

Article

# Assisting Users in Decisions Using Fuzzy Ontologies: Application in the Wine Market

Juan Antonio Morente-Molinera <sup>1,\*</sup> , Francisco Javier Cabrerizo <sup>1</sup> , Sergio Alonso <sup>1</sup> ,  
Ignacio Javier Pérez <sup>2</sup>  and Enrique Herrera-Viedma <sup>1</sup> 

<sup>1</sup> Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, 18010 Granada, Spain; cabrerizo@decsai.ugr.es (F.J.G.); zerjioi@ugr.es (S.A.); viedma@decsai.ugr.es (E.H.-V.)

<sup>2</sup> Department of Computer Sciences and Engineering, University of Cadiz, 11003 Cadiz, Spain; ignaciojavier.perez@uca.es

\* Correspondence: jamoren@decsai.ugr.es

Received: 29 July 2020; Accepted: 26 September 2020; Published: 7 October 2020



**Abstract:** Nowadays, wine has become a very popular item to purchase. There are a lot of brands and a lot of different types of wines that have different prices and characteristics. Since there is a lot of options, it is easy for buyers to feel lost among the high number of possibilities. Therefore, there is a need for computational tools that help buyers to decide which is the wine that better fits their necessities. In this article, a decision support system built over a fuzzy ontology has been designed for helping people to select a wine. Two different possible architecture implementation designs are presented. Furthermore, imprecise information is used to design a comfortable way of providing information to the system. Users can use this comfortable communication system to express their preferences and provide their opinion about the selected products. Moreover, mechanisms to carry out a constant update of the fuzzy ontology are exposed.

**Keywords:** decision support systems; fuzzy ontologies; computing with words

## 1. Introduction

Wines are a popular item to purchase. Depending on the manufacture, they can have different characteristics. For instance, there are wines with different levels of alcohol, acidity, year, etc. Therefore, finding the perfect wine for a specific buyer is a quite difficult task. There is not a perfect wine for everybody since each person has different tastes and searches for a specific experience. The high number of brands and wines that are available on the market make it difficult for the buyers to select the wine that better fits their necessities. They cannot handle by themselves the high amount of information about all the features, prices and brands. Therefore, there is a need for designing decision support systems (DSS) [1–4] that allow them to choose the wine that better fulfils their needs. This process must be carried out in an organized and fair way. The system should ask the buyers for parameters about what they need and provide them with a short list of wines that fulfil them. This way, buyers can decide which wine they should buy based on objective criteria. Thanks to this, they avoid being misled by the high amount of information. Furthermore, they do not rely on criteria that may make the wrong choice.

In this paper, a novel DSS whose main purpose is to help buyers to choose a wine is designed. Users provide information to the system and they obtain several suggestions that they can use to select their most preferred wine. Fuzzy ontologies (FOs) are used to store all the information in the system. For the DSS to work properly, the set of elements stored on the FO must be constantly updated. For this purpose, two different updating information processes are described. For testing purposes, the method is applied over a real case example. Concretely, the Wine Fuzzy Ontology [5]

that contains real information about 623 different wines is used. Linguistic modelling (LM) [6–8] is employed for storing wines features in the FO and for buyers to provide preferences. LM provides a fluid user-system commutation for the buyers. When an item is requested from the FO, users are usually not capable of providing accurate features values. Instead, they prefer to communicate and express themselves using imprecise words instead of numbers. For instance, if users are interested in a high alcohol wine, they probably would like one with around 15% alcohol. On the contrary, they would not like a wine that only has 5% alcohol. Our system takes into account this issue and allows users to specify every feature value in a non-accurate way avoiding the troublesome task of providing numerical values to the system. Moreover, the novel designed DSS implements a feedback process that allows users to benefit from the opinion of other users that have already tested the device. Furthermore, to implement the system, two different architecture schemes are proposed. In conclusion, a novel DSS system that allows users to choose a wine fairly and impartially is designed in this article. Our method relies only on the wines' features and quality. Thanks to this, the system can be trusted.

It is possible to adapt the designed system to solve other problems. That is, the FO can be adapted to deal with other data related to other decision making problems. For instance, it is possible to build a DSS that help buyers in renting a flat on a location. Thanks to this, the tenant can reduce the high number of possible houses into the ones that really fit their needs. Another possible application of the presented design would be, for instance, in deciding among different films.

The organization of the paper is done in the following way. Section 2 presents several concepts that the novel developed method uses. In Section 3, the DSS is exposed in detail. In Section 4, a brief use example is shown. In Section 5 the method is discussed in detail. The paper ends with conclusions.

## 2. Preliminaries

Some concepts needed to correctly comprehend the method are introduced here. Concretely, in Section 2.1, basis of LM are introduced. In Section 2.2, FOs are exposed.

### 2.1. Group Decision Making and Linguistic Modelling

Linguistic modelling [6,7,9,10] has had a clear impact over Group Decision Making methods over the years. Its main purpose is to allow experts to communicate using words instead of numbers. This way, the communication gap between users and the computational system that manage the decision process is reduced. In our system, LM is used to design a framework that allows users to provide information to the FO reasoner in a comfortable way. Words and expression like Very high, Low, Very low or Medium are employed by them to send their preferences to the system. Since FOs can deal with imprecise information, they can effectively manage the data. Thanks to LM, users do not have to provide accurate numerical information. On the contrary, they can just indicate more or less the importance and the values that each characteristic of the alternative have for them. The relations of the FO also use LM for storing the information.

LM uses linguistic variables that contain different terms. Each term represents a different grade of the variable. Formally, a linguistic variable can be defined as a quintuple  $\langle L, T(L), U, S, M \rangle$ , where  $L$  is the variable name,  $T(L)$  is a set that contains the labels that conform the linguistic variable,  $U$  is the universe of discourse,  $M(X)$  is a subset of  $U$  and  $S$  is a set of rules that associates the set  $X$  with  $M(X)$ .

To express the meaning of a fuzzy set  $M(X)$  from  $U$ , it is possible to define a membership function such that:

$$\mu_{M(X)} : U \rightarrow [0, 1] \quad (1)$$

where  $\mu_{M(X)}(z)$  is membership degree of  $z$ . It should be noticed that  $z \in U$  [11].

In Figure 1, a graphical representation of the linguistic variable *Acidity* is shown. This representation is the one used on the Wine FO.

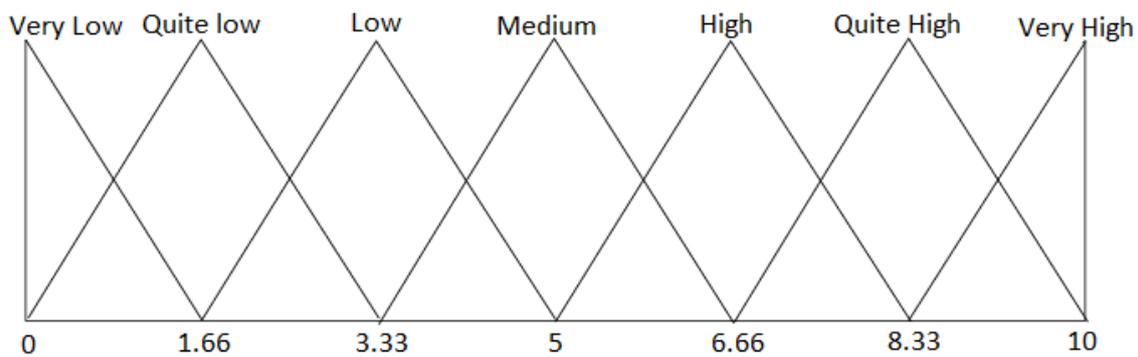


Figure 1. Representation of the linguistic variable *Acidity*.

LM is an area that nowadays is still having a high impact in the recent literature [12–15]. Most of the articles describe application research that employs LM to represent imprecise information. DSS are also a prime example of this as can be stated in [16–20]. Furthermore, this field has had a clear impact over FOs [21–24].

In the Group Decision Making area, it is very important that the experts are provided with means that allow them to provide information in a comfortable way. That is why LM has been of great use on the area. Formally, a Group Decision Making method can be defined as follows:

Let us define two sets, of experts and alternatives respectively,  $E = \{e_1, \dots, e_n\}$  and  $X = \{x_1, \dots, x_m\}$ . The main aim of a group decision making methods is to rank elements from  $X$  using the preferences values  $P^k, \forall k \in [1, n]$ , that has been provided by experts in  $E$ .

A typical Group Decision Making method follows the next steps [2]:

- **Providing preferences:** Experts carry out a thorough debate in which they discuss about the advantages and drawbacks of the alternatives. Afterwards, they provide their preferences to the system.
- **Calculating the collective preference value:** All the preferences provided by the experts are aggregated into a single collective piece of information containing the overall opinion of all the experts.
- **Calculating consensus:** Consensus is a helping feature that allows Group Decision Making processes to be fair and include all experts' points of view on the decision process. The main idea of applying consensus is to promote that experts make their opinion closer in order to reach a final result as consensual as possible. Consensus measures analyze the experts' preferences and indicate how similar two experts's preferences are. This way, it is possible to know which experts have different opinions than the others. Furthermore, it is possible to measure the overall consensus reached. If the consensus is high enough, it is possible to calculate the final ranking of alternatives. On the contrary, experts should carry out more debate in order to bring their opinions closer.
- **Ranking alternatives:** By using the collective preference piece of information, alternatives are ranked. The first alternative on the ranking is considered the most promising one.

In Figure 2, a graphical representation of this process is shown.

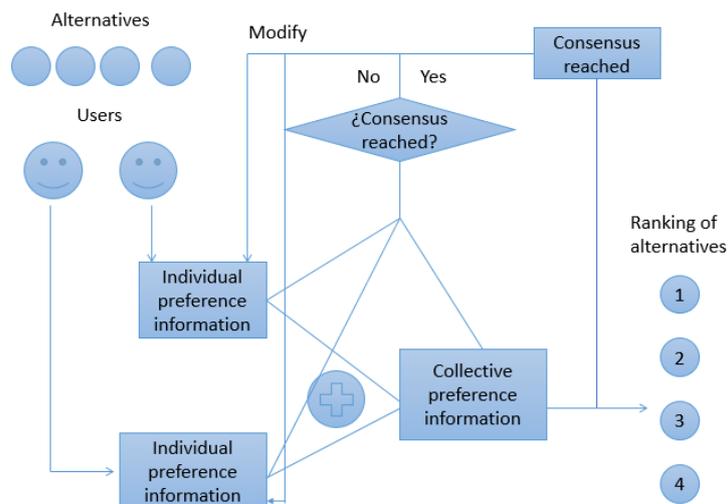


Figure 2. Group Decision Making general scheme.

2.2. Fuzzy Ontologies

Ontologies [25,26] are a very interesting tool that allows the information to be represented in a conceptual way inside a computational system. Nevertheless, they require the information to be accurate and numerical. Consequently, they do not allow data to be represented using LM. For this reason, there is a need to update ontologies in a way that they can work using imprecise information.

FOs [27–30] were designed to achieve this goal. A FO can be formally defined by using a quintuple  $O_F = \{I, C, R, F, A\}$ .  $I$  is the element that represents the set of individuals.  $C$  contains a set of concepts that are used to describe the individuals.  $R$  establishes the relationships between sets  $I$  and  $C$ . Furthermore, elements in  $I$  can be related. These relationships are crisp making them unable to employ LM on them.  $F$  represents fuzzy relationships among elements. In this case, a fuzzy set is employed to relate the different elements. Therefore,  $F$  values are typically used when establishing imprecise relationships among the FO elements. Finally,  $A$  represents the set of axioms.

The described scheme is shown in Figure 3. All the elements of the quintuple that conform the FO are depicted.

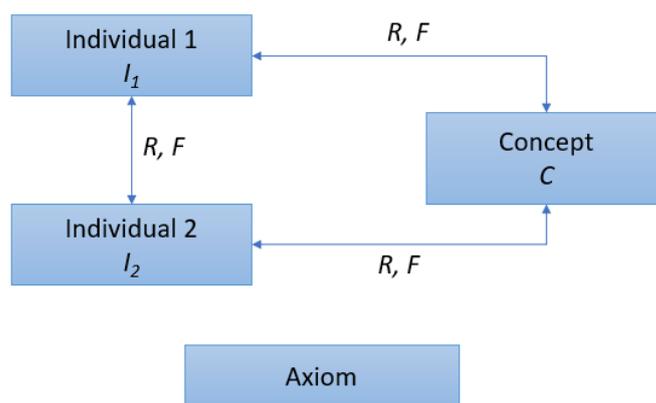


Figure 3. How elements in a FO interact.

There exist some variations of FOs such as the known as Fuzzy Grassroots Ontologies (FGOs) [31]. They employ clustering in order to create groups of similar elements. The further an element is from the center of the cluster, the less similarity it has. Formally, a fuzzy grass ontology is a quintuple  $\langle X, C, T, N, A \rangle$  where  $X$  indicates the set of normalized terms,  $C$  the set of fuzzy clusters,  $N$  the set

of non-taxonomy relations among the clusters and  $A$  a set of axioms. Each cluster is built as a fuzzy set on the dataset  $X$ . Some Fuzzy Grassroots Ontology applications can be seen on [32,33]. In these articles, Fuzzy Grassroots Ontologies are used for extracting information about weblogs and for improving social semantic web search respectively.

Before ending the subsection, some papers that are part of the recent research carried out in the FO area are shown. For example, Huitzil et al. [21] employ FOs and Kinect sensor data to recognize gait. In [34], authors use FOs to manage fuzzy roles in risk-based environments. In [35], a DSS that uses a fuzzy OWL 2 ontology is presented. In [36], description logics are used in order to define the multi-criteria group decision making problem. In [37], FOs are used to define the necessary semantic for controlling smart homes using the internet of Things. In [38], Sumathi et al. propose an ontology that stores information about an information processing technique for query recommendation applications. It can be concluded that FO is a quite recent field that is evolving and changing due to the research made on the area. In [39], authors present some algorithms for information retrieval and realization in fuzzy ontologies. In [23], authors review several constructions referring to type-2 fuzzy ontologies. That is, fuzzy ontologies that employ type-2 fuzzy sets in order to represent their elements.

### 3. A Novel Decision Support System for Assisting Users in the Selection of a Wine

The novel designed DSS is thoroughly described in this section. Concretely, in Section 3.1, the wine FO is described in detail. In Section 3.2, the DSS procedure for generating recommendations is exposed. In Section 3.3, two possible architectures that can be used to carry out the DSS implementation are analysed. In Section 3.4, the users' feedback process is exposed. Finally, in Section 3.5, two possible ways of carrying out the updating information task of the FO are exposed.

#### 3.1. The Wines' Fuzzy Ontology

All the data that are used to support the users in their decision is stored in a FO. Thanks to it, it is possible to have an overview of the wine that are present in the market in a concrete time. To provide up to date information to buyers, it is necessary to constantly update the FO. In this paper, two different ways of carrying out this update process are presented. The first one is to let wine experts to do it without any computational intervention. This option is not a good one in cases like the tackled one where there is a high amount of information available. On the other hand, it is possible to design an automatic process that analyses certain Webpages to extract information about the new wines that keep appearing in the shops. This computational process would be in charge of deleting wines that are no longer available and include the new ones that appear on the market. The complete updating process is thoroughly described in Section 3.5.

For testing purposes, a real case example has been chosen: the Wine FO. Wine FO contains 623 individuals related to six different concepts. The fuzzy ontology is built in a way that each individual of  $I$  represents one wine and each concept of  $C$  represents a wine characteristic. The set of concepts that appear in the FO have been obtained by studying which are the features that wine buyers care the most. Finally,  $F$  and  $R$  indicates how each wine on  $I$  is related to the features in  $C$ . The relations related information have been obtained from different famous webpages (alko.fi, winesfromspain.com, smooth.com). The used FO can be accessed and downloaded on the following webpage [5]. The elements that are part of the wine FO set  $C$  are described below:

- **Year:** This feature indicates the wine year. Older wines have different tastes than recent ones. Therefore, it is important to take into account the wine year on the reasoning process. The associated labels for this concept include wines from 1999 in advance. A linguistic label set of four different labels,  $\{NovelloYear, RegularYear, OldYear, ExclusiveYear\}$  is used.
- **Acidity:** It indicates the level of acidity of the wine. Acid wines have different properties than sweet ones. The concept is defined in the interval  $[0, 10]$ . A linguistic label set of seven different

labels,  $S^7 = \{s_1, \dots, s_7\}$  is used. A graphical representation of how the labels are distributed on the scale can be seen in Figure 1.

- **Alcohol:** This feature indicates the level of alcohol that the wine has. The level of alcohol has a high impact on the wine final taste. Values related to this feature are located on the interval  $[0, 20]$ . A linguistic label set of seven different labels,  $S^7 = \{s_1, \dots, s_7\}$  is used. The same distribution than the one used for acidity is employed. Concretely, values  $\{3.33, 6.66, 10, 13.33, 16.66\}$  are used for defining the support of the fuzzy sets used for representing the labels.
- **Price:** Indicates how much the wine costs. This concept main purpose is to provide buyers with options whose price matches with the amount of money that the buyers want to spend. The range of the values of this concept are located on the interval  $[6.5, 23]$ . In order to represent this concept, a linguistic label set of three different labels,  $S^3 = \{s_1, s_2, s_3\}$  is used. Fuzzy sets used for representing  $S^3$  labels are shown in Figure 4.
- **Wine color:** Indicates the color of the wine. This concept is related to the individuals by using a crisp relation instead of a fuzzy one. There are three different values for this option: red, white and rose.
- **Other users' opinions:** This fuzzy concept stores the overall experience that other users have had with the wine. This value is in constant update since it summarizes the values obtained on the feedback process. The range of values used in defining the linguistic labels is  $[0, 10]$ , being 10 the maximum punctuation that can be given to a wine and 0 the worst one. More information about its representation and how the feedback process is performed can be seen on Section 3.4.

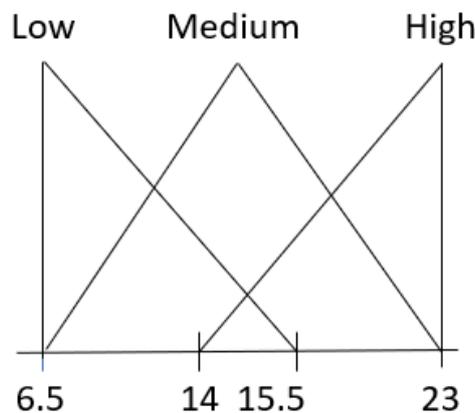


Figure 4. Representation of the linguistic variable *Price* in the Wine FO.

### 3.2. Decision Support System Design

This subsection describes how the designed DSS for choosing wines works. Let  $X = \{x_1, \dots, x_n\}$  be the overall set of alternatives and  $C = \{c_1, \dots, c_m\}$  the overall set of criteria. The main purpose of the process is to obtain a ranking set,  $R = \{r_1, \dots, r_o\}$  that includes the  $o$  best options among the set  $X$ . It should be noticed that  $o < n$ . In order to carry out this process, the following steps are followed:

1. **Preferences providing step:** Buyers provide their preferences according to the features that they want the selected wine to have. They can choose which features they want to provide information for and which labels they want to associate to them. Thanks to this, only the features that matter to the buyers are used on the search. Buyers express themselves using LM. They can choose labels from the linguistic label sets that conform the FO to provide information to the system. Let  $S = \{S^{c_1}, \dots, S^{c_m}\}$  be the linguistic label sets used for describing the criteria specified in  $C$ . In case that a specific criterion is crisp instead of fuzzy, the possible values that the relation can

have are listed. In the preference providing step, buyers provide the sets  $W$ ,  $QC$  and  $QS$  to the system where:

$$W = \{w_1, \dots, w_{m_1}\} \tag{2}$$

$$QC = \{qc_1, \dots, qc_{m_1}\} \tag{3}$$

$$QS = \{s_i^{qc_1}, \dots, s_j^{qc_{m_1}}\} \tag{4}$$

where  $m_1$  is the number of features selected by the buyers,  $QC$  is the set of features that have been selected and  $s_i^{qc_1}$  indicates the label whose index is  $i$  inside the linguistic label set  $S^{qc_1}$ . Finally,  $W$  is the set of weights.

2. **FO search:** The preferences that the buyers have sent to the system are used for retrieving the individuals that better fulfil them. To carry out this task, the following steps are used:

- Buyers determine the importance that each feature has to them by providing the set of weights  $W$ . If buyers do not want to provide this information or they feel unsure about it, it is possible to assign the same importance to all the features included on the query. Weights have an important role in the presented DSS. This is because they are the tools that buyers employ in order to determine the importance that should be given to each FO concept. In the Wine FO, the represented information is subjective and more related to tastes. Therefore, there not exist objective good solutions. Nevertheless, there can be other applications where this statement is not fulfilled. For instance, imagine a FO filled with information about smartphones and their characteristics. It is clear that everyone would want a smartphone with a high screen resolution, fast and cheap. In this case, the importance given to each concept will determine the set of smartphones that the buyers will receive. For instance, if the price has the highest weight on the query, cheap smartphones will be highlighted over the ones that have, for instance, a high screen resolution. Concretely, the query will end up returning cheap smartphones whose screen resolution is the highest possible one. This behavior is interesting in DSS environments that have a high number of alternatives. This is because the FO will focus on pareto-optimal smartphones, that is, the smartphones that, by fulfilling all the desired characteristics, have higher values for all the mentioned features. Smartphones that are worse in every sense to this pareto-optimal set, will be discarded.
- The weighting values along with the preferences conform the user query. This query is taken by the FO reasoner to process it and generate a result. The query can be formally represented as:

$$Q = \{w_1 \cdot [qc_1, s_i^{qc_1}], \dots, w_k \cdot [qc_k, s_j^{qc_k}], \dots, w_{m_1} \cdot [qc_{m_1}, s_j^{qc_{m_1}}]\}, k = 1 \dots, m_1. \tag{5}$$

- The FO reasoner uses an aggregation operator to calculate the similarity that each individual of the FO has with the query. Depending on the type of relation, the similarity of the individual according to one of the concepts included in the query is calculated as follows:
  - **Crisp relation:** Crisp relations are employed for binary relations, that is, the concept is fulfilled or not by the FO individual. If the query includes one of these concepts, the similarity is 1 if the individual fulfils the concept included on the query or 0 otherwise.
  - **Fuzzy relation:** In the case of fuzzy relations, the similarity is calculated as a number in the interval  $[0, 1]$ . The similarity value of the FO individual is calculated using the membership function to the fuzzy set that represents the linguistic label specified for the concept included on the query.

To calculate the similarity that alternative  $x_h$  has to the query  $Q$  and the feature  $Cq_k$ , the following expression can be used:

$$SimC(x_h, Cq_k, Q) = \mu\{x_h, s_l^{q_{c_k}}\}, \tag{6}$$

where  $\mu\{x_h, s_l^{q_{c_k}}\}$  indicates the membership value that alternative  $x_h$  has for the label  $s_l^{q_{c_k}}$  and the concept  $q_{c_k}$ . By using the weighted mean operator, it is possible to calculate the similarity value for  $x_h$  and the query  $Q$  using the following expression:

$$Sim(x_h, Q) = \phi(w_k, SimC(x_h, Cq_k, Q)), k = 1, \dots, m_1. \tag{7}$$

Once that the similarity value for each alternative is calculated, it is possible to select the  $o$  closer elements to the query. For this purpose, the following expression can be used:

$$R = \bigcap_o rank(Sim(x_h, Q)), h = 1, \dots, n, \tag{8}$$

where  $rank$  is a function that sort all the results according to their similarity value.  $\bigcap_o$  takes the first  $o$  elements of the set.

3. **Feedback advice:** The users opinion system is called for the recommended wines. If one of the resulting wines have a low opinion value and a reasonable number of provided opinions, this issue is showed to the user as a warning. This way, users are alerted that the recommended wine did not fulfil other users expectations.
4. **Providing opinion:** After buying and testing the selected wine, the user is asked to carry out the feedback process. This process is exposed in more detail in Section 3.4. Its main purpose is to warn other users about problems with the selected choices.

In Figure 5, this process is presented schematically.

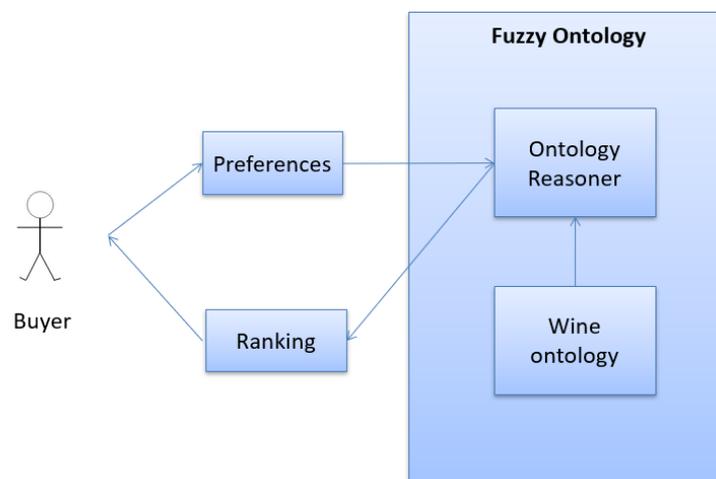


Figure 5. Designed DSS scheme.

### 3.3. Architecture Scheme

Two possible architecture scheme options for implementing the process described in Section 3.2 were analysed: a client-server architecture and a distributed architecture. Both of them have their own advantages and drawbacks. Therefore, implementers should choose one or the other according to their needs.

The client-server architecture design has the following advantages:

- All the information is concentrated in a single point. This way, carrying out changes to the stored information is easy since only the server values have to be modified. In cases when the FO is in constant update, this is a really important point that should be taken into account.
- All the calculations are carried out in the server. This way, users do not have to carry out any computations or install anything in their client devices. Easing the way that users interact and manage the designed system is critical if we want to encourage them to use it.

Nevertheless, the use of a client-server architecture presents the following disadvantages:

- Since all the information is concentrated in the same point, if the server fails or a connection error occurs between the server and the client devices, the system will stop working. Consequently, users will be unable to use the system until the error is fixed.
- Since a single server holds all the computations, if a very high number of users access the system at the same time, the server may collapse.

The main advantages of using a distributed architecture for carrying out the process are exposed below:

- Distributed architectures are a good option when a high number of accesses are expected to the system. Due to the fact that the computational effort is divided among the different nodes, the system is rarely overcharged.
- When there are several requests at the same time, the overall system response time is increased since several user petitions can be carried out at the same time.
- One of the most important advantages of distributed architectures is the scalability that they provide. This way, it is easy to add and remove nodes from the computational network as needed. Furthermore, information from the FO becomes scalable since new information storing nodes can be added with new FO information in an easy way.
- If the information is replicated in each information node, the system is more fault-tolerant. This is due to the fact that, if one of the network nodes fail, the system can rely on the other nodes to continue the operations as usual.

The main drawbacks of using a distributed architecture are exposed below:

- Maintaining a computational network is more expensive than maintaining a single server holding all the computations. If a very high number of users is not expected, to use a distributed architecture may not be worth it due to the expenses that it entails.
- The overall computation of a single request is increased since time is lost during the network nodes communication. Therefore, using a distributed architecture can slow down the overall request resolving time when a low number of requests are being resolved by the system.
- When the stored information is replicated in different nodes, making changes to the information is costly since all the nodes that stores the data must be modified. On the other hand, if the whole information is distributed in different nodes, this problem is solved but the overall network becomes less fault-tolerant. This is due to the fact that, for every request, all the information must be accessed. Therefore, if one of the information storing nodes fails, all the user client requests cannot be resolved.

#### 3.4. Feedback Process

In order to increase the reliability of the designed DSS, an user feedback process has been added to the system. Its main purpose is to aid users who are unsure about what wine they should choose. Thanks to this module, they can get benefit from the experience that other users that own their chosen wines had. The idea consists of allowing the users to provide an opinion about the wine after buying and trying it. The feedback process follows the next steps:

1. The users, using their client devices, indicate that they want to provide their opinions to the DSS. They should carry out this process after choosing the wine that they prefer and having some time to test it.
2. In order to make this process as simple as possible and for avoiding users to lose a high amount of time, they only have to answer a single question. That question is: did the wine fulfil your expectations? The chosen wine will be recommended to people with similar tastes as the user who has already bought it. Therefore, the answer to this question will confirm if the DSS has done a good job recommending that specific wine according to that user type.
3. The user provides their opinion using the linguistic label set  $B = \{\text{verylow, low, medium, high, veryhigh}\}$ . This way, the users can provide their opinion using words instead of numbers.
4. The users opinions are dealt as another wine feature inside the FO. Therefore, the provided value is aggregated to the previously stored ones and assigned as the relation value for the individual representing the wine. For this purpose, it is handy to store the actual number of users that have already provided their opinion for each wine. Having this value stored as a wine feature allows users to include it in their FO queries.

A scheme of this process along with a small example is showed in Figure 6.

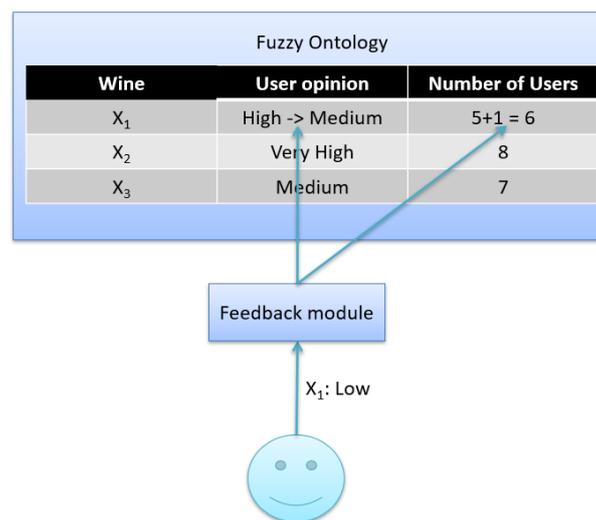


Figure 6. Feedback process example.

### 3.5. Updating Wine Data Procedure

The wine market is in a constant update. Wine companies keep creating and presenting new labels and options. Therefore, in order for our DSS to be useful, information stored in the FO must be periodically updated. In this subsection, two different updating information methods are proposed. The first one takes automatically the information from the internet. In the second one, the updating process is made manually by a wine expert. These two processes are described in more detail below:

- **Automatic updating information process:** The next steps are followed:
  1. The system searches for new wines on the internet using a set of predefined web pages. It should be noticed that, in order to ensure that the information is reliable, it is necessary that the web pages that are used are reliable and recognized.
  2. For each web page, the system uses a text analyser script in order to automatically retrieve the wines information that is specified there. For each web page, the text analyser knows where each of the required piece of data. For each web page, the text analyser knows where each of the data is specified, which will then be used to establish the relationships between

the individuals and the concepts. Once the information is collected, the text is automatically analysed to determine if the values obtained are correct. In order to make the quality of the information more reliable, it is possible to generate a report with the data obtained from the internet and make an expert to revise it.

3. For each wine, the system checks if it is already loaded in the FO. If the system cannot find it, the wine and all its features are added to the FO.
4. When all the specified web pages and their wines are analysed, the automatic updating information process finishes. This process will automatically start again when a predefined amount of time has passed.

The main advantage of this updating method is that all the operations are performed automatically. Therefore, there is no need for human supervision. Its main disadvantage is that, due to the fact that no supervision is made, there is some chance of errors occurring during the process.

- **Manual updating information process:** The manual updating information process is carried out by a wine expert that introduces the new individuals into the FO. For each individual, they manually specifies the relationships values for each of the FO concepts. The main advantage of this method is that all the data is validated and checked by a human expert. Its main disadvantage is that this process can become long and tedious due to the amount of information that the expert has to deal with. It is important that the expert uses information that comes from reliable sources.

It should be pointed out that both processes are non-exclusive. Therefore, the automatic process can be used to carry out periodic updates while the manual one can be performed in order to refine the automatic process and for finding and delete old wines that are no longer available in the market. A Strength-Weaknesses-Opportunities-Threats (SWOT) analysis of both processes can be seen on Tables 1 and 2.

**Table 1.** SWOT analysis for the automatic approach.

<b>Strenghts</b>	It does not require human intervention making it a fast process that can deal with high amount of information.
<b>Weaknesses</b>	The information is less reliable since there is not human checking on the process. The information must be extracted from a predefined set of webpages since the method needs to know where the information is located.
<b>Opportunities</b>	It can work well for FOs that represent a high amount of information.
<b>Threats</b>	Its use may introduce false data on the FO if the information is not properly analyzed or the webpages are incorrect or fail.

**Table 2.** SWOT analysis for the manual approach.

<b>Strenghts</b>	All the information is checked by the person who introduces it. Therefore, the information is very reliable. There is no limits of where the information may come from. It is up to the expert to check the sources.
<b>Weaknesses</b>	Requires human intervention since the information must be extracted and introduced manually.
<b>Opportunities</b>	This is a good option in cases where the information that is represented on the FO must be reliable and checked before their use.
<b>Threats</b>	This is not a good option in cases where there is high amounts of information to deal with.

#### 4. Illustrative Example

This section presents a brief use example to enhance the comprehension of the presented process. Imagine that some buyers want to search for a wine. They want to debate options that have certain characteristics. First, they need to decide which features they matter the most. After debating, they decide to make the query using the following information:

- *alcohol*: high.
- *acidity*: low.
- *price*: medium.

Linguistic label sets exposed in Section 3.1 are used for describing the features. As commented, only the features indicated by the buyers are used on the query. The rest are discarded and do not influence the selection of the final alternatives.

After providing values for the features, buyers need to decide the importance that each of them must have on the decision process. For this purpose, they decide to use the following set of weights:

$$w = \{0.43, 0.33, 0.23\}. \tag{9}$$

This selection indicates that alcohol is more relevant than acidity. Furthermore, acidity should have more weight on the decision than price.

Once that the query is formulated, it is resolved by the FO reasoner. Formally, the query,  $Q$ , can be expressed as follows:

$$Q = \{0.43 \cdot alcohol\_high, 0.33 \cdot acidity\_low, 0.23 \cdot price\_medium\}.$$

By using the provided query, similarity values for all the wines on the FO is calculated. For the sake of simplicity and a better understanding of the designed method, only the 10 most similar alternatives are shown on this example. Table 3 depicts their features' values.

**Table 3.** Relation values for the individuals and concepts in the wine FO.

idWine	Name	HA	LA	MP	User Opinions
172	GerardBertrandReserveSpecialePinotNGR	0.900901	0.802395	0.23	9.1
528	GerardBertrandB.BonCabernetMerlot	0.900901	0.838323	0.49875	8.3
103	GewurztraminerHugel	0.900901	0.718563	0.530588	8.8
295	TurningLeafChardonnay	0.948949	0.419162	0.49875	9.2
230	RosiereSyrah	0.900901	0.598802	0.31125	8
389	MarquesdeCaceresCrianza	0.900901	0.179641	0.87	9
101	Regolo	0.948949	0	0.99875	9.6
127	BeringerFoundersEstateMerlot	0.948949	0	0.99875	9.7
191	ColumbiaCrestTwoVinesCabernetSauvig.	0.948949	0.239521	0.6225	8.8
554	LaDamedeJacquesCoeur	0.900901	0.0598802	0.967059	8.7

In order to calculate the similarity values, the weighted arithmetic mean operator is used. Results are shown in Table 4.

**Table 4.** Obtaining the similarity value for each item in the FO.

Wine id	Operations	Similarity
172	$0.43 \times 0.900901 + 0.33 \times 0.802395 + 0.23 \times 0.6225$	0.7953528
528	$0.43 \times 0.900901 + 0.33 \times 0.838323 + 0.23 \times 0.49875$	0.77874655
103	$0.43 \times 0.900901 + 0.33 \times 0.718563 + 0.23 \times 0.530588$	0.74654853
295	$0.43 \times 0.948949 + 0.33 \times 0.419162 + 0.23 \times 0.49875$	0.661084
230	$0.43 \times 0.900901 + 0.33 \times 0.598802 + 0.23 \times 0.31125$	0.6565796
389	$0.43 \times 0.900901 + 0.33 \times 0.179641 + 0.23 \times 0.87$	0.646769
101	$0.43 \times 0.948949 + 0.33 \times 0 + 0.23 \times 0.99875$	0.6377606
127	$0.43 \times 0.948949 + 0.33 \times 0 + 0.23 \times 0.99875$	0.6377606
191	$0.43 \times 0.948949 + 0.33 \times 0.239521 + 0.23 \times 0.6225$	0.630265
554	$0.43 \times 0.900901 + 0.33 \times 0.0598802 + 0.23 \times 0.967059$	0.6295715

After calculating the similarity value for all the wines, the following wines occupy the top 5 in the ranking:

$$R = x_{172} \succ x_{528} \succ x_{103} \succ x_{295} \succ x_{230}.$$

It is possible to observe that  $x_{172}$  is the wine whose features values are more similar to the specified ones. Although  $x_{528}$  has reached good results on the desired values for alcohol and acidity, it has a lower value for the price. Since alcohol has a high weight on the decision, that is why most of the results show a high similarity in this aspect. Since all the wines that are included on the ranking have high user opinion values, no warning is shown to the buyers.

### 5. Discussion

This paper employs a FO to aid users in decisions that are made over a high number of alternatives. The FO query search can reduce the high initial amount of alternatives into a feasible set of promising alternatives that the buyers can carefully analyze. Therefore, in the designed method, there is no pair-to-pair comparison like in other decision approaches [40,41]. Instead of that, similarity values between the buyers necessities and the wines features are calculated. The main advantage of this approach is that the buyers only need to provide information about the wines' features. Therefore, they do not have to directly deal with the high number of alternatives that are available on the system. It should be noticed that making a pairwise comparison among a high number of alternatives could be a long and complex process that can turn unaffordable for the buyers. Moreover, they have access to other users opinions about the selected wines to make the decision.

Thanks to LM, the way that the users provide their needs to the system becomes more comfortable for them. Users are more accustomed to working with conceptual and imprecise information than with numerical and precise data. Therefore, they are more comfortable if they deal with a system that uses words instead of accurate numbers. The designed DSS can work with imprecise opinions and concepts and, at the same time, can perfectly assist the users in their decisions. In the world of opinions, users usually tend to provide inexact information. Therefore, there is a need for methods that are capable of dealing with that kind information and provide the required results to the users. If users are asked to provide accurate information and they do not know how to do it, they will provide false information that will spoil the whole DSS process.

Since the wine market constantly grows, there is a need for methods that carry out constant updates in the wines FO. Two different methods are proposed, one that is automatic and does not need human supervision and another one that relies totally on humans to carry out the process. The first one is faster since the process is carried out automatically running scripts. Nevertheless, since there is no human supervision, it is less error-tolerant. On the other hand, the human supervised method

can become tedious if a high amount of information must be added. Nevertheless, fewer errors are produced since the information is validated by the human experts that introduce it.

It is important to develop means that allow the experts to deal and process all the information in a comfortable way. For instance, it is impossible for experts to deal with 600 alternatives at a time. When comparing the presented method with others on the literature, it is possible to find several papers that deal with the same issue. When these methods are analyzed, it is possible to sort them into the following categories:

- **Clustering methods over the set of experts:** In this category, clustering methods are used in order to create groups of experts that have a similar way of thinking or that fulfil a certain criteria. Thanks to this, it is possible to carry out a better management of them. Some examples can be seen on [42–45]. In [42], authors develop a Group Decision Making method that focuses on applying consensus in environments with a high number of experts. In [43], authors employ hesitant fuzzy sets in order to solve a large scale Group Decision Making environment. Clusters are used in order to deal with the high number of experts. Finally, in [44], they focus on defining personalized individual semantics for decision environments that have a high number of experts. For this purpose, clustering among the experts is performed. In [45], a process to handle minority opinions over environments with high number of experts is presented.
- **Clustering methods over the set of alternatives:** In this category, it is possible to find methods that manage a high number of alternatives. Again, clustering methods are used for this purpose. These methods create groups of alternatives making it easier for the experts to identify and comprehend them. Some examples can be seen in [46,47]. In [46], an alternative ranking base method based on clustering methods is used in order to rank the alternatives in a Group Decision Making environment. In [47], authors classify alternatives in different groups using a hierarchical clustering method in order to allow experts to carry out a double Group Decision Making process in order to select the best alternative.
- **Clustering methods over the set of experts and alternatives:** These methods combine both approaches creating what it is called a double-large scale group decision making method. Some examples can be seen in [48] where the authors present an application in an e-commerce environment.
- **Using networks to relate the DSS elements:** The methods that are classified in this category uses networks in order to establish relations among the elements. Thanks to this, it is possible to establish relations among the elements and analyze the network in order to extract useful information. Some example of methods that use this approach can be found in [49–52]. In [49], social networks are used in order to perform a Group Decision Making method for a high number of experts. In [50], a trust network is built in order to connect experts based on their opinions. It considers situations with a high number of experts. In [51], type-2 fuzzy trust propagation operators that work over social networks are used in order to aggregate information coming from a large set of the experts that are linked among them. In [52], a Group Decision Making method for dealing with minority opinions of a set of experts that are organized in a social network is presented.

Compared to the presented approaches, our method presents the following advantages:

- **Advantages over the clustering approach:** While clustering approaches require the information to be classified into groups, FO queries do not. In an FO, the information is classified according to the properties they meet. Our method only looks for those alternatives that best meet the indicated characteristics. Therefore, it is not necessary to establish similarity criteria between the elements of the FO. Another disadvantage of the clustering approach is that those methods do not totally hide alternatives which are not promising for the experts. Therefore, they are less comfortable. With the FOs, experts do not have to know which alternatives are represented since they only have to provide a query specifying the most desirable characteristics.

- **Advantages over the network approach:** Although networks between different elements that influence decision making can be very useful for experts, they have to understand and interpret them. The method presented does not require this type of action on the part of the experts. They act as a black box leaving the experts with the only duty of providing the desirable characteristics to the FO query. After obtaining the reduced set of alternatives, they can discuss over a reduced set of alternatives. Moreover, as with clustering methods, experts have to deal with all alternatives. Therefore, these methods are less comfortable to the experts than the FO presented one. The less elements the experts have to deal with, the most comfortable is the method for them.

The main disadvantages of the FO approach are that experts do not really analyze all the alternatives that are available. The comfortable environment provided has the cost of not allowing the experts to go over all the alternatives. Experts need to rely on the system and the query provided. In cases where the criteria are clear, not going over all the alternatives is not really a problem at all. Moreover, the FO approach requires precise information about the alternatives. This is because this information is needed for building the FO.

Finally, the presented work will be improved. The presented wine FO is in an initial state and has been designed for testing purposes. More features and conceptualization will be added to it.

## 6. Conclusions

This article presents a DSS that help buyers to make decisions using an FO for generating recommendations. Buyers provide preferences to the system using LM and they receive the set of alternatives that better fulfil their needs according to the provided information. To test the system, an application that aid users in the choosing wines task has been designed. It is important that the FO that contains all the related information has the last items stored on it in order to provide updated information to the buyers. In order to carry out the updating task, two different processes have been designed.

The designed DSS works with imprecise information. For this purpose, LM is used as follows. First, to allow buyers to provide the information to the system. Second, for storing the data in the FO. The designed query process for obtaining recommendations is flexible. This is because the buyers can choose the features that they matter most and they can select the importance that the features will have in the searching process. The FO reasoner works well with imprecise information and it is capable of providing high-quality results. The designed system also benefits from wines users' opinions. The opinion of the users that have already tried the wines is extremely valuable since new buyers can learn about their experience. This information is included in the FO. Moreover, two different architectures that can be used to implement this design are presented. Both of them have different advantages and drawbacks and they should be selected according to the system use.

As future work, it is possible to modify the query in order to provide more flexibility to the users. For instance, it would be possible to modify the query in a way that some of the specified characteristics must be met and some other are met but not required. In order to do this, asymmetric disjunction and conjunctions can be used [53]. The use of weighted mean has several drawbacks [54]. For instance, it is possible for a user to provide full neutrality for all the concepts involved on the query. Furthermore, weights of the most important concepts on the query can become quite low since all of them need to sum 1. In the future, we will try to improve the query by employing aggregation operators that do not have these disadvantages.

**Author Contributions:** Conceptualization, J.A.M.-M. and I.J.P.; methodology, F.J.C. and S.A.; validation, J.A.M.-M. and I.J.P.; formal analysis, J.A.M.-M.; investigation, J.A.M.-M. and S.A.; writing—original draft preparation, J.A.M.-M.; writing—review and editing, F.J.C. and E.H.-V.; supervision, E.H.-V. and I.J.P.; funding acquisition, E.H.-V. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work has been supported by the grant from the FEDER funds provided by the Spanish Ministry of Economy and Competitiveness (No. TIN2016-75850-R), by the Spanish State Research Agency through the project PID2019-103880RB-I00 / AEI / 10.13039/501100011033, and grants from the National Natural Science Foundation

of China(#71725001 and #71910107002). Finally, we would like to thank the “Juan de la Cierva Incorporación” grant from the Spanish Ministry of Economy and Competitiveness.

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

## References

1. Batmaz, Z.; Yurekli, A.; Bilge, A.; Kaleli, C. A review on deep learning for recommender systems: Challenges and remedies. *Artif. Intell. Rev.* **2019**, *52*, 1–37. [CrossRef]
2. Cabrerizo, F.J.; Morente-Molinera, J.A.; Pedrycz, W.; Taghavi, A.; Herrera-Viedma, E. Granulating linguistic information in decision making under consensus and consistency. *Expert Syst. Appl.* **2018**, *99*, 83–92. [CrossRef]
3. Jorro-Aragoneses, J.L.; Recio-García, J.A.; Díaz-Agudo, B.; Jimenez-Díaz, G. RecoLibry-core: A component-based framework for building recommender systems. *Knowl.-Based Syst.* **2019**, *182*, 104854. [CrossRef]
4. Zuheros, C.; Li, C.C.; Cabrerizo, F.J.; Dong, Y.; Herrera-Viedma, E.; Herrera, F. Computing with words: Revisiting the qualitative scale. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2018**, *26*, 127–143. [CrossRef]
5. The Fuzzy Wine Ontology. Available online: <http://users.abo.fi/~rowikstr/FuzzyWineOntology/> (accessed on 28 September 2020).
6. Li, C.C.; Dong, Y.; Herrera, F.; Herrera-Viedma, E.; Martínez, L. Personalized individual semantics in computing with words for supporting linguistic group decision making. An application on consensus reaching. *Inf. Fusion* **2017**, *33*, 29–40. [CrossRef]
7. Zadeh, L.A. *Computing with Words in Information/Intelligent Systems 1: Foundations*; Physica: Berlin/Heidelberg, Germany, 2013; Volume 33.
8. Zadeh, L.A. The concept of a linguistic variable and its application to approximate reasoning. *Inf. Sci.* **1975**, *8*, 199–249. [CrossRef]
9. Qiyas, M.; Abdullah, S.; Ashraf, S.; Abdullah, L. Linguistic Picture Fuzzy Dombi Aggregation Operators and Their Application in Multiple Attribute Group Decision Making Problem. *Mathematics* **2019**, *7*, 764. [CrossRef]
10. Zhang, S.; Gao, H.; Wei, G.; Wei, Y.; Wei, C. Evaluation based on distance from average solution method for multiple criteria group decision making under picture 2-tuple linguistic environment. *Mathematics* **2019**, *7*, 243. [CrossRef]
11. Zadeh, L.A. Fuzzy Sets. *Inf. Control* **1965**, *8*, 338–353. [CrossRef]
12. Bhatia, S.; Richie, R.; Zou, W. Distributed semantic representations for modeling human judgment. *Curr. Opin. Behav. Sci.* **2019**, *29*, 31–36. [CrossRef]
13. He, T.; Wei, G.; Lu, J.; Wei, C.; Lin, R. Pythagorean 2-Tuple Linguistic VIKOR Method for Evaluating Human Factors in Construction Project Management. *Mathematics* **2019**, *7*, 1149. [CrossRef]
14. Li, C.; Yi, J.; Wang, H.; Zhang, G.; Li, J. Interval data driven construction of shadowed sets with application to linguistic word modelling. *Inf. Sci.* **2020**, *507*, 503–521. [CrossRef]
15. Ponti, E.M.; Ohoran, H.; Berzak, Y.; Vulić, I.; Reichart, R.; Poibeau, T.; Shutova, E.; Korhonen, A. Modeling language variation and universals: A survey on typological linguistics for natural language processing. *Comput. Linguist.* **2019**, *45*, 559–601. [CrossRef]
16. Alonso, S.; Pérez, I.J.; Cabrerizo, F.J.; Herrera-Viedma, E. A linguistic consensus model for Web 2.0 communities. *Appl. Soft Comput.* **2013**, *13*, 149–157. [CrossRef]
17. Cabrerizo, F.J.; Al-Hmouz, R.; Morfeq, A.; Martínez, M.Á.; Pedrycz, W.; Herrera-Viedma, E. Estimating incomplete information in group decision making: A framework of granular computing. *Appl. Soft Comput.* **2020**, *86*, 105930. [CrossRef]
18. Cabrerizo, F.J.; Ureña, R.; Pedrycz, W.; Herrera-Viedma, E. Building consensus in group decision making with an allocation of information granularity. *Fuzzy Sets Syst.* **2014**, *255*, 115–127. [CrossRef]
19. Porcel, C.; Ching-Lopez, A.; Bernabé-Moreno, J.; Tejada-Lorente, A.; Herrera-Viedma, E. Fuzzy Linguistic Recommender Systems for the Selective Diffusion of Information in Digital Libraries. *JIPS J. Inf. Process. Syst.* **2017**, *13*, 653–667. [CrossRef]

20. Porcel, C.; Herce-Zelaya, J.; Bernabé-Moreno, J.; Tejeda-Lorente, Á.; Herrera-Viedma, E. Trust Based Fuzzy Linguistic Recommender Systems as Reinforcement for Personalized Education in the Field of Oral Surgery and Implantology. *Int. J. Comput. Commun. Control* **2020**, *15*. [[CrossRef](#)]
21. Huitzil, I.; Alegre, F.; Bobillo, F. GimmeHop: A recommender system for mobile devices using ontology reasoners and fuzzy logic. *Fuzzy Sets Syst.* **2019**. [[CrossRef](#)]
22. Morente-Molinera, J.A.; Cabrerizo, F.J.; Mezei, J.; Carlsson, C.; Herrera-Viedma, E. A dynamic group decision making process for high number of alternatives using hesitant Fuzzy Ontologies and sentiment analysis. *Knowl.-Based Syst.* **2020**, *195*, 105657. [[CrossRef](#)]
23. Qasim, I.; Alam, M.; Khan, S.; Khan, A.W.; Malik, K.M.; Saleem, M.; Bukhari, S.A.C. A comprehensive review of type-2 fuzzy ontology. *Artif. Intell. Rev.* **2020**, *53*, 1187–1206. [[CrossRef](#)]
24. Sweidan, S.; El-Bakry, H.; Sabbeh, S.F. Construction of Liver Fibrosis Diagnosis Ontology From Fuzzy Extended ER Modeling: Construction of FibrOnto From an EER Model. *Int. J. Decis. Support Syst. Technol. (IJDSST)* **2020**, *12*, 46–69. [[CrossRef](#)]
25. Giaretta, P. Philosophy and the Ontologies of Knowledge Representation in AI. In *Ontology Makes Sense*; IOS Press: Amsterdam, The Netherlands, 2019; pp. 24–36.
26. Norris, E.; Finnerty, A.N.; Hastings, J.; Stokes, G.; Michie, S. A scoping review of ontologies related to human behaviour change. *Nat. Hum. Behav.* **2019**, *3*, 164–172. [[CrossRef](#)] [[PubMed](#)]
27. Bobillo, F.; Straccia, U. The fuzzy ontology reasoner fuzzyDL. *Knowl.-Based Syst.* **2016**, *95*, 12–34. [[CrossRef](#)]
28. Carlsson, C.; Brunelli, M.; Mezei, J. Decision making with a fuzzy ontology. *Soft Comput.* **2012**, *16*, 1143–1152. [[CrossRef](#)]
29. Carlsson, C.; Mezei, J.; Brunelli, M. Fuzzy Ontology Used for Knowledge Mobilization. *Int. J. Intell. Syst.* **2013**, *28*, 52–71. [[CrossRef](#)]
30. Morente-Molinera, J.A.; Kou, G.; Pang, C.; Cabrerizo, F.J.; Herrera-Viedma, E. An automatic procedure to create fuzzy ontologies from users' opinions using sentiment analysis procedures and multi-granular fuzzy linguistic modelling methods *Inf. Sci.* **2020**, *476*, 222–238. [[CrossRef](#)]
31. Portmann, E. *The FORA Framework: A Fuzzy Grassroots Ontology for Online Reputation Management*; Springer Science & Business Media: Berkeley, CA, USA, 2012.
32. Portmann, E.; Meier, A. TA fuzzy grassroots ontology for improving weblog extraction. *J. Digit. Inf. Manag.* **2010**, *8*, 276–284.
33. Portmann, E. A fuzzy grassroots ontology for improving social semantic web search. In Proceedings of the 6th International Summer School on Aggregation Operators, Benevento, Italy, 11–15 July 2011.
34. Subramanian, C.M.; Cherukuri, A.K. Fuzzy Role Based Access Control Design using Fuzzy Ontology. *Int. J. Inf. Technol. Proj. Manag. (IJITPM)* **2019**, *10*, 118–136. [[CrossRef](#)]
35. Di Noia, T.; Mongiello, M.; Nocera, F.; Straccia, U. A fuzzy ontology-based approach for tool-supported decision making in architectural design. *Knowl. Inf. Syst.* **2019**, *58*, 83–112. [[CrossRef](#)]
36. Straccia, U. Multi criteria decision making in fuzzy description logics: A first step. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 78–86.
37. Lesani, M.; Naderan, M.; Alavi, S.E. Fuzzy Ontology with ANFIS Neural Network for Semantic Sensor Networks in Smart Homes based on Internet of Things. *Int. J. Web Res.* **2019**, *2*, 26–38.
38. Sumathi, G.; Akilandeswari, J. Improved fuzzy weighted-iterative association rule based ontology postprocessing in data mining for query recommendation applications. *Comput. Intell.* **2020**, *36*, 773–782. [[CrossRef](#)]
39. Huitzil, I.; Bernad, J.; Bobillo, F. GimmeHop: Algorithms for Instance Retrieval and Realization in Fuzzy Ontologies. *Mathematics* **2020**, *8*, 154. [[CrossRef](#)]
40. Fu, C.; Chang, W.; Xue, M.; Yang, S. Multiple criteria group decision making with belief distributions and distributed preference relations. *Eur. J. Oper. Res.* **2019**, *273*, 623–633. [[CrossRef](#)]
41. Luo, S.Z.; Zhang, H.Y.; Wang, J.Q.; Li, L. Group decision-making approach for evaluating the sustainability of constructed wetlands with probabilistic linguistic preference relations. *J. Oper. Res. Soc.* **2019**, *70*, 2039–2055. [[CrossRef](#)]
42. Tang, M.; Zhou, X.; Liao, H.; Xu, J.; Fujita, H.; Herrera, F. Ordinal consensus measure with objective threshold for heterogeneous large-scale group decision making. *Knowl.-Based Syst.* **2019**, *180*, 62–74. [[CrossRef](#)]

43. Wu, Z.; Xu, J. A consensus model for large-scale group decision making with hesitant fuzzy information and changeable clusters. *Inf. Fusion* **2018**, *41*, 217–231. [[CrossRef](#)]
44. Xiao, J.; Wang, X.; Zhang, H. Managing personalized individual semantics and consensus in linguistic distribution large-scale group decision making. *Inf. Fusion* **2020**, *53*, 20–34. [[CrossRef](#)]
45. Gou, X.; Xu, Z.; Liao, H.; Herrera, F. Consensus Model Handling Minority Opinions and Noncooperative Behaviors in Large-Scale Group Decision-Making Under Double Hierarchy Linguistic Preference Relations. *IEEE Trans. Cybern.* **2020**, in press. [[CrossRef](#)]
46. Liu, X.; Xu, Y.; Montes, R.; Ding, R.X.; Herrera, F. Alternative ranking-based clustering and reliability index-based consensus reaching process for hesitant fuzzy large scale group decision making. *IEEE Trans. Fuzzy Syst.* **2018**, *27*, 159–171. [[CrossRef](#)]
47. Morente-Molinera, J.A.; Ríos-Aguilar, S.; González-Crespo, R.; Herrera-Viedma, E. Dealing with group decision-making environments that have a high amount of alternatives using card-sorting techniques. *Expert Syst. Appl.* **2019**, *127*, 187–198. [[CrossRef](#)]
48. Wu, T.; Liu, X.; Qin, J. A linguistic solution for double large-scale group decision-making in E-commerce. *Comput. Ind. Eng.* **2018**, *116*, 97–112. [[CrossRef](#)]
49. Du, Z.J.; Luo, H.Y.; Lin, X.D.; Yu, S.M. A trust-similarity analysis-based clustering method for large-scale group decision-making under a social network. *Inf. Fusion* **2020**. [[CrossRef](#)]
50. Liu, B.; Zhou, Q.; Ding, R.X.; Palomares, I.; Herrera, F. Large-scale group decision making model based on social network analysis: Trust relationship-based conflict detection and elimination. *Eur. J. Oper. Res.* **2019**, *275*, 737–754. [[CrossRef](#)]
51. Tian, Z.P.; Nie, R.X.; Wang, J.Q. Social network analysis-based consensus-supporting framework for large-scale group decision-making with incomplete interval type-2 fuzzy information. *Inf. Sci.* **2019**, *502*, 446–471. [[CrossRef](#)]
52. Ren, R.; Tang, M.; Liao, H. Managing Minority opinions in micro-grid planning by a social network analysis-based large scale group decision making method with hesitant fuzzy linguistic information. *Knowl.-Based Syst.* **2020**, *189*, 105060. [[CrossRef](#)]
53. Hudec, M.; Mesiar, R. The axiomatization of asymmetric disjunction and conjunction. *Inf. Fusion* **2020**, *53*, 165–173. [[CrossRef](#)]
54. Dujmovic, J. *Soft Computing Evaluation Logic: The LSP Decision Method and Its Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2018.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).