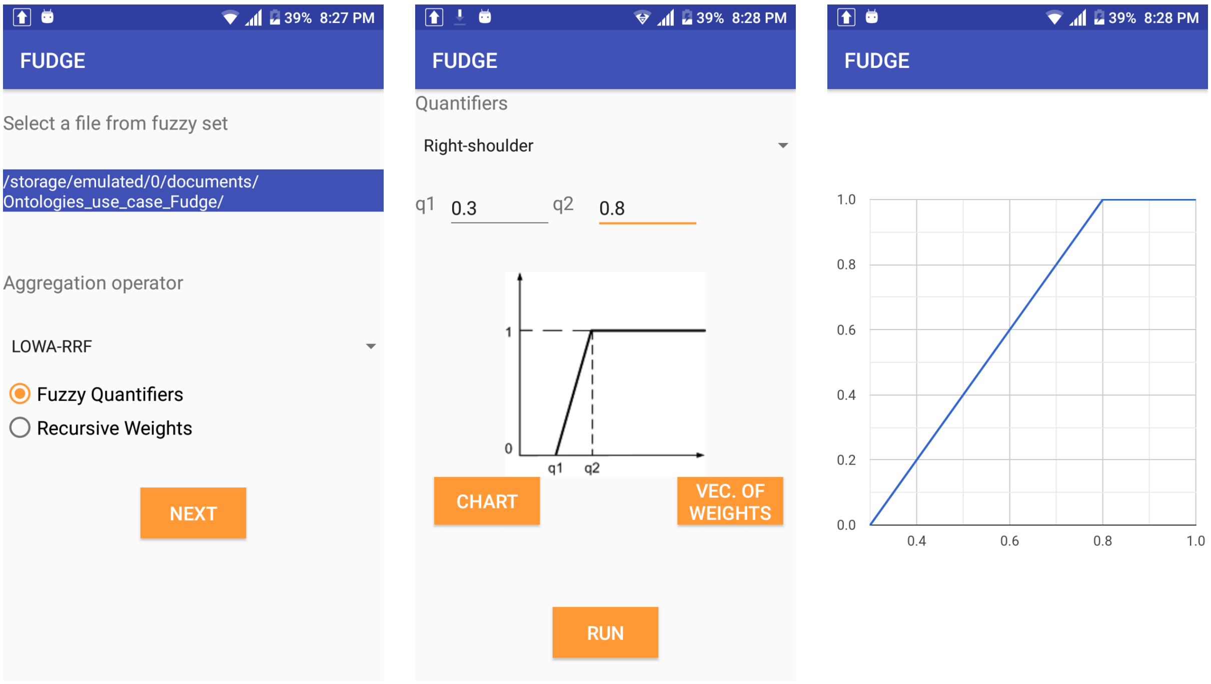
Recursive Form

Left (LRF)

Orness 0.6

```
Weights
#: 4
w1= 0.368
w2= 0.245
w3= 0.205
w4= 0.182
#: 3
w1= 0.45
w2= 0.3
w3= 0.25
#: 2
w1= 0.6
w2= 0.4
```

Figure 12: Snapshot of Fudge: computing the weights from an orness value



Select a file from fuzzy set

/storage/emulated/0/documents/Ontologies_use_case_Fudge/

Aggregation operator

LOWA-RRF

● Fuzzy Quantifiers

○ Recursive Weights

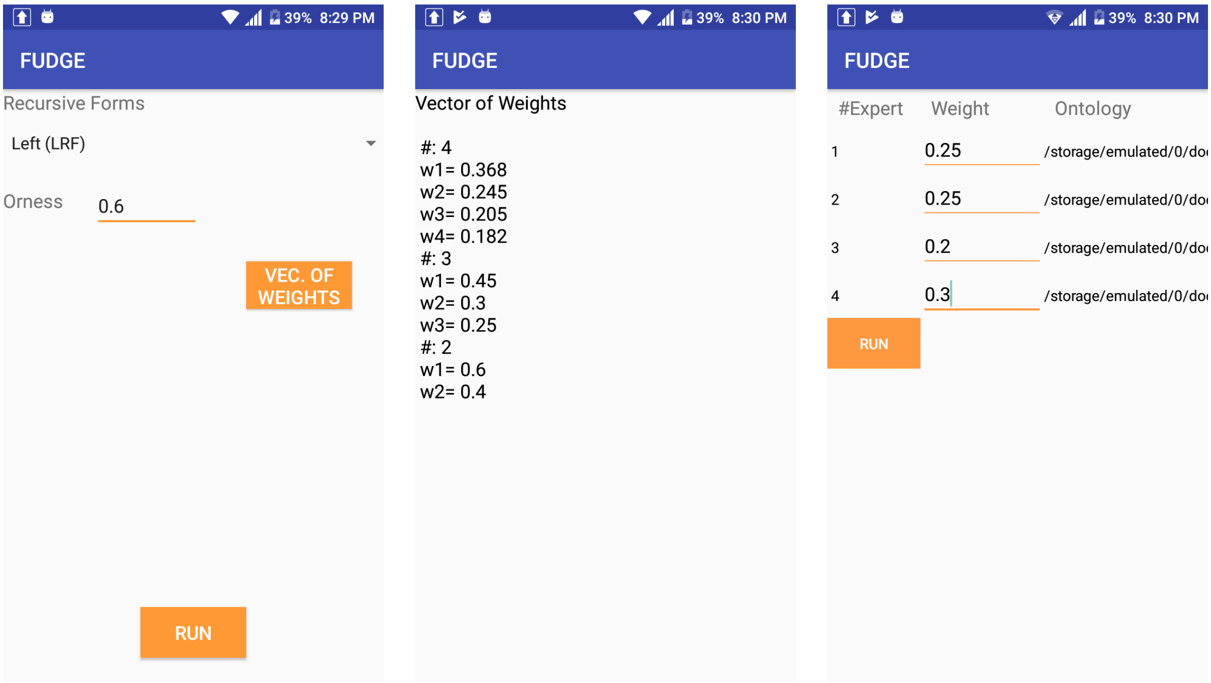
Quantifiers

Right-shoulder

q1 0.3 q2 0.8

a) b) c)

Figure 13: Snapshots of Fudge app: (a) set of input fuzzy ontologies and aggregation operator, (b) quantifiers interface, and (c) customized picture for a right-shoulder quantifier



Recursive Forms

Left (LRF)

Orness 0.6

a)

Vector of Weights

```
#: 4
w1= 0.368
w2= 0.245
w3= 0.205
w4= 0.182
#: 3
w1= 0.45
w2= 0.3
w3= 0.25
#: 2
w1= 0.6
w2= 0.4
```

b)

#Expert	Weight	Ontology
1	0.25	/storage/emulated/0/do
2	0.25	/storage/emulated/0/do
3	0.2	/storage/emulated/0/do
4	0.3	/storage/emulated/0/do

c)

Figure 14: Snapshots of Fudge app: (a) interface for recursive operators, (b) LFR vector of weights, and (c) vector of weights for CONV and WMEAN.