Neurocomputing 500 (2022) 921–937

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Survey paper

A gentle introduction and survey on Computing with Words (CWW) methodologies

Prashant K. Gupta [a,c], Javier Andreu-Perez [b,c,d,*]

[a] German Research Center for Artificial Intelligence (DFKI) GmbH, D3.1, Campus D3.2, Saarbrucken 66123, Saarland, Germany
[b] Centre for Computational Intelligence, School of Computer Science and Electronic Engineering, University of Essex, Colchester, United Kingdom
[c] Institute for Advancing Artificial Intelligence, Colchester, United Kingdom
[d] Simbad2, University of Jaén, Jaén, Spain

ARTICLE INFO

Article history:
Received 28 April 2022
Revised 23 May 2022
Accepted 25 May 2022
Available online 28 May 2022
Communicated by Zidong Wang

Keywords:
Augmented Extension Principle based CWW methodology
Extension Principle based CWW methodology
General Type-2 Fuzzy Sets based CWW methodology
Intuitionistic Fuzzy Sets based CWW methodology
Linear General Type-2 Fuzzy Sets based CWW methodology
Perceptual Computing
Rough Sets based CWW methodology
Symbolic Method based CWW methodology
2-tuple based CWW methodology

ABSTRACT

Human beings have an inherent capability to use linguistic information (LI) seamlessly even though it is vague and imprecise. Computing with Words (CWW) was proposed to impart computing systems with this capability of human beings. The interest in the field of CWW is evident from a number of publications on various CWW methodologies. These methodologies use different ways to model the semantics of the LI. However, to the best of our knowledge, the literature on these methodologies is mostly scattered and does not give an interested researcher a comprehensive but gentle guide about the notion and utility of these methodologies. Hence, to introduce the foundations and state-of-the-art CWW methodologies, we provide a concise but a wide-ranging coverage of them in a simple and easy to understand manner. We feel that the simplicity with which we give a high-quality review and introduction to the CWW methodologies is very useful for investigators or especially those embarking on the use of CWW for the first time. We also provide future research directions to build upon for the interested and motivated researchers.

1. Introduction

Computing with Words (CWW)[1] was conceptualized and put forth in the research community for the first time by Prof. Zadeh [1]. The motivation behind the CWW was that if computing systems be built on the principles of CWW, they can think, reason, make decisions and solve problems using a "concepts" driven analytical framework that resembles human symbolic cognition. Human cognition has a remarkable capability to process and reason using semantic uncertainty, which is inevitable in a number of day to day life situations. These situations require decision making using the variables which tend to take qualitative values, naturally and hence the information pertaining to the situation is vague, imprecise and semantically uncertain.[2] An example of such a situation can be deciding whom to befriend when meeting people at a gathering. Here, the decision is based on the perception about the nature or behavior of a person. The variable of interest viz., nature or behavior of a person, used here naturally tends to assume qualitative description and is thus vague, imprecise and contains semantic uncertainty. Another scenario can be the vagueness or imprecision in the semantic understanding of the knowledge provided by experts or decision-makers.

It is a common observation that the classical theory tends the use probabilistic treatment of qualitative concepts and thus tries to quantify the semantic uncertainty by allocating a precise number. Such concepts cannot be defined by boxing them on the basis of their odds of occurring. More often than not, in such situations, the uncertainty is of non-probabilistic nature based on its degree of truth. Thus, there was a need to search for soft information structures to describe these scenarios. This suitable information representation structure was the use of linguistic descriptors for the scenario.

* Corresponding author.
[1] For readers' convenience, all the abbreviations are listed in Table 1
[2] Semantic uncertainty arises due to subjectivity.

https://doi.org/10.1016/j.neucom.2022.05.097
0925-2312/

Table 1
Abbreviations and Their Full Forms.

Abbreviation	Full-Form
AEPCM	Augmented Extension Principle based CWW methodology
CWW	Computing with Words
EPCM	Extension Principle based CWW methodology
FOU	Footprint of Uncertainty
FS	Fuzzy Sets
GT2	General Type-2
GFSCM	General Type-2 Fuzzy Sets based CWW methodology
IFS	Intuitionistic Fuzzy Sets
IFSCM	Intuitionistic Fuzzy Sets based CWW methodology
IT2	Interval Type-2
LGT2	Linear General Type-2
LFSCM	Linear General Type-2 Fuzzy Sets based CWW methodology
LMF	Lower Membership Function
MF	Membership Fucntion
RSCM	Rough Sets based CWW methodology
SMCM	Symbolic Method based CWW methodology
T1	Type-1
2TPCM	2-tuple based CWW methodology
UMF	Upper Membership Function

Human beings have a highly superior capability (with respect to animals) to effortlessly understand, process and operate using linguistic descriptors or 'words'. Words make up the sentences in the natural language.[3] They are vague because people have different understanding and interpretations of the same 'word' [3].[4] Hence, CWW uses words as the units of computation.

Over the years, massive literature has been proposed which presents the views of researchers on the CWW. Prof. Zadeh's other remarkable work in the CWW is [4]. Here, the importance of linguistic information over precise numeric measurements has been advocated by drawing a comparison between perception based information and the numeric measurements. The former is quite closely related to everyday human capabilities to do a number of tasks seamlessly in day to day life. Prof. Yager was of the opinion that Prof. Zadeh's CWW had a very specific way of using the words and hence gave a CWW Model [2], which is made up of three steps (Please see Fig. 1). In the first step, called translation, a mapping or conversion of the linguistic information to its numeric counterpart is performed. The importance of this step is that a computing machine understands only numbers and hence cannot directly operate on the linguistic data. In the next step viz., manipulation, the mapped numeric information is aggregated. This emphasises that numeric information may come from various sources or one source may provide numeric information multiple times and thus aggregation is important for decision making. Finally, in the last step, called retranslation, the aggregated numeric information is mapped to linguistic form, as human beings attach comparatively more relevance to the linguistic information.

Recently there has been a surge in the area of CWW methodologies, which is evident from numerous literary works being contributed in this direction. The popular CWW methodologies proposed in the literature (as per our knowledge so far) are: Extension Principle based CWW methodology (EPCM) [5], Augmented Extension Principle based CWW methodology (AEPCM) [5], Intuitionistic Fuzzy Sets (IFS) based CWW methodology (IFSCM) [5], Symbolic Method based CWW methodology (SMCM) [5], Rough Sets based CWW methodology (RSCM) [5], 2-tuple based CWW methodology (2TPCM) [6,7], Perceptual Computing [3], Linear General Type-2 (LGT2) Fuzzy Sets based CWW methodology (LFSCM) [8], and General Type-2 (GT2) Fuzzy Sets based CWW methodology (GFSCM[5]) [9]. These CWW methodologies use different units of uncertainty for modeling the word semantics. The EPCM, AEPCM and IFSCM, use the type-1 (T1) fuzzy sets (FSs). The SMCM and RSCM, use the ordinal term sets. The 2TPCM performs information representation and computation using a combination of T1 FSs and ordinal term sets. The Perceptual Computing makes use of interval type-2 (IT2) FSs. The LFSCM and GFSCM use the GT2 FSs, however, in the former, the secondary membership function (MF) is a linear function whereas it can be any arbitrary mathematical function in the latter. A categorisation of these CWW methodologies is shown in the form of a mindmap in the Fig. 2.

Table 2 depicts the comparison of these CWW methodologies based on various criteria. It can be seen from the table that the criteria chosen for differentiating the CWW methodologies is primarily the instrument used to model the semantics of linguistic terms (LTs) and linguistic weights (LWs). The instrument are T1 FSs (for the EPCM, AEPCM and IFSCM), ordinal term sets (for SMCM and RSCM), combination of T1 FSs as well as ordinal sets (for 2TPCM), IT2 FSs (for perceptual computing) and GT2 FSs (for LFSCM and GFSCM). Further to differentiate the CWW methodologies from each other within respective instrument, division is extended on the basis of how respective CWW methodology achieves CWW in respective three steps: translation, manipulation and retranslation, of Prof. Yager's CWW (please see Fig. 1). For example, within T1 FSs based CWW methodologies, there are three prominent placeholders: EPCM, AEPCM and IFSCM. EPCM converts T1 MFs of LTs in translation to tri-tuples. These are aggregated in manipulation and converted back to linguistic form using linguistic approximation in last step. AEPCM on the other hand, approximate the T1 MFs of LTs and LWs using tri-tuples in translation, followed by weighted aggregation in manipulation and conversion back to linguistic form using linguistic approximation in retranslation. IFSCM resorts to tri-tuple representation of membership and non-membership for both LTs and LWs in translation. This is followed by weighted aggregation and linguistic approximation in manipulation and retranslation, respectively. Further details about other CWW methodologies will be discussed in detail along with mathematical equations in Sections 3–6.

An important fact needs mention here. We chose these criteria because it enables the reader to see the CWW methodologies in holistic as well as detailed manner. To exemplify, one can see that using T1 FSs as the semantic modeling instrument, three CWW methodologies have been developed viz., EPCM, AEPCM and IFSCM, giving a holistic view of the T1 FSs based CWW methodologies. Each of these CWW methodologies have different internal working for processing the LI (please see Section 2 for details), thus giving a detailed peek into the internal working of the T1 FSs based CWW methodologies. Similar argument applies for other CWW methodologies listed in Table 2. For the convenience of the readers, we have developed a taxonomy of these CWW methodologies, to highlight these differences amongst these methodologies. Please see Fig. 3.

There have been various applications and theoretical research works on these CWW methodologies. EPCM has been used in clinical decision making [10] and multi-criteria decision making [11]. SMCM has been used for multi-person decision making [12] and group decision making [13]. An overview of the application of 2TPCM in decision making can be found in [14], decision support system in [15], risk assessment [16] and decision analysis in [17]. Theoretical works related to 2TPCM have been the proposition of

[3] By natural language is meant the language used by human beings for communication in day to day life.

[4] According to Prof. Mendel, this can be specified in a sentence 'words mean different things to different people'.

[5] It is mentioned here that in the work [5], the EPCM, AEPCM, IFSCM, SMCM and RSCM have been shown to process linguistic information just like the Yager's CWW model of Fig. 1. 2TPCM, Perceptual Computing, LFSCM and GFSM call these respective steps by different names.

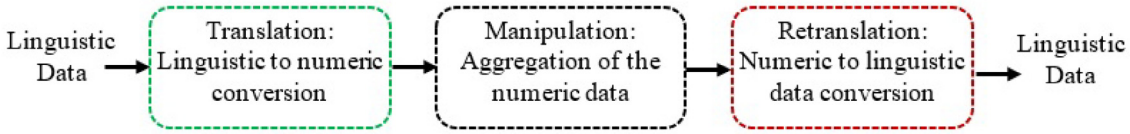

Fig. 1. Prof. Yager's CWW Model [2].

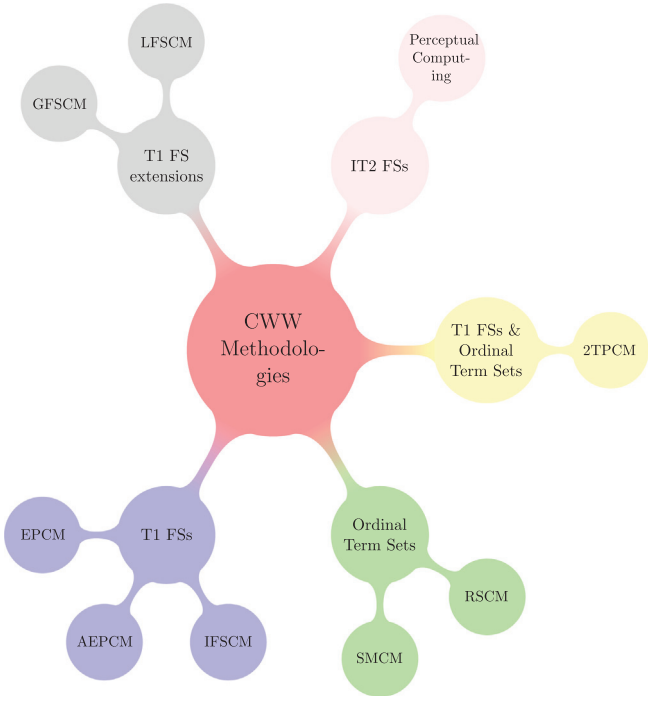

Fig. 2. Mindmap of CWW Methodologies.

a new 2-tuple model for CWW in [18], extended 2-tuple to deal with unbalanced term sets [19], etc. Perceptual computing has been applied for the design of social judgement advisor [20], hierarchical decision making [21], investment judgement analysis [22], journal publication decision making [23], love selection [24], etc. Theoretical works on perceptual computing are [25–35]. The LFSCM has been applied for cooking recipe recommendations in Ambient Intelligent Environments [8,36,37]. Other research works on the LFSCM are [38–42].

These works provide some interesting conclusions. It can be seen that the literature on these CWW methodologies is scattered which makes it difficult to grasp an idea of the field, especially for anyone starting the work on CWW methodologies. The works [7] and [43] tried to provide an overview of 2-tuple methodology and perceptual computer, respectively. They are a good start, however, they have quite limited focuses. Hence, we found no single work which attempted to compile the literature on CWW methodologies and provide the interested readers with a holistic but easily comprehensible view. We feel strongly that absence of such literature will hinder the interested readers to realize the utilities of these methodologies, understand their subtle differences as well as develop a sense of their strengths-limitations.

All these have motivated us to put forth a succinct but wide-ranging coverage of these methodologies, in a simple and easy to understand manner. We feel that the simplicity with which we give a high-quality review and introduction to the CWW methodologies, is very useful for investigators or especially for those embarking on the use of CWW for the first time. We also provide future research directions to build upon for the interested and motivated researchers.

The major contributions of this manuscript are as follows:

- Provide a comprehensive guide and survey that introduces the most relevant state-of-the-art for CWW methodologies, which can always act as a manual for the interested readers.
- Serve as starting point reference to develop an understanding of the differentiating criteria, strengths and limitations of the CWW methodologies.

Table 2
Comparison of CWW Methodologies.

CWW Methodology*	Criteria			
	MS**	LN**	A**	NL**
EPCM	LTs: using T1 MFs	Tri-tuples of T1 MFs	Arithmetic mean	Linguistic approximation
AEPCM	LTs and LWs: using T1 MFs	Tri-tuples of T1 MFs	Weighted Average	Linguistic approximation
IFSCM	Membership and non-membership of LTs and LWs: using T1 MFs	Tri-tuples of T1 MFs	Weighted Average	Linguistic approximation
SMCM	LTs: using ordinal term sets	Indices of ordinal terms	Recursion	Ordinal term set
RSCM	LTs: using ordinal term sets	Indices of ordinal terms	Recursion	Ordinal term set
2TPCM	LTs: using T1 MFs and Ordinal term sets	Combination of T1 MFs and ordinal term sets	Weighted Average and Ordered Weighted Average	Symbolic translation
Perceptual Computing	IT2 FSs	IT2 FS word models	Interval weighted average, fuzzy weighted average and linguistic weighted average	Similarity, ranking, and subsethood
LFSCM	T2 FSs; Secondary MF is a linear function for Shoulder FOUs	Sensory mapping	Conversion to IF-THEN rules and aggregation	Mapping and display to the user by GUI
GFSCM	Finite automata	State space	Transition function	Accepting state

*For full forms, please see Table 1
**MS = Methodology used for modelling semantics of linguistic terms (LTs) and/ or linguistic weights (LWs)
**LN = Linguistic to numeric mapping, **A = Aggregation, **NL = Numeric to linguistic mapping

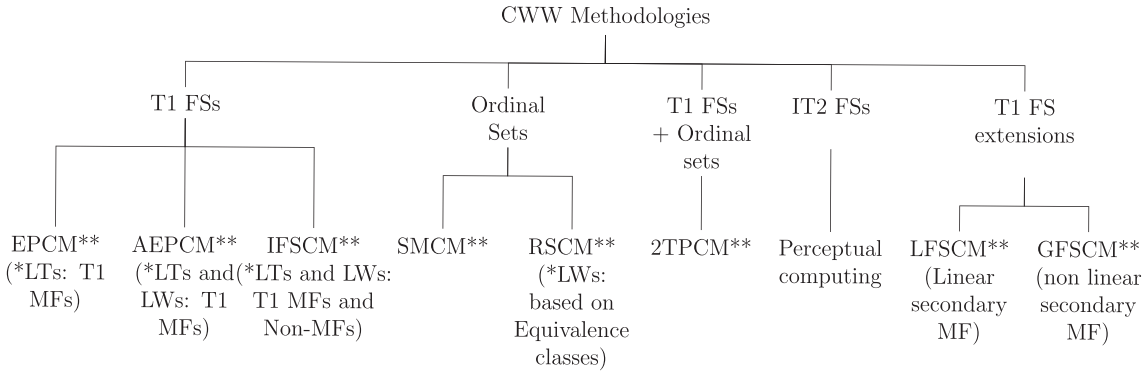

Fig. 3. Taxonomy of CWW Methodologies *LTs = Linguistic terms, LWs = Linguistic Weights, **For full forms, please see Table 1.

- Put forth substantial helpers and tables that compare the utilities and properties of the CWW methodologies, in order to realize the potential of the field.

The remainder of this literary work follows the following organization: Section 2 discusses EPCM, AEPCM and IFSCM; Section 3 gives details on SMCM and RSCM; Section 4 and Section 5 discusses the 2TPCM and Perceptual Computing, respectively. Section 6 gives the details of CWW methodologies based on T2 FSs or extensions of T1 FSs, Section 7 gives the important discussions, based on this research survey, and finally, Section 8 concludes this article and puts forth its future scope. For the convenience of the text readers, we list down all the symbols used for describing the CWW methodologies in Table 3.

2. CWW based on T1 FSs

This section discusses the details of the CWW methodologies that model the word semantics using the T1 FSs. These include the EPCM, AEPCM and IFSCM. These methodologies achieve CWW in the same three steps as in Yager's CWW Model (Please see Fig. 1).

2.1. EPCM

Let's assume that users are required to provide their linguistic preferences in a decision making scenario. These preferences come from a linguistic term set containing $g+1$ distinct linguistic labels. Examples of such labels can be 'very low', 'low', 'medium', etc. Mathematically, if we denote the term set as T and the linguistic labels or terms as $t_0, t_1, \ldots t_g$, then in the set notation form, the term set can be denoted as:

$$T = \{t_0, \ldots, t_g\} \tag{1}$$

As stated earlier, the EPCM models the linguistic terms using the T1 FSs. However, these T1 FSs are generally in the form of uniformly shaped and distributed T1 triangular MFs on a bounded information representation scale. Let's assume an information representation scale whose ends are limited by p and q.[6] Thus, the individual linguistic labels of the term set T can be represented as shown in Fig. 4.

Let's say i number of human subjects choose to elicit their preferences using the linguistic labels from the term set T of (1). Hence, a set called the user preference vector, containing the user feedback can be given as:

[6] It is a common observation that frequently used values of p and q, are respectively 0 and 10. However, any value can be used for them.

Table 3
Symbols used in CWW methodologies and Their meanings.

Symbol	Meaning
T	Linguistic term set containing linguistic terms associated to EPCM, AEPCM, IFSCM, SMCM
UP_{EP}	Collective preference vector containing feedbacks of the users for EPCM, AEPCM, IFSCM, SMCM
UPT_{EP}	UP_{EP} in tri-tuple form for EPCM
C	Collective Preference Vector obtained in Manipulation phase of EPCM, AEPCM, IFSCM
$d(t_q, C)$	Distance between linguistic term t_q and the preference vector C
W	Weight Vector Corresponding to weights of linguistic terms for AEPCM, IFSCM, SMCM, RSCM
UPT_A	UP_{EP} in tri-tuple form for AEPCM
UWT_A	W in tri-tuple form for AEPCM
$L_k \otimes W_k$	Product of k^{th} linguistic preference and its associated weight in AEPCM
UPT_{IFS}	UP_{EP} in tri-tuple form for IFSCM
UWT_{IFS}	W in tri-tuple form for IFSCM
$d'(t_q, C)$	Distance between linguistic term t_q and the preference vector C for non-membership in IFSCM
UPS_{SM}	UP_{EP} in sorted from according to indices for SMCM
AG^i	Recursive function for combining terms from UPS_{SM} and W in SMCM
AG^2	Boundary condition of AG^i
UPS_{RS}	UP_{EP} in sorted from according to indices for RSCM
AG'^i	Recursive function for combining terms from UPS_{RS} and W in RSCM
AG'^2	Boundary condition of AG'^i
UP_{2TP}	UP_{EP} converted to 2-tuple form for 2TPCM
W_{2TP}	W converted to 2-tuple form for 2TPCM
β_{2tp}	Aggregation result of UP_{2TP} and W_{2TP} in 2TPCM
α_{2tp}	Symbolic translation corresponding to β_{2tp}
$\tilde{Y}_{LWA}$	Aggregation result obtained in Perceptual Computing
$C_{\tilde{A}(x)}$	Decoded centroid in Perceptual Computing
Q	Set of states used in GFSCM
$\sum'$	Set of symbols used in GFSCM
δ	Type-2 fuzzy transition function used in GFSCM
q_0	Initial state used in GFSCM
F	GT2 FS of final states used in GFSCM

$$UP_{EP} = \{t_1, \ldots t_i\} \tag{2}$$

where UP_{EP} stands for user preferences in EPCM and each $t_k \in T; k = 1$ to i. This means that each k^{th} index of the linguistic term in (2) equals any one $j = 0$ to g, in (1). The following subsections outline how EPCM processes these user feedbacks in three steps.

2.1.1. Step-1: Translation

In the first step, the linguistic terms constituting the user preference vector, given in (2), are mapped to numeric form. EPCM makes use of the T1 triangular MFs (for semantic representation

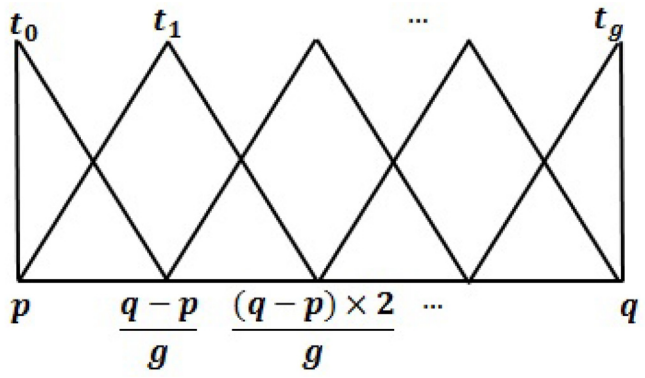

Fig. 4. T1 FS based semantic representation of the terms in term set T [5].

of the linguistic terms) to perform this mapping, which is shown in Fig. 4. The numeric mappings of the linguistic labels given in (2) are provided in the tri-tuple forms like: $(l_k, m_k, r_k); k = 1$ to i. Here, m_k corresponds to the mid point of the triangular T1 MF where the MF achieves its highest value, whereas l_k and r_k correspond to the two ends of the T1 MF where it rests on the x-axis. The tri-tuples are shown in Fig. 4. Therefore, the modified user preference vector becomes:

$$UPT_{EP} = \{(l_1, m_1, r_1), \ldots (l_i, m_i, r_i)\} \tag{3}$$

where UPT_{EP} represents the set containing user preferences in tri-tuple form.

2.1.2. Step-2: Manipulation

In the second step, the mapped numeric representations of the users preferences, given in (3), are aggregated. This is accomplished by averaging the respective l_k's, m_k's and r_k's, to obtain the collective preference vector C as:

$$C = (l_c, m_c, r_c) = \left(\frac{\sum_{k=1}^{i} l_k}{i}, \frac{\sum_{k=1}^{i} m_k}{i}, \frac{\sum_{k=1}^{i} r_k}{i}\right), \quad l_c = \frac{l_1+l_2+\ldots+l_i}{i}, m_c = \frac{m_1+m_2+\ldots+m_i}{i}, r_c = \frac{r_1+r_2+\ldots+r_i}{i} \tag{4}$$

2.1.3. Step-3: Retranslation

In the final step, the numeric collective preference vector, given in (4), is mapped again to the linguistic form. This is useful for understanding by a human subject. As the linguistic terms, corresponding to the decision making scenario at hand, are contained in the linguistic term in T of (1), therefore, the linguistic output from this step should also be a linguistic label from this term set.

In order to map the collective preference vector of (4) to a linguistic label given in T of (1), we calculate the Euclidean distance of the respective three defining points viz., l_c, m_c, r_c of the preference vector from those of each linguistic term in T of (1).[7] The computations are shown in (5):

$$d(t_q, C) = \sqrt{P_1(l_j - l_c)^2 + P_2(m_j - m_c)^2 + P_3(r_j - r_c)^2} \tag{5}$$

where $t_j = (l_j, m_j, r_j), j = 0$ to g, is the tri-tuple representation of each linguistic terms in T of (1), $C = (l_c, m_c, r_c)$, is the collective preference vector of (4), and $P_1 = P_3 = 0.2$ and $P_2 = 0.6$ are the weights. These $P_i, i = 1, 2, 3$ values are taken from [6], however, no restriction exists on the use of a different set of values for respective P_i's. Thus, the desired linguistic term, $t_b^* \in T$, is recommended on the basis of maximum similarity which is essentially equal to minimum Euclidean distance, which can be formally stated as $d(t_b^*, C) \leqslant d(t_j, C), \forall t_j \in T$.

2.2. AEPCM

AEPCM was proposed in [5], as an improvement of EPCM (discussed in Section 2.1). The AEPCM can be utilized for achieving CWW in scenarios the users' linguistic preferences carry different importance and hence can be assigned different weights. The next subsections discuss the working of AEPCM.

2.2.1. Step-1: Translation

Translation involves mapping the linguistic information to numeric form. Consider again the vector holding the containing i users' linguistic feedback as given in (2). Let's say that each of these users' feedback is assigned a linguistic weight, which can be similar or distinct from other weights. Hence, the vector containing the respective weights of the linguistic user preferences can be written as:

$$W = \{w_1, \ldots, w_i\} \tag{6}$$

where $w_p, p = 1, \ldots, i$ is the associated linguistic weight of the p^{th} user's linguistic preference. It is mentioned here that the AEPCM chooses to model the semantics of each $w_p, p = 1, \ldots, i$ using uniformly shaped triangular T1 FS (similar to Fig. 4). Now, just like EPCM, the numeric mapping of each of these linguistic weights is a tri-tuple $\{l, m, r\}$, where m corresponds to the MF point of highest membership values, whereas l and r are the points where the membership value is zero and triangular T1 MF touches the x-axis (please see (3)). Thus, the users' preference vector (given in (2)), and the associated linguistic weights (given in (6)), when represented in the tri-tuple form are given respectively in (7) and (8):

$$UPT_A = \{(l_1, m_1, r_1), \ldots (l_i, m_i, r_i)\} \tag{7}$$

$$UWT_A = \{(l_{w_1}, m_{w_1}, r_{w_1}), \ldots (l_{w_i}, m_{w_i}, r_{w_i})\} \tag{8}$$

where UPT_A and UWT_A stand for user preferences and associated weights, respectively in tri-tuple form for AEPCM. Each $(l_k, m_k, r_k), k = 1, \ldots, i$ and $(l_{w_k}, m_{w_k}, r_{w_k}), k = 1, \ldots, i$ are three defining points of the triangular T1 MF for the user preferences and the respective linguistic weights.

2.2.2. Step-2: Manipulation

In this step, the mapped linguistic information from previous step is combined by performing the weighted aggregation, using the concept of α-cuts given in [5]. If the tri-tuple of a randomly selected linguistic user preference from (7) and its associated weight from (8) are given respectively as $L_k = (l_k, m_k, r_k), k = 1, \ldots, i$ and $W_k = (l_{w_k}, m_{w_k}, r_{w_k}), k = 1, \ldots, i$, then their product is given as:

$$L_k \otimes W_k = \{ll_k, mm_k, rr_k\} = \{l_k \times l_{w_k}, m_k \times m_{w_k}, r_k \times r_{w_k}\}, \quad k = 1, 2, \ldots, i \tag{9}$$

These products are obtained for each linguistic user preference as its associated weight. Hence, the collective preference vector, as an obtained by averaging the respective ll_k's,mm_k's, and rr_k's is given as:

$$C = (l_c, m_c, r_c) = \left(\frac{\sum_{k=1}^{i} ll_k}{i}, \frac{\sum_{k=1}^{i} mm_k}{i}, \frac{\sum_{k=1}^{i} rr_k}{i}\right) \tag{10}$$

[7] Euclidean distance is used as a measure of similarity in [5,6]. However, one is free to choose other measures such as support or cardinality.

2.2.3. Step-3: Retranslation

In the final step, we map the numeric collective preference vector of (10) to linguistic form, for the usefulness of end user. We perform the Euclidean distance based similarity computation as that of EPCM.

2.3. IFSCM

In this section, we discuss the working of IFSCM. For basics on IFS, please see A.

2.3.1. Step-1: Translation

The first step viz., translation, will provide a linguistic to numeric mapping. Consider again the vector holding the linguistic feedbacks of i number of users given in (2) and the associated linguistic weights from (6) of these linguistic feedbacks. As every element in an IFS has an associated degree of membership and non-membership (as shown in A), therefore in IFSCM, each linguistic user preference and the associated weight is represented by a T1 MF. However, each preference and its respective weight has an associated membership and the non-membership degree. The membership degree value follows directly as the three defining points of the tri-tuples (as seen in Fig. 4). In contrast, the non-membership degree values are taken as the average of the tri-tuples of all linguistic terms except the one corresponding to the user linguistic preference (or weight).

For the purpose of elaboration, let's take up as an example the feedback of the first user from the term set of (2) and its associated weight from (6). The user's linguistic preference and its associated weight in the tri-tuple form can be written: $t_1 = (l_1, m_1, r_1)$ and $w_1 = (l_{w_1}, m_{w_1}, r_{w_1})$.[8] The degree of non-membership for the user linguistic preference t_1 (respective linguistic weight w_1) is obtained by averaging respectively the three defining points l_k, m_k, r_k of tri-tuples of all linguistic terms from T except that of t_p (t_q), which is mathematically given as:

$$(l'_1, m'_1, r'_1) = \left(\frac{\sum_{k=0,k\neq p}^{g} l_k}{g}, \frac{\sum_{k=0,k\neq p}^{g} m_k}{g}, \frac{\sum_{k=0,k\neq p}^{g} r_k}{g} \right) \tag{11}$$

$$\left(l'_{w_1}, m''_{w_1}, r'_{w_1}\right) = \left(\frac{\sum_{k=0,k\neq p}^{g} l_k}{g}, \frac{\sum_{k=0,k\neq p}^{g} m_k}{g}, \frac{\sum_{k=0,k\neq p}^{g} r_k}{g} \right) \tag{12}$$

In this manner, the membership degrees and non-membership degrees can be computed for all the linguistic preferences in (2) and their associated weights in (6). Hence, the vector depicting the collection of tri-tuples for the memberships degrees and non-membership degrees for linguistic preferences (associated weights) is given in (13) ((14)) as:

$$UPT_{IFS} = \left[(l_1, m_1, r_1), (l'_1, m'_1, r'_1)\right], ..., \left[(l_i, m_i, r_i), (l'_i, m'_i, r'_i)\right] \tag{13}$$

$$UWT_{IFS} = \left[(l_{w_1}, m_{w_1}, r_{w_1}), \left(l'_{w_1}, m'_{w_1}, r'_{w_1}\right)\right], ..., \left[(l_{w_i}, m_{w_i}, r_{w_i}), \left(l'_{w_i}, m'_{w_i}, r'_{w_i}\right)\right] \tag{14}$$

Here, $m_k(m_{w_k}), k = 1, ..., i$ is the point where the degree of membership for linguistic user preference (associated linguistic weight) achieves its maximum value, where as $l_k, r_k((l_{w_k}, r_{w_k})), k = 1, ..., i$ are two points where the degree of membership for linguistic user preference (associated linguistic weight) has a null value and rests on x-axis. The $l'_k, m'_k, r'_k\left(\left(l'_{w_k}, m'_{w_k}, r'_{w_k}\right)\right)$ are the corresponding values for the degree on non-membership for linguistic user preference (associated respective linguistic weight).

2.3.2. Step-2: Manipulation

The next step is to perform the aggregation of the weighted linguistic preferences and the respective associated linguistic weights. However, as each one is represented in the form of T1 MFs, hence we need to use the α-cuts (similar to Section 2.2). Further, an added processing in IFS comes from performing these computations on the degree of memberships and non-memberships, separately for each user preference. Once this is done, next we aggregate the degrees of memberships (non-memberships) by averaging the three defining points of the weighted user preferences to get the collective preference vector as:

$$\begin{aligned} C &= \{(l_c, m_c, r_c), (l'_c, m'_c, r'_c)\} \\ (l_c, m_c, r_c) &= \left(\frac{\sum_{k=1}^{i} l_k}{i}, \frac{\sum_{k=1}^{i} m_k}{i}, \frac{\sum_{k=1}^{i} r_k}{i} \right) \\ (l'_c, m'_c, r'_c) &= \left(\frac{\sum_{k=1}^{i} l'_k}{i}, \frac{\sum_{k=1}^{i} m'_k}{i}, \frac{\sum_{k=1}^{i} r'_k}{i} \right) \end{aligned} \tag{15}$$

Here, (l_c, m_c, r_c) corresponds to the membership and (l'_c, m'_c, r'_c) corresponds to the non-membership.

2.3.3. Step-3: Retranslation

Now we map the aggregated numeric data from the manipulation step into the linguistic form, using the Euclidean distance as a similarity measure as done for EPCM and AEPCM. The difference lies in the fact that the mapping is performed for each of the membership and non-membership values in the collective preference of (15), which is given as:

$$d(t_q, C) = \sqrt{P_1(l_j - l_c)^2 + P_2(m_j - m_c)^2 + P_3(r_j - r_c)^2} \tag{16}$$

$$d'(t_q, C) = \sqrt{P_1(l_j - l'_c)^2 + P_2(m_j - m'_c)^2 + P_3(r_j - r'_c)^2} \tag{17}$$

where the $P_i, i = 1, 2, 3$ are the weights, with values $0.2, 0.6$ and 0.2 respectively.

Finally, there are two recommended linguistic terms viz., $t_b^* \in T$ and $t_b^{*\prime} \in T$, corresponding respectively to the membership and non-membership.

3. CWW based on Ordinal term sets

In this section, we discuss the CWW methodologies which use the ordinal term sets for achieving the CWW. These are the SMCM and the RSCM. It is mentioned here that both SMCM and RSCM can process the differentially weighted linguistic user preferences. Also, both these methodologies achieve CWW in three steps viz., translation, manipulation and retranslation.

3.1. SMCM

Let's now discuss the working of the SMCM, where the user preferences, as well as the associated respective weights, are represented through their indices in the term sets.

3.1.1. Step-1: Translation

The starting point of the symbolic method based linguistic computational model is a linguistic preference vector that includes the user preferences. Consider the linguistic preference set from (2). Each of these linguistic preferences may have an associated weight,

[8] It's mentioned here that we have assumed that $t_1, w_1 \in T, T$ being given in (1). Therefore, $\exists t_p \in T, p = 0, ..., g$, such that $t_1 = t_p$. Also $\exists t_q \in T, q = 0, ..., g$, such that $w_1 = t_q$. Further, t_p can be equal or unequal to t_q.

given in the form of a weight vector given in (6). An additional condition that SMCM imposes on this weight vector is that all the weights must add to 1. This is stated as $\forall w_p \in W, w_p \in [0,1]; p = 1$ to i, is the weight associated to the p^{th} user linguistic preference in the term set given in (2).

3.1.2. Step-2: Manipulation

In the second step, the linguistic user preferences from (2) are sorted in descending order according to the indices of the linguistic terms drawn from T of (1). After ordering, the user linguistic preferences may be given as:

$$UPS_{SM} = \{T_1, \ldots T_i\} \tag{18}$$

where UPS_{SM} stands for user preferences in sorted order for the SMCM, $T_k \in T, k = 1, ..., i$. The linguistic preference vector from (18) is then order weighted aggregated using the recursive function $\left(AG^i\right)$, given as:

$$AG^i\{w_p, I_{T_p}, p = 1, 2, \ldots, i | i > 2, i \in Z\} = \{w_1 \odot I_{T_1}\} \oplus \left\{(1 - w_1) \odot AG^{i-1}\{\delta_h, I_{T_h}, h = 2, \ldots, i\}\right\} \tag{19}$$

where $I_{T_p}, p = 1, ..., i, I_{T_{jh}}, h = 2, ..., i$ are the indices of the linguistic terms given in (18) and $\delta_h = \frac{w_h}{\sum_{l=2}^{i} w_l}; h = 2, 3, ..., i$. As the aggregation function given in (19), is a recursive function, therefore, we need a base condition when the recursion bottoms out. This is achieved, when the number of terms to be aggregated is reached at two. Hence, upon reaching the boundary condition, the aggregation function becomes as shown in (20):

$$AG^2\{\{w_{i-1}, w_i\}, \{I_{T_{i-1}}, I_{T_i}\}, |i = 2\} = \{w_{i-1} \odot I_{T_{i-1}}\} \oplus \{w_i \odot I_{T_i}\} \tag{20}$$

where $I_{T_{i-1}}$ and I_{T_i} are the respective indices of the remaining terms from the preference vector (18), with respective weights w_{i-1} and w_i. Hence, the linguistic term is recommended in the next step.

3.1.3. Step-3: Retranslation

In this step, a linguistic term is recommended in the output. As a starting point for the same, for the (20), a numeric index value is recommended using the computations shown in (21) as:

$$I_r = min\left\{i, I_{T_i} + round\left(\frac{w_{i-1} - w_i + 1}{2} \cdot \left(I_{T_{i-1}} - I_{T_i}\right)\right)\right\} \tag{21}$$

here $round()$ is the round function, given as $round(x) = \lfloor x + 0.5 \rfloor, x \in R$, $\lfloor\rfloor$ being the floor function[9]

Now, starting from (19), the recursive function AG^i is called AG^{i-1}, which in turn calls for AG^{i-2} and so on until AG^2 is reached, where the recursion bottoms out. The recommended numeric index I_r using (21) for AG^2 is then fed to AG^3, which again recommends a linguistic term. Thus, backtracking $i-2$ intermediate recursive equations in this manner, we reach the original recursive function AG^i and recommend a numeric index for it too.[10]

Finally, the recommended numeric index for the recursive function AG^i can be matched to one of the terms from (1), to generate a linguistic recommendation in the output.

[9] A special case of (21) can arise when $w_{i-1} = w$ and $w_i = 1 - w$, for a random value of the weight w. Hence, rewriting the indices $I_{T_{i-1}} = I_l$ and $I_{T_i} = I_q$, (20) and (21) get modified to (22) and (23), respectively as:

$$AG^2\{\{w, 1 - w\}, \{I_l, I_q\}\} = \{w \odot I_l\} \oplus \{(1 - w) \odot I_q\} \tag{22}$$

$$I_r = min\{i, I_q + round(w.(I_l - I_q))\} \tag{23}$$

[10] For detailed computations, please see [5]

3.2. RSCM

The RSCM is based on the rough sets [44–48], which are obtained from crisp sets by drawing a lower and upper approximations. The RSCM uses the indiscernibility property of rough sets and the concepts of SMCM. For details on indiscernibility, please see B.

3.2.1. Step-1: Translation

For the purpose of illustrating the working of linguistic to numeric mapping in RSCM, let's take the starting point as the linguistic feedbacks of i users in the vector given in (2). RSCM proceeds by dividing these user feedbacks into equivalence classes using the indiscernibility property. Hence, the vector containing users' feedbacks takes the form of a vector of equivalence classes and is given in (24) as:

$$\{C_1, C_2, \ldots C_n\} \tag{24}$$

where each $C_i, i = 1, ..., n$ or an equivalence class is a set containing same linguistic preferences grouped together and defined as $C_i = (t_1, t_2, \ldots t_p)$, where each $t_q \in T; q = 1$ to $p, t_1 = t_2 = \ldots = t_p, T$ being taken from (1). As each C_i is a set, therefore we define the class cardinality or $|C_i|, i = 1, ..., n$, as the number of linguistic preferences constituting the class.

From Section 3.1 it follows that the summation of the respective associated weights of the corresponding linguistic preferences yields 1. As there are n equivalence classes in (24), therefore, each class receives a weight of $1/n$. Further, since the class cardinality is $|C_i|, i = 1, ..., n$, so every linguistic term within a class can be allocated a weight of $1/(n \times |C_i|)$. Consequently, each linguistic preferences given in (2) is allocated a weight and the new weight vector can be given in (25) as:

$$W = \{w'_1, ..., w'_i\} \tag{25}$$

here each $w'_p \in [0,1]; p = 1$ to i is the respective associated weight of the user feedback taken from (2). Also, it follows trivially that $\sum_{p=1}^{i} w'_p = 1$.

3.2.2. Step-2: Manipulation

The aggregation in RSCM is performed similarly to that of SMCM discussed in Section 3.1, using the recursive function. Initially, the linguistic preferences are sorted according to their indices, and therefore the new vector depicting the user preferences becomes:

$$UPS_{RS} = \{T_1, T_2, \ldots T_i\} \tag{26}$$

where $T_k \in T, k = 1, ..., i$. It is mentioned here that each of the T_k may or may not be equal to respective t_k. RSCM uses a recursive function $\left(AG'^i\right)$ for aggregation similar to SMCM and is given in (27) as:

$$AG'^i\left\{w'_p, I_{T_p}, p = 1, 2, \ldots, i | i > 2, i \in Z\right\} = \{w'_1 \odot I_{T_1}\} \oplus \left\{(1 - w'_1) \odot AG'^{i-1}\{\delta_h, I_{T_h}, h = 2, \ldots, i\}\right\} \tag{27}$$

where $I_{T_p}, p = 1, ..., i, I_{T_h}, h = 2, ..., i$ are the indices of the linguistic terms given in (26) and $\delta_h = \frac{w'_h}{\sum_{l=2}^{i} w'_l}; h = 2, 3, ..., i$. The recursive function $\left(AG'^i\right)$ reaches a boundary condition when the number of terms to be aggregated is two, when the recursive function looks like as shown in (28):

$$AG'^2\{\{w'_{i-1}, w'_i\}, \{I_{T_{i-1}}, I_{T_i}\}, |i = 2\} = \{w'_{i-1} \odot I_{T_{i-1}}\} \oplus \{w'_i \odot I_{T_i}\} \tag{28}$$

where $I_{T_{i-1}}$ and I_{T_i} are the indices of the remaining terms from the user linguistic preference vector (26), and w'_{i-1} and w'_i are their respective associated weights.

3.2.3. Step-3: Retranslation

Now we generate a numeric index for the aggregated user preferences from the previous step. This numeric value is mapped into an output linguistic recommendation. The numeric index of the term for (28) is I_r, given as:

$$I_r = min\left\{i, I_{T_i} + round\left(\frac{w'_{i-1} - w'_i + 1}{2}.(I_{T_{i-1}} - I_{T_i})\right)\right\} \tag{29}$$

where $round()$ is the round function[11]

Thus, similar to SMCM, we started the aggregation process with the recursive function AG'^i in (27), which calls the function AG'^{i-1}, which will in turn call function AG'^{i-2} and so on till we reach AG'^3 which in the boundary step case calls AG'^2 in (28), thus in all we pass through $i-2$ intermediate recursion calls. The recommended index for AG'^2 is calculated using (29). From here, we back track and will use this value as input to AG'^3. Hence, backtracking through a series of $i-2$ calls, the final recommended numeric index for AG'^i can be found. This recommended index is matched to terms in (1) to generate a linguistic recommendation.

4. CWW based on 2TPCM

CWW can be achieved by a novel methodology called the 2TPCM, which uses a combination of T1 FSs and ordinal term sets. T1 FSs are used to model the semantics of the linguistic terms, whereas the processing involves using the indices of these terms in the term set. In the term set, these terms are represented using the ordinal term sets. The 2TPCM achieves CWW in three steps viz., information gathering phase, aggregation phase and exploitation phase.

4.1. Step-1: Information gathering Phase

In the 2TPCM, every piece of the linguistic preference is represented as twin valued viz., the linguistic term and its translation or distance from the nearest linguistic term (called symbolic translation) in the term set. Simply for the purpose of illustration, consider a term set as: $\{s_0 : VL, s_1 : L, s_2 : M, s_3 : H, s_4 : VH\}$, where VL stands for *VeryLow*, identified by the index s_0 or 0. The L, M, H and VH stand respectively for $Low, Medium, High$ and *VeryHigh*, with respective indices being s_1 (1), s_2 (2), s_3 (3) and s_4 (4). Thus, a piece of numeric information, say 3.2 is represented as $(s_3, +0.2)$ or $(H, +0.2)$ viz., it is at a distance of $+0.2$ from the linguistic term H. Further, a numeric information with a value 3.8 is represented as $(s_4, -0.2)$ or $(VH, -0.2)$ viz., it is at a distance of -0.2 from the linguistic term VH.

Consider again the user preferences given in (2) and their associated linguistic weights in (6). The user preferences and the respective associated weights are represented as: $(t_k, \alpha), t_k \in T, T$ being the term set given in (1), and α, called the symbolic translation is given as $\alpha \in [-0.5, 0.5)$. Thus, the two sets are given as:

$$UP_{2TP} = \{(t_1, \alpha_1), \ldots, (t_i, \alpha_i)\} \tag{32}$$

$$W_{2TP} = \{(w_1, \alpha_1), \ldots, (w_i, \alpha_i)\} \tag{33}$$

As each of $t_k \in T$ and $w_k \in T, k = 1$ to i, therefore all the $\alpha_k = 0$.

4.2. Step-2: Aggregation Phase

The users' feedbacks and their associated weights from (32) and (33) respectively, are aggregated using the weighted arithmetic mean to compute the aggregation value β_{2tp} as:

$$\beta_{2tp} = \frac{(w_1 \times I_{t_1}) + \ldots + (w_i \times I_{t_i})}{w_1 + \ldots + w_i} \tag{34}$$

where $I_{t_k}, k = 1, \ldots, i$ are the indices of the linguistic terms are given in (32).

4.3. Step-3: Exploitation Phase

In this step, the β_{2tp}, obtained by aggregation in (34), needs to be converted back to the linguistic form. For this purpose, the symbolic translation α_{2tp} for the β_{2tp} in (34) is given as:

$$\alpha_{2tp} = \beta_{2tp} - round(\beta_{2tp}), -0.5 \leqslant \alpha_{2tp} < 0.5 \tag{35}$$

Finally, the linguistic term output from the exploitation phase is:

$$t_{recommended} = \left(t_{round(\beta_{2tp})}, \alpha_{2tp}\right) \tag{36}$$

The linguistic value of the recommended linguistic term is obtained by matching $t_{round(\beta_{2tp})}$ to the terms of T, of (1).

5. CWW based on Perceptual Computing

Perceptual Computing is a novel CWW methodology that models the word semantics using IT2 FSs. Perceptual computing also achieves CWW in three steps similar to that of Yager's CWW framework as shown in Fig. 1. However, the three steps are called encoder, CWW engine and decoder. The three steps are bound in a framework called the perceptual computer or Per-C, which is shown in Fig. 5. We now discuss the internal working of the Per-C in detail.

5.1. Encoder

The encoder converts the words into their nine point IT2 FS models, which form the numeric representations for the linguistic information. The words and their IT2 FS word models are stored together in a codebook. First of all, a vocabulary of problem-specific words is decided. Following this, the generating of the respective IT2 FS word model for a word involves the collection of endpoint data intervals from a group of subjects generally through a survey. The subjects are asked to provide their opinions about the locations of endpoints. They are asked a question, "Assume that the endpoints of the word can be located on a scale of 0 to 10. Where do you think the endpoints of the word lie?".

Scenarios may arise where the subjects are unavailable or the system needs to be designed to generate personalized recommendations. By personalized recommendations, we mean that the requirements of an individual user play a greater role in the system design than the group of users. For such cases, a novel approach of Person Footprint of Uncertainty (FOU) was proposed in [49]. The Person FOU approach proceeds by taking an interval for each of the left and right endpoints of every word in the vocabulary by a single user, instead of a group of subjects. Then a uniform distribution is assumed to exist for both these data intervals and fifty ran-

[11] A special case of (29) is possible when $w'_{i-1} = w$ and $w'_i = 1 - w$, for any weight w. Hence, denoting $I_{T_{i-1}} = I_l$ and $I_{T_i} = I_q$, the boundary condition and the recommended term index equations can be given as:

$$AG'^2\{\{w, 1-w\}, \{I_l, I_q\}\} = \{w \odot I_l\} \oplus \{(1-w) \odot I_q\} \tag{30}$$

$$I_r = min\{i, I_q + round(w.(I_l - I_q))\} \tag{31}$$

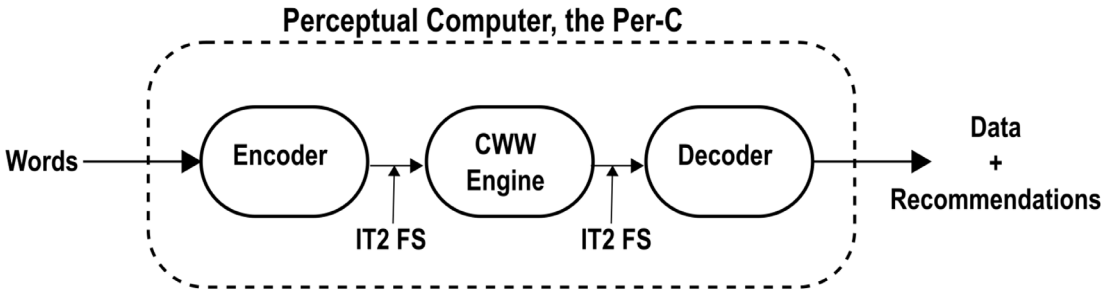

Fig. 5. The Perceptual Computer (Per-C).

dom numbers are produced in each of the left and right intervals, denoted as $(L_1, ..., L_{50})$ and $(R_1, ..., R_{50})$, respectively. Following this, pairs of the form $(L_i, R_i), i = 1$ to 50, are constructed by picking a value from each of the left and right intervals, so that now each data pair becomes an interval provided by i^{th} (virtual) subject.

The data intervals, collected in the case of perceptual computing or constructed in the case of Person FOU can be processed by the age-old Interval Approach (IA) [25]. However, IA was later on identified to have some limitations and therefore, the improved Enhanced Interval approach (EIA) [26] was proposed, which in turn was improved into the Hao-Mendel Approach (HMA) [27]. The IA, EIA or HMA accomplishes the data processing using the data part and the FS part. We will describe in detail each of the IA, EIA as well HMA, starting from IA. All the Equations used in the data and FS part of IA are given in Table 4.

5.1.1. IA: Data Part

The data part is broadly divided into data pre-processing of the data intervals collected from a group of subjects (or constructed as in Person FOU) and statistics computation for the remaining data intervals from pre-processing. The pre-processing involves subjecting the data intervals to bad data processing, removal of the outliers, low confidence interval removal by tolerance limit processing, and reasonable-interval processing. We assume that the number of subjects are n (there are 50 virtual subjects in Person FOU), and the endpoint data intervals for a word as $\left[a^{(i)}, b^{(i)}\right], i = 1, 2, ..., n$.

Step-1: Removing bad data Bad data is synonymous with the data intervals that either lie outside the assumed information scale of 0 to 10 or where lower endpoint value of the interval is greater than (or equal to) the right endpoint value. All such data intervals are considered unsuitable for further processing and hence dropped from the set of useful data intervals by performing the comparisons given in (37). Hence, some of the data intervals may be dropped, thereby reducing the number of data intervals to $n', n \geqslant n'$.

Step-2: Outliers removal This is the next step within data pre-processing. The outliers are identified in the n' remaining data intervals through Box and whisker test. These computations are shown in (38), where $Q^a(0.25), Q^b(0.25)$, and $Q^L(0.25)$ are the first quartiles for the interval left endpoints, right endpoints and lengths, respectively. $Q^a(0.75), Q^b(0.75)$, and $Q^L(0.75)$ are the corresponding third quartiles, whereas the IQR^a, IQR^b, and IQR^L are the corresponding words, interquartile ranges. The data intervals appearing as outliers in the test are filtered out, thereby reducing the number of data intervals to $m', n' \geqslant m'$.

Step-3: Removing low confidence data intervals using tolerance limit calculation Tolerance factor k provides $100(1-\gamma)\%$ confidence that an interval contains at least the proportion $(1-\alpha)\%$ of data values. The tolerance limit calculation is shown in (39), where m^a, m^b and m^L are the mean values for interval left endpoints, right endpoints and lengths, respectively. In contrast, s^a, s^b and s^L are the respective standard deviation values. Thus, out of the m' surviving data intervals from the previous step, the ones not satisfying (39) are dropped from further processing, thereby giving rise to a reduced set of data intervals totalling $m'', m' \geqslant m''$.

Step-4: Reasonable interval processing For the m'' surviving data intervals from the previous step, computations are performed as given in (40), using an optimal value of a threshold ε^*. In (41), m^a and m^b are the mean of the interval left and right endpoints, respectively, whereas s^a and s^b are the corresponding standard deviation values, for surviving m'' data intervals of tolerance limit processing. The intervals satisfying (40) have a high overlap with other intervals and hence are retained, and others are possibly dropped thereby attracting a trimming in the numbers of data intervals to $m, m'' \geqslant m$.

Step-5: Statistics computation for data intervals In this step, mean and standard deviation are computed for all the m remaining data intervals from the previous step, assuming a probability distribution. The computations for this step are given in (42) and (43), assuming uniform probability distribution.

5.1.2. IA: FS Part

The possible remaining n data intervals obtained from the data part are processed using the FS part in various steps, as discussed now.

Step-6: Selecting T1 FS models In this step, the T1 MFs are classified into either left-shoulder, interior or right shoulder based on the mean and standard deviation values computed in (42) and (43).

Step-7: Determining FS uncertainty measures After identifying the T1 MFs to belong to one of the three types viz., interior, left shoulder and right shoulder, the two FS uncertainty measures: mean and standard deviation, are established for all the T1 MFs.

Step-8: Computing the uncertainty measures for T1 FS models Now, the mean and standard deviation FS uncertainty measures are computed for T1 FS models as shown in (44)–(46). In these equations, a_{MF} and b_{MF} are the points on x-axis where the ends of T1 MF rest.

Step-9: General formulae for T1 FS model parameters In this step, the FS uncertainty measure values of the T1 MFs (42) and (43) are equated to the corresponding values from (44)–(46). This is done in order to compute the general formulae for parameters defining the T1 FS models, which are given in (47)–(49).

Step-10: Establishing the type of FOU Now, it is established that the FOU belongs to which one of the interior or shoulder FOUs by using the t-test on the mean and standard deviation values computed in (47)–(49).

Step-11: Computing the embedded T1 FSs In this step, the T1 FSs, called the embedded T1 FSs, are obtained for the m data intervals (after deciding on whether the FSs are interior or shoulder

Table 4
Equations of IA

Name	Equation
Step 1: Bad Data Processing	$0 \leqslant a^{(i)} < b^{(i)} \leqslant 10, i = 1, ..., n$ (37)
Step 2: Outlier processing	$\left.\begin{array}{l} a^{(i)} \in \left[Q^a(0.25) - 1.25IQR^a, Q^a(0.75) + 1.5IQR^a\right] \\ b^{(i)} \in \left[Q^b(0.25) - 1.25IQR^b, Q^b(0.75) + 1.5IQR^b\right] \\ L^{(i)} \in \left[Q^L(0.25) - 1.25IQR^L, Q^L(0.75) + 1.5IQR^L\right] \end{array}\right\} \mathrm{i} = 1, ..., \mathrm{n}\prime$ (38)
Step 3: Tolerance limit processing	$\left.\begin{array}{l} a^{(i)} \in \left[m^a - ks^a, m^a + ks^a\right] \\ b^{(i)} \in \left[m^b - ks^b, m^b + ks^b\right] \\ L^{(i)} \in \left[m^L - ks^L, m^L + ks^L\right] \end{array}\right\} \mathrm{i} = 1, ..., \mathrm{m}\prime$ (39)
Step 4: Reasonable interval processing	$a^{(i)} < \varepsilon^* < b^{(i)}, i = 1, ...m$ (40)
	$\varepsilon^* = \frac{\left(m^b(s^a)^2 - m^a(s^b)^2\right) \pm s^a s^b \left[\left(m^a - m^b\right)^2 + 2\left((s^a)^2 - (s^b)^2\right)\ln\frac{s^a}{s^b}\right]^{\frac{1}{2}}}{(s^a)^2 - (s^b)^2}$ (41)
Step 5: Computing the statistics for surviving data intervals	$S_i = \left(m_Y^{(i)}, s_Y^{(i)}\right), i = 1, ..., m$ (42)
	$m_Y^{(i)} = \frac{(b^{(i)} + a^{(i)})}{2}, s_Y^{(i)} = \frac{(b^{(i)} - a^{(i)})}{\sqrt{12}}, i = 1, ..., m$ (43)
Step 6: Computing the uncertainty measures for T1 FS models	$IMF^1 : m_{MF} = \frac{(b_{MF} + a_{MF})}{2}, s_{MF} = \frac{(b_{MF} - a_{MF})}{2\sqrt{6}}$ (44)
	$LMF^1 : m_{MF} = \frac{(b_{MF} + 2a_{MF})}{3}, s_{MF} = \left[\frac{1}{6}\left[(a_{MF} + b_{MF})^2 + 2a_{MF}^2\right] - m_{MF}^2\right]^{\frac{1}{2}}$ (45)
	$RMF^1 : m_{MF} = \frac{(b_{MF} + 2a_{MF})}{3}, s_{MF} = \left[\frac{1}{6}\left[(a\prime_{MF} + b\prime_{MF})^2 + 2a\prime_{MF}^2\right] - m\prime_{MF}^2\right]^{\frac{1}{2}}$ (46)
	$a\prime_{MF} = M - b_{MF}, b\prime_{MF} = M - a_{MF} and m\prime_{MF} = M - m_{MF}$
Step 7: Computing general formulae for parameters of T1 FS models	$IMF^1 : a_{MF}^{(i)} = \frac{1}{2}\left[\left(a^{(i)} + b^{(i)}\right) - \sqrt{2}\left(b^{(i)} - a^{(i)}\right)\right], b_{MF}^{(i)} = \frac{1}{2}\left[\left(a^{(i)} + b^{(i)}\right) + \sqrt{2}\left(b^{(i)} - a^{(i)}\right)\right]$ (47)
	$LMF^1 : a_{MF}^{(i)} = \frac{(a^{(i)} + b^{(i)})}{2} - \frac{(b^{(i)} - a^{(i)})}{\sqrt{6}}, b_{MF}^{(i)} = \frac{(a^{(i)} + b^{(i)})}{2} + \frac{\sqrt{6}(b^{(i)} - a^{(i)})}{3}$ (48)
	$RMF^1 : a_{MF}^{(i)} = M - \frac{(a\prime^{(i)} + b\prime^{(i)})}{2} - \frac{\sqrt{6}(b\prime^{(i)} - a\prime^{(i)})}{3} b_{MF}^{(i)} = M - \frac{(a\prime^{(i)} + b\prime^{(i)})}{2} + \frac{(b\prime^{(i)} - a\prime^{(i)})}{\sqrt{6}}$ (49)
	$a\prime^{(i)} = M - b^{(i)}, b\prime^{(i)} = M - a^{(i)}$
Step 8: Calculating the embedded T1 FSs	$\left(a^{(i)}, b^{(i)}\right) \rightarrow \left(a_{MF}^{(i)}, b_{MF}^{(i)}\right), i = 1, 2, \ldots, m$ (50)
Step 9: Deleting the inadmissible T1 FSs	$a_{MF}^{(i)} \geqslant 0 and b_{MF}^{(i)} \leqslant 10, i = 1, ..., m$ (51)
Step 10: Computing an IT2 FS	$\widetilde{W} = \bigcup_{i=1}^{m^*} W^{(i)}$ (52)
Step 11: Computing the mathematical model for FOU	$\underline{a}_{MF} \equiv \min_{i=1,\ldots,m^*}\left\{a_{MF}^{(i)}\right\}, \bar{a}_{MF} \equiv \max_{i=1,\ldots,m^*}\left\{a_{MF}^{(i)}\right\}$ (53)
	$\underline{b}_{MF} \equiv \min_{i=1,\ldots,m^*}\left\{b_{MF}^{(i)}\right\}, \bar{b}_{MF} \equiv \max_{i=1,\ldots,m^*}\left\{b_{MF}^{(i)}\right\}$
	$C_{MF}^{(i)} = \frac{a_{MF}^{(i)} + b_{MF}^{(i)}}{2}, \underline{C}_{MF} \equiv \min\left\{C_{MF}^{(i)}\right\}, \bar{C}_{MF} \equiv \max\left\{C_{MF}^{(i)}\right\}$

[1]I MF = Interior Membership Function, L MF = Left Shoulder Membership Function,
[1]R MF = Right Shoulder Membership Function

FOUs), by using the computations shown in (50). The embedded FSs are denoted as $W^{(i)}, i = 1, ..., m$.[12]

Step-12: Deleting the inadmissible T1 FSs This step involves the removal of inadmissible intervals, which fail to satisfy the (51). This may be followed by reducing in the number of data intervals from m to $m^*, m \geqslant m^*$.

Step-13: Computing an IT2 FS word models Here, the wavy slice representation theorem comes into play for the computation of the IT2 FS word model using the embedded T1 FSs from the previous step. The computation for the IT2 FS word model is shown in (52).

Step-14: Computing the mathematical model for FOU In this final step, the mathematical model of the FOU viz., $FOU\left(\widetilde{W}\right)$ is computed as shown in (53). Once this mathematical model is computed, the parameters for the Upper MF and the Lower MF are computed for all the IT2 FS models, which are denoted as $UMF\left(\widetilde{W}\right)$ and $LMF\left(\widetilde{W}\right)$), respectively.

In this manner, the IT2 FS word models are generated for every word of the vocabulary, using the data part and FS part. The IT2 FS word models belong to one of three categories viz., left shoulder, interior or right shoulder FOU as shown in Fig. 6. The obtained IT2 FS word models are defined using the nine point parameters: four for the UMF and five for the LMF, which are collectively called the FOU data. The FOU data along with corresponding words are stored in the form of a codebook.

5.1.3. EIA & HMA: Data Part

EIA was proposed to improve the data processing in the data part of the IA, to ensure the better acquisition of the targeted data intervals. Hence, the EIA differs in the data pre-processing of data part, though EIA also consists of the data part and the FS part. So, we will discuss the differences only here. Again, let's assume that our data collection set involves data intervals provided by n subjects (there are 50 virtual subjects in Person FOU) and their data intervals are in the form $\left[a^{(i)}, b^{(i)}\right], i = 1, 2, ..., n$. When these data intervals are subjected to the data pre-processing, we proceed as follows.

Step-1[13]**: Bad data processing** In this step, the IA ensures that the endpoints of the interval lie within the scale of 0 to 10 as well as the left endpoint of the interval is less than the right one. The EIA imposes these constraints on the interval length also. These computations are shown in (37). Thus, the intervals satisfying these computations are accepted, which may trim the number of data intervals to $n\prime$ from n.

[12] The m number of data intervals were obtained at the end of Data part.

[13] **replace the same step in IA**

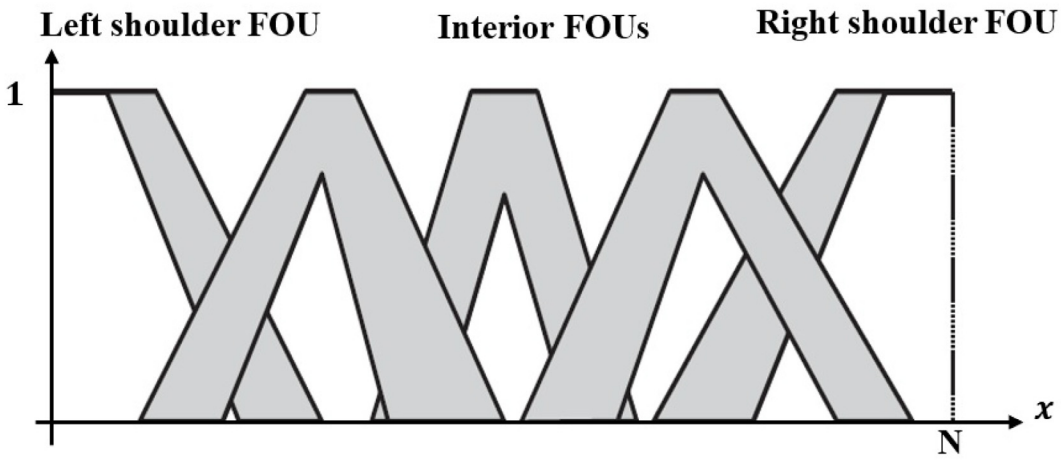

Fig. 6. The IT2 FS word models obtained with IA.

$$0 \leqslant a^{(i)} < b^{(i)} \leqslant 10, b^{(i)} - a^{(i)} < 10, i = 1, ..., n \tag{37}$$

Step-2]: Outlier processing The EIA initially identifies the outliers within the surviving n' interval endpoints using Box and whisker test, thereby leading to a possible reduced set of data intervals totalling $n'', n' > n''$. The computations are given as:

$$\left.\begin{aligned} a^{(i)} &\in \left[Q^a(0.25) - 1.25\mathrm{IQR}^a, Q^a(0.75) + 1.5\mathrm{IQR}^a\right] \\ b^{(i)} &\in \left[Q^b(0.25) - 1.25\mathrm{IQR}^b, Q^b(0.75) + 1.5\mathrm{IQR}^b\right] \end{aligned}\right\} \mathrm{i} = 1, ..., \mathrm{n}' \tag{38}$$

Here, $Q^a(0.25), Q^a(0.75)$ and IQR^a are respectively the first quartiles, third quartiles and interquartile range for the interval left endpoints. The $Q^b(0.25), Q^b(0.75)$ and IQR^b are the corresponding values for the interval right endpoints.

Following this, the outliers are identified for the interval lengths of the n'' surviving data intervals, using again the Box and whisker test. These computations are given as:

$$L^{(i)} \in \left[Q^L(0.25) - 1.25\mathrm{IQR}^L, Q^L(0.75) + 1.5\mathrm{IQR}^L\right], i = 1, 2, \ldots, n'' \tag{39}$$

Here, $Q^L(0.25), Q^L(0.75)$ and IQR^L are respectively the first quartile, third quartile and interquartile range. Thus, after performing an additional Box and Whisker test sequentially on the interval endpoints and on the interval lengths, there is a possibility of trimming the number of data intervals to $m', m' \leqslant m'$.

Step-3]: Tolerance limit processing In EIA, the tolerance limit processing also enables us to identify high confidence intervals. Likewise the previous step, the tolerance limit processing is applied on the interval endpoints, using a confidence factor k, shown as:

$$\left.\begin{aligned} a^{(i)} &\in \left[m^a - ks^a, m^a + ks^a\right] \\ b^{(i)} &\in \left[m^b - ks^b, m^b + ks^b\right] \end{aligned}\right\} \mathrm{i} = 1, ..., \mathrm{m}' \tag{40}$$

Here, m^a and s^a are the mean and standard deviation, respectively of the interval left endpoints. m^b and s^b are the corresponding values for interval right endpoints. The computations in (40) may lead to a reduction in the number of data intervals from m' to m^+. Following this, the computations are performed on interval lengths of the surviving m^+ data intervals, using a confidence factor k' as:

$$L^{(i)} \in \left[m^L - k's^L, m^L + k's^L\right], i = 1, ..., m^+ \tag{41}$$

where m^L is the mean of the interval length and s^L is the standard deviation. k' in (41) is given as:

$$k' = min(k_1, k_2, k_3) \tag{42}$$

The value of k_1 enables that one can state with 95% confidence that the interval $\left[m^L - k_1 s^L, m^L + k_1 s^L\right]$ contains at least the proportion 95% of data values. The values k_2 and k_3 are given as:

$$k_2 = \frac{m^L}{s^L}, k_3 = \frac{(10 - m^L)}{s^L} \tag{43}$$

The computations in (43) are performed to retain only the not too small or not too large intervals. Hence, at the end of the tolerance limit processing step, the number of data intervals may be reduced to m''.

Step-4]: Reasonable interval processing Reasonable interval processing accepts those intervals that have a high amount of overlap with other intervals. These intervals are identified using the computations shown in (44):

$$a^{(i)} < \varepsilon^* < b^{(i)}, 2m^a - \varepsilon^* \leqslant a^{(i)}, b^{(i)} \leqslant 2m^b - \varepsilon^*, i = 1, ...m'' \tag{44}$$

here m^a and s^a are the mean and standard deviation, respectively of the left endpoints. The m^b and s^b are the corresponding values for the right endpoints. Further, ε^* is the optimal value of the threshold which is computed as shown in (41). Finally, at the end of reasonable interval processing, the set of data intervals may be trimmed to a length of m.

5.1.4. EIA: FS part

The EIA FS part performs all the computations in the same nine steps as that in IA, except in the last step viz., calculating the mathematical model of FOU, where the LMF height of interior FOUs is calculated in such a manner that it avoids completely filled and flat FOUs.

Hence, EIA also generates the IT2 FS word models for all the vocabulary words and stores them in a codebook. It is mentioned here that EIA FOUs are also either interior or shoulder, as shown in Fig. 6.

5.1.5. HMA: FS part

HMA improves the FS part IA/EIA in order to extract more information from the data intervals. Upon receiving the m surviving data intervals from the data part, they are processed for overlap computation in the FS part. Then the IT2 FS word models are then computed using a smaller set of remaining data intervals that do not contain the overlap. Let's discuss the working of the various steps in the FS part in detail. It is mentioned here that only the

steps in the FS part of HMA which differ from that of EIA are discussed here.

Step-6]: Establishing the nature of FOU To establish whether a data interval may be associated to an interior or a shoulder FOU, one-sided tolerance limits are calculated for the m remaining data intervals of the data part. This is given in (45) as:

$$\underline{a} = \hat{m}_a - k(m)\hat{s}_a, \bar{b} = \hat{m}_b - k(m)\hat{s}_b \tag{45}$$

where $\hat{m}_a$ and $\hat{s}_a$ are respectively the sample mean and standard deviation for m data intervals' left endpoints. The $\hat{m}_b$ and $\hat{s}_b$ are the corresponding values for the right endpoints. $k(m)$ is the one-sided tolerance factor (and a function of m). Given the value of the one-sided tolerance factor, it can be stated with 95% confidence that the given limit contains greater than or equal to 95% data.

After computing these one sided tolerance limits, the word model (W) is classified into the interior or one of the shoulder FOUs based on these comparisons:

$$\begin{cases} \textit{if } \underline{a} \leqslant 0 : \textit{Left Shoulder FOU} \\ \textit{if } \bar{b} \geqslant 10 : \textit{Right Shoulder FOU} \\ \textit{Otherwise} : \textit{Interior FOU} \end{cases} \tag{46}$$

Step-7]: Computing the interval overlap Here, the computations for the overlap intervals $[o_a, o_b]$ are performed for the interior as well as the shoulder FOUs as:

$$[o_a, o_b] = \begin{cases} \left[0, \min_i b^{(i)}\right], i = 1, \ldots, m : \textit{Left Shoulder FOU} \\ \left[\max_i a^{(i)}, 10\right], i = 1, \ldots, m : \textit{Right Shoulder FOU} \\ \left[\max_i a^{(i)}, \min_i b^{(i)}\right], i = 1, \ldots, m : \textit{Interior FOU} \end{cases} \tag{47}$$

Step-8]: Removing the interval overlap The purpose of calculating the overlap intervals $[o_a, o_b]$ in the previous step was to remove them from all the m data intervals to arrive at a smaller sets of data intervals for the interior and the shoulder FOUs by performing the computations as:

$$\begin{aligned} &\left[o_b = \min_i b(i), b(i)\right], i = 1, \ldots, m : \textit{Left Shoulder FOU} \\ &\left[a^{(i)}, o_a = \max_i a^{(i)}\right], i = 1, \ldots, m : \textit{Right shoulder FOU} \\ &\left[a^{(i)}, o_a = \max_i a^{(i)}\right] \& \left[o_b = \min_i b(i), b(i)\right], i = 1, \ldots m : \textit{Interior FOU} \end{aligned} \tag{48}$$

Step-9]: Mapping the data intervals into FOU parameters Once the smaller set of intervals is obtained using the computations in (48), the quantities in the equation are mapped into the FOU parameters of the interior or shoulder FOUs. For left shoulder FOU the parameters $[b_l, b_r]$ are mapped to the respective interval values from (47). The same action follows for the parameters $[a_l, a_r]$ of the right shoulder FOU. The interior FOU mapping is slightly different, and looks like: $\left[a^{(i)}, o_a = \max_i a^{(i)}\right]$ to $[a_l, a_r]$, and $\left[o_b = \min_i b^{(i)}, b^{(i)}\right]$ to $[b_l, b_r]$.

Based on these parameter values, the obtained FOU plots with the HMA look similar to the ones shown in Fig. 7. By visual inspection, one can make out the difference between the IA/ EIA FOU plots shown in Fig. 6 and HMA FOU plots shown in Fig. 7. In the IA/ EIA IT2 FS word models, the UMF has a height of 1. The LMF may have a height of 1 for the shoulder IT2 FS word models and generally not for the interior ones. The HMA IT2 FS word models, on the other hand, plot a height of 1 for both the UMF and the LMF.

5.2. CWW Engine

CWW engine performs the task of aggregating the user feedback in the form of IT2 FS word models for the codebook words. Whenever multiple stakeholders provide their inputs, at the encoder, in the form of words, these words are converted to IT2 FS word models as outlined above in Section 5.1. Their nine point data about the IT2 FS word models are extracted from the codebook and given as input to the CWW engine. CWW engine can use different types of aggregation operators like interval weighted average (IWA), fuzzy weighted average (FWA), linguistic weighted average (LWA), etc. The differentiating factor for the use of these operators is the nature of the data to be aggregated. IWA is used when the data and the weights are no more than the intervals. FWA is used when at least one piece of information from the data or the weights are T1 FSs but not IT2 FSs, whereas the LWA is used when at least one piece of information from the data or the weights are IT2 FSs.

The computations for aggregating the IT2 FS word models for the words and their associated weights using the LWA are accomplished as:

$$\widetilde{Y}_{LWA} = \frac{\sum_{i=1}^{n} \widetilde{X}_i \widetilde{W}_i}{\sum_{i=1}^{n} \widetilde{W}_i} \tag{49}$$

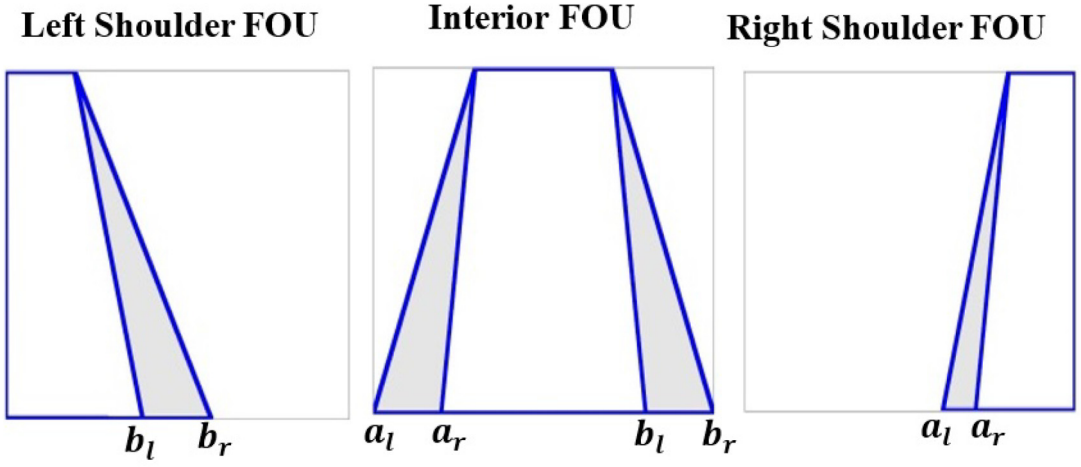

Fig. 7. FOU plots obtained with HMA.

Here $\widetilde{X}_i$ is the IT2 FS models of the words to be aggregated and $\widetilde{W}_i$ are those of the corresponding weights.

5.3. Decoder

The output of the CWW engine viz., $\widetilde{Y}_{LWA}$, is also an IT2 FS and is generally not an exact match to any of the IT2 FS word models in the codebook. Hence, the decoder section performs the task of generating linguistic recommendations by performing various computations. The linguistic recommendation or the 'word' is generated using Jaccard's similarity measure.

The decoder is also capable of generating 'ranking' and 'class' as recommendations. The most commonly used ranking method is the centroid ranking, where the centroid of an IT2 FS is given by the union of the centroids of all its embedded T1 FSs, $c(A_e)$, as shown in (50):

$$C_{\underset{\sim}{A(x)}} = \bigcup_{\forall A_e} c(A_e) \equiv [c_l, c_r] \tag{50}$$

The values of c_l and c_r in (50) are given as:

$$c_l = \frac{\sum_{i=1}^{L} x_i \bar{\mu}_{\underset{\sim}{A}}(x_i) + \sum_{i=L+1}^{N} x_i \underline{\mu}_{\underset{\sim}{A}}(x_i)}{\sum_{i=1}^{L} \bar{\mu}_{\underset{\sim}{A}}(x_i) + \sum_{i=L+1}^{N} \underline{\mu}_{\underset{\sim}{A}}(x_i)} \tag{51}$$

$$c_r = \frac{\sum_{i=1}^{R} x_i \underline{\mu}_{\underset{\sim}{A}}(x_i) + \sum_{i=R+1}^{N} x_i \bar{\mu}_{\underset{\sim}{A}}(x_i)}{\sum_{i=1}^{R} \underline{\mu}_{\underset{\sim}{A}}(x_i) + \sum_{i=R+1}^{N} \bar{\mu}_{\underset{\sim}{A}}(x_i)} \tag{52}$$

In (51)–(52), L and R are called switch points at which the aggregation switches between the UMF and LMF. They can be computed using EIASC algorithm, Karnik Mendel (KM) algorithm and Enhanced Karnik Mendel (EKM) algorithm (Fig. 8).

The mean, $c\left(\widetilde{A}\right)$, of c_l and c_r is used in the centroid ranking method and computed as:

$$c\left(\widetilde{A}\right) = \frac{(c_l + c_r)}{2} \tag{53}$$

6. CWW based on T1 FS extensions

In this section, we discuss the details of the CWW methodologies that model the semantics of the linguistic terms using the T2 FSs, and hence called the CWW methodologies based on the extension of T1 FSs. These are: LFSCM and the GFSCM. In the LFSCM, the secondary MF of the shoulder FOUs is a linear function. On the other hand, in GFSCM, the MF of the interior, as well as shoulder FOUs, need not necessarily be a linear function.

6.1. LFSCM

LFSCM models the word semantics using the T2 FSs, where the secondary MF is a linear function, and hence these FSs are called the LGT2 FSs [8]. The LFSCM uses a CWW architecture, as shown in Fig. 9. The LFSCM consists of mainly two parts viz., granulation and causation-organization (with each part consisting of several other interacting components). Granulation works on the principle of decomposing a whole unit into parts, whereas causation associates the causes with effects. The organization works opposite to the granulation by integrating parts into a whole. Inside the LFSCM, a sub-component exists commonly referred to as the 'neural architecture for perceptual decision-making', which contains four modules viz., Sensory Evidence or NA1 (responsible for accumulation and comparison of sensory evidence), Uncertainty or NA2 (has the task of perceptual uncertainty detection), Decision Variables or NA3 (decision variables are represented) and User Feedback/ Performance Monitoring or NA4 (monitors the performance by detecting errors and adjustment of decision strategies). Further, the system requires Memory for data accumulation and hence it is also one of the components. In the terminology of the neural architecture, it is commonly referred to as the Experience.

6.1.1. Step-1: Granulation

The operation of the LFSCM is triggered by the problem specific input 'words', to the granulation segment. Here, these words are mapped into their numerical representations, based on the sensory evidence. The sensory evidence acts as a solution descriptor relates to decision variables in human reasoning.

6.1.2. Step-2: Causation-Organization

In the next segment, viz., causation-organization, these numerical representations are associated with the appropriate fuzzy IF-THEN rules (fuzzy IF-THEN rules deal with the uncertainty of the human reasoning). The output of these IF-THEN rules is aggregated and converted back to the linguistic form, for communication to the user. The words are displayed to the user through a Graphical User Interface (GUI). The user is asked to evaluate the output of the system and provide his/ her feedback. Thus, based on the user feedback, the IF-THEN rules can be modified to incorporate the user preferences. Further, it is pertinent to mention that for the first time, the system output is generated based on the default rule con-

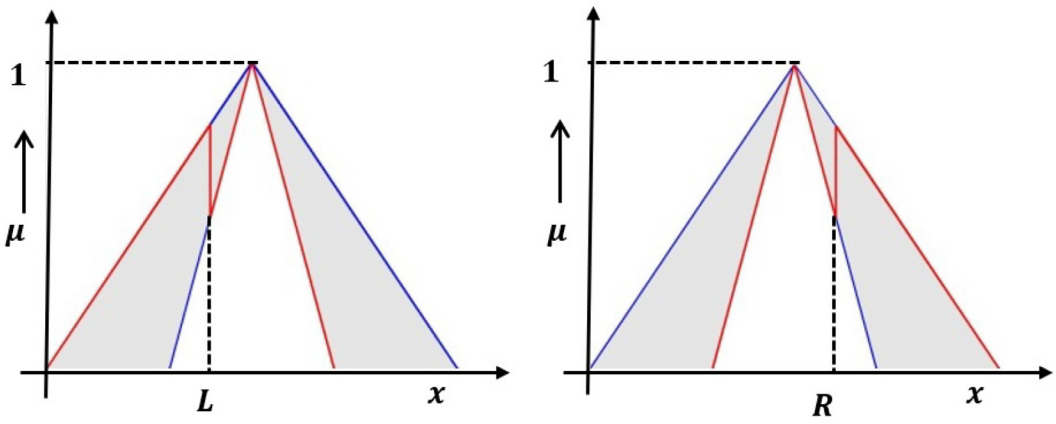

Fig. 8. Switch Points for Centroid Calculation in IT2 FS.

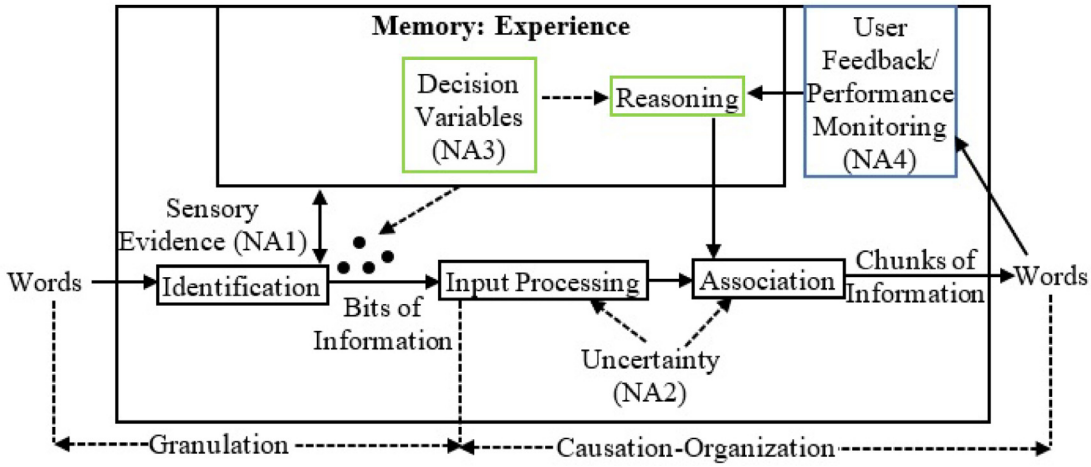

Fig. 9. Architecture for LFSCM [8].

figuration. Subsequent user interaction enables the populating of IF-THEN rule base with user preference based rules and hence more user acceptable output words can be obtained from the system.

6.2. GFSCM

A GFSCM using various types of fuzzy automata was proposed in [9], of which we discuss here the one based on universally GT2 fuzzy finite automata. According to the definition given in [9], the universally GT2 fuzzy finite automata for CWW is a quin tuple given as: $M = (Q, \sum', \delta, q_0, F)$. Here, the meanings of each of the quantities in the quin tuple are given as:

- $Q = \{q_0, q_1, ..., q_N\}$ is the finite set of states.
- The $\sum'$ is the finite set of symbols, so-called the underlying input alphabet. It is mentioned that $\sum'$ is the type-2 fuzzy subset of $GT2F(\sum)$, where $\sum$ is the finite input alphabet.
- δ is the type-2 fuzzy transition function defined as: $\delta : Q \times \sum' \rightarrow GT2F(Q)$. Thus, for any $q_i \in Q$ and $a \in \sum', \delta(q_i, a)$ may be considered as a possibility distribution of the states that the automata in state q_i can enter given the input alphabet a. This can be written more generally as $\delta(q_i, a)(q_j), q_j$ being the state to enter from q_i with the given input alphabet a.
- $q_0 \in Q$ is the initial state.
- F is the GT2 fuzzy subset of Q, called the GT2 FS of final states. More generally, $F(q_i), \forall q_i \in Q$, denotes the degree to which q_i is the final state.

In particular, when $\sum' = GT2F(\sum)$, then the universal GT2 automata is the fuzzy automata for CWW (for all words).

7. Discussions

We now discuss some important views related to the work presented here.

A remarkable work [5], has reasoned the supremacy of AEPCM and IFSCM over the EPCM as well as RSCM over the SMCM. Further, there has been a number of works in the literature where the IT2 FS based Perceptual Computing has been shown to possess much better performance than the EPCM, SMCM and 2TPCM. The details can be found here [50]. However, no comparison has been made so far amongst perceptual computing and AEPCM, IFSCM and RSCM. This remains an area worth exploring.

In [8], the LFSCM is shown to give almost 50% better performance than the perceptual computing. Thus, it is safe to conclude that as we use the higher level FSs to model the 'word' semantics, the model accuracy increases. However, all this comes at the expense of a higher computational cost.

It is quite evident that a lot of research work has been done in Perceptual computing. However, there has not been much research on the other CWW methodologies. Further, a recent novel work [51], gives a thorough literature review on various types of FSs. Out of these, there are various types of FSs on which CWW methodologies have not been proposed so far. Interested researchers can work on this line of research.

The CWW methodology perceptual computing [3] models the word semantics using the IT2 FSs, the words come from human perception. However, the basis for perception [52,53] is information acquisition through the senses as well as making something useful out of it. The process of how human beings perceive through the senses and acquire information has a lot of complex psychological processes acting underlying it. It is in itself a very broad research area. On the contrary, in all these works on perceptual computing, the basis for information acquisition is the data collected about the location of the endpoints of a word, on a scale of 0 to 10, which a human subject thinks. There is no such process involved in the perceiving of any information through the senses. It is just how a human subject has an opinion about the location of word endpoints. So, in our opinion, the word "perception" is slightly misleading.

A research article that attempted to put forth the idea of achieving the CWW using Machine learning, was [54]. In this article, the author used the scenario of medical diagnostics as a bedrock and highlighted the use of linguistic information for expressing a physician's knowledge. The author uses the concept of fuzzy random variable based probability theory to model the linguistic semantics. Also, in [55], the authors used the concept of Choquet integral for achieving the CWW. However, in [7,14], it has been advocated that more often than not, the uncertainty in real-life observations is non-probabilistic in nature. It stems from the semantic uncertainty of linguistic information. Hence, the CWW methodologies [54,55] are not close to human cognition in processing linguistic information.

It has been stated in [56] that CWW should make use of fuzzy logic. FSs do a great job at modeling the intra and inter uncertainty in the semantics of linguistic terms in a manner close to human beings. However, they also suffer from some shortcomings. According to us, major amongst those is the MF generation for

modeling the semantics of linguistic terms in CWW. The T1 FSs based CWW methodologies divide the information representation scale uniformly amongst the T1 fuzzy MFs and the ordinal term sets based CWW methodologies assume the complete information to be modeled in terms of ordered locations of linguistic terms on the information representation scale. The 2TPCM uses a combination of both T1 MFs and the ordinal terms. Hence, all these three category of CWW methodologies have a reduced capability to model word semantics compared to IT2 FSs based perceptual computing of the CWW methodologies based on T1 FS extensions. The IT2 FSs based perceptual computing, on the other hand, generate MFs by collecting data from group of people. This has its limitations, like being time consuming, etc. and T1 extension based CWW methodologies are computationally intensive. Therefore, to overcome the limitation of IT2 FS based CWW methodology, a work is being done in this direction [57]. The challenge is to automate the MF generation as well as simultaneously modeling the word semantics in best possible way.

Recently there has been a surge in the area of symbolic learning [58]. It represents concepts using symbols and then relationships are defined amongst them. Further, they are increasingly being used in conjunction with sub-symbolic systems for various purposes. CWW is often considered synonymous to the symbolic learning. However, there are some subtle differences. CWW attempts to model the semantic uncertainty of the linguistic terms so as to bring it close to the human cognitive process. This is required so that if a computing system be developed using the CWW principles, it should be able to process the linguistic information in a manner similar to the human beings. To ensure this, it is paramount that the system handles the semantics uncertainty in a manner as close as possible to the human cognition. Further, as can be seen from Fig. 1, there is no learning involved in any CWW methodology. Its emphasis lies on mapping the linguistic information to numeric (step 1: translation) by using the instrument that can capture and model the linguistic uncertainty in best possible way. This is followed by aggregation (step 2: manipulation) and finally generating linguistic recommendation back (step 3: retranslation).

8. Conclusions and Future Scope

Computing with Words (CWW) was proposed by Prof. Zadeh, as a novel approach that aims to impart computing machines with the capability to process linguistic information like human beings. There have been numerous notable works in the CWW, both application-based and theoretical. It has been successfully applied in various application areas like risk assessment, decision analysis, decision support systems, etc.

On the theoretical side, an important line of work has been the development of various CWW methodologies, which has resulted in various diverse research works. These include research works on the EPCM, AEPCM, IFSCM, SMCM, RSCM, 2TPCM, Perceptual computing, LFSCM and GFSCM. However, to the best of our knowledge, the literature on these methodologies is mostly scattered and does not give an interested researcher a comprehensive overview of the notion and utility of these methodologies. Hence, we have addressed this issue in this work and given the readers a succinct but wide survey of the CWW methodologies.

We have also highlighted the future research directions to build upon for the interested and motivated researchers. These include the performance comparisons between the various CWP methodologies (wherever non-existent), dealing with the trade-off between methodology accuracy and computational cost, and a direction for developing more CWP methodologies based on different types of FSs.

CRediT authorship contribution statement

Prashant K. Gupta: Conceptualization, Methodology, Writing - review & editing. **Javier Andreu-Perez:** Conceptualization, Methodology, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Basics of IFS

We give here a brief introduction to the IFSs [59–62]. IFSs generalize the concept of T1 FSs. The T1 FS represents the uncertainty about each set element through its membership value. The IFS extend the notion of this uncertainty by an added quantity to each set element viz., the degree of non-membership. Mathematically, an IFS (A) is defined on a universe X as:

$$A = \{(x, \mu_A(x), \nu_A(x) | x \in X\} \tag{A.1}$$

where $\mu_A(x)$ and $\nu_A(x)$ is the degree of membership and degree of non-membership, respectively (Graphically are shown in Fig. 10), both satisfying the condition: $0 \leqslant \mu_A(x), \nu_A(x) \leqslant 1$, and $\mu_A(x) + \nu_A(x) \leq 1$. An IFS differs from a T1 FS as in the latter, $\nu_A(x) = 1 - \mu_A(x)$ and law of excluded middle hold.

Appendix B. Indiscernibility of Rough Sets

Indiscernibility is an important property of rough sets and forms a basis of various operations. To illustrate the concept, consider a system for modeling the information I. Let's assume that there is a universe of discourse U and a non-empty finite attribute set A defined on it. Let's say there exists an attribute $a \in A$ and there is defined a set V_a, such that a can take values from V_a. Then I is a mapping defined in the following manner: $I : U \rightarrow V_a$.

Let's say there is a mapping mechanism to assign a value $a(x)$ from V_a to each attribute a and object x in the universe U. Then, P-indiscernibility, $IND(P)$, of a rough set defined mathematically as:

$$IND(P) = \left\{(x, y) \in U^2 | \forall a \in P, a(x) = a(y)\right\} \tag{B.1}$$

where $P \subseteq A$, forms an associated equivalence relation. The U can be divided into a set of equivalence classes $IND(P)$ denoted by $U/IND(P)$ or U/P.

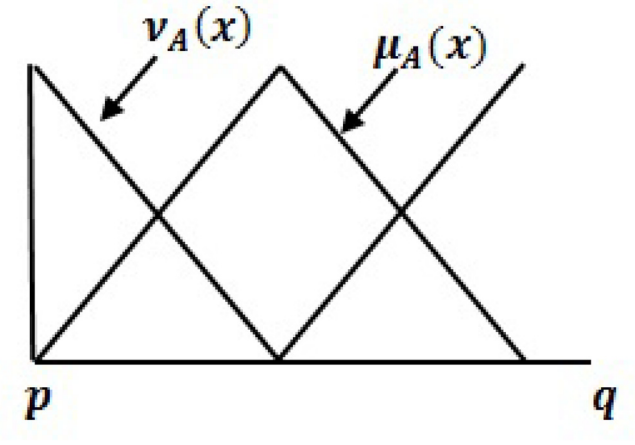

Fig. 10. IFS: Representation.

References

[1] L. Zadeh, Fuzzy logic = computing with words, IEEE Transactions on Fuzzy Systems 4 (1996) 103–111, https://doi.org/10.1109/91.493904.
[2] R. Yager, Computing with words and information/intelligent systems 2: applications, chapter approximate reasoning as a basis for computing with words, 1999..
[3] J. Mendel, D. Wu, Perceptual computing: aiding people in making subjective judgments, volume 13, John Wiley & Sons, 2010.
[4] L.A. Zadeh, From computing with numbers to computing with words. from manipulation of measurements to manipulation of perceptions, IEEE Transactions on circuits and systems I: fundamental theory and applications 46 (1999) 105–119..
[5] P.K. Gupta, D. Sharma, J. Andreu-Perez, Enhanced linguistic computational models and their similarity with yager's computing with words, Information Sciences (2021).
[6] F. Herrera, L. Martínez, A 2-tuple fuzzy linguistic representation model for computing with words, IEEE Transactions on fuzzy systems 8 (2000) 746–752.
[7] L. Martínez, R.M. Rodriguez, F. Herrera, 2-tuple linguistic model, in: The 2-tuple Linguistic Model, Springer, 2015, pp. 23–42..
[8] A. Bilgin, H. Hagras, A. Ghelli, D. Alghazzawi, G. Aldabbagh, An ambient intelligent and energy efficient food preparation system using linear general type-2 fuzzy logic based computing with words framework [application notes], IEEE Computational Intelligence Magazine 10 (2015) 66–78.
[9] Y. Jiang, A general type-2 fuzzy model for computing with words, Intl. Journal of Intelligent Systems 33 (2018) 713–758.
[10] R. Degani, G. Bortolan, The problem of linguistic approximation in clinical decision making, International Journal of Approximate Reasoning 2 (1988) 143–162.
[11] W. Pedrycz, P. Ekel, R. Parreiras, Fuzzy multicriteria decision-making: models, methods and applications, John Wiley & Sons, 2011.
[12] R.R. Yager, Non-numeric multi-criteria multi-person decision making, Group Decision and Negotiation 2 (1993) 81–93.
[13] Z. Xu, A method based on linguistic aggregation operators for group decision making with linguistic preference relations, Information sciences 166 (2004) 19–30.
[14] L. Martı, F. Herrera, et al., An overview on the 2-tuple linguistic model for computing with words in decision making: Extensions, applications and challenges, Information Sciences 207 (2012) 1–18.
[15] L. Martinez, D. Ruan, F. Herrera, Computing with words in decision support systems: an overview on models and applications, International Journal of Computational Intelligence Systems 3 (2010) 382–395.
[16] J. Liu, L. Martínez, H. Wang, R.M. Rodríguez, V. Novozhilov, Computing with words in risk assessment, International Journal of Computational Intelligence Systems 3 (2010) 396–419.
[17] T. Malhotra, A. Gupta, A systematic review of developments in the 2-tuple linguistic model and its applications in decision analysis, Soft Computing (2020) 1–35.
[18] J.-H. Wang, J. Hao, A new version of 2-tuple fuzzy linguistic representation model for computing with words, IEEE transactions on fuzzy systems 14 (2006) 435–445.
[19] M.-A. Abchir, I. Truck, Towards an extension of the 2-tuple linguistic model to deal with unbalanced linguistic term sets, arXiv preprint arXiv:1304.5897 (2013)..
[20] D. Wu, J.M. Mendel, Social judgment advisor: An application of the perceptual computer, in: FUZZ-IEEE, IEEE, 2010, pp. 1–8..
[21] D. Wu, J.M. Mendel, Computing with words for hierarchical decision making applied to evaluating a weapon system, IEEE Transactions on Fuzzy Systems 18 (2010) 441–460.
[22] J. Mendel, D. Wu, Assisting in making investment choices-investment judgment advisor (ija)-this chapter was written with the assistance of ms. jhiin joo, a ph, in: d. student working under the supervision of the authors, 2010.
[23] J.M. Mendel, D. Wu, Computing withwords for hierarchical and distributed decision-making, in: Computational Intelligence in Complex Decision Systems, Springer, 2010, pp. 233–271..
[24] M.M. Korjani, J.M. Mendel, Fuzzy love selection by means of perceptual computing, in: 2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013, pp. 766–770, https://doi.org/10.1109/IFSA-NAFIPS.2013.6608497.
[25] F. Liu, J.M. Mendel, Encoding words into interval type-2 fuzzy sets using an interval approach, IEEE transactions on fuzzy systems 16 (2008) 1503–1521.
[26] D. Wu, J.M. Mendel, S. Coupland, Enhanced interval approach for encoding words into interval type-2 fuzzy sets and its convergence analysis, IEEE Trans. on Fuzzy Sys. 20 (2011) 499–513.
[27] M. Hao, J.M. Mendel, Encoding words into normal interval type-2 fuzzy sets: Hm approach, IEEE Transactions on Fuzzy Systems 24 (2015) 865–879.
[28] J.M. Mendel, D. Wu, Perceptual reasoning for perceptual computing, IEEE Trans. on Fuzzy sys. 16 (2008) 1550–1564.
[29] D. Wu, J.M. Mendel, Aggregation using the linguistic weighted average and interval type-2 fuzzy sets, IEEE Transactions on Fuzzy Systems 15 (2007) 1145–1161.
[30] J.T. Rickard, J. Aisbett, R.R. Yager, G. Gibbon, Linguistic weighted power means: comparison with the linguistic weighted average, in: 2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011), IEEE, 2011, pp. 2185–2192.
[31] N.N. Karnik, J.M. Mendel, Centroid of a type-2 fuzzy set, information SCiences 132 (2001) 195–220..
[32] M. Nie, W.W. Tan, Ensuring the centroid of an interval type-2 fuzzy set, IEEE Transactions on Fuzzy Systems 23 (2014) 950–963.
[33] J.M. Mendel, A comparison of three approaches for estimating (synthesizing) an interval type-2 fuzzy set model of a linguistic term for computing with words, Granular Comp. 1 (2016) 59–69.
[34] J.M. Mendel, Computing with words and its relationships with fuzzistics, Information Sciences 177 (2007) 988–1006.
[35] N.N. Karnik, J.M. Mendel, Operations on type-2 fuzzy sets, Fuzzy sets and systems 122 (2001) 327–348.
[36] A. Bilgin, H. Hagras, J. van Helvert, D. Alghazzawi, A linear general type-2 fuzzy-logic-based computing with words approach for realizing an ambient intelligent platform for cooking recipe recommendation, IEEE Trans. on Fuzzy Systems 24 (2015) 306–329.
[37] A. Bilgin, H. Hagras, S. Upasane, A. Malibari, M.J. Alhaddad, D. Alghazzawi, An adaptive ambient intelligent platform for recommending recipes using computing with words, in: in: 2014 International Conference on Intelligent Environments, IEEE, 2014, pp. 372–373.
[38] A. Bilgin, H. Hagras, A. Malibari, D. Alghazzawi, J. Mohammed, A computing with words framework for ambient intelligence, in: 2013 IEEE International Conference on Systems, Man, and Cybernetics, IEEE, 2013, pp. 2887–2892..
[39] A. Bilgin, H. Hagras, A. Malibari, M.J. Alhaddad, D. Alghazzawi, Towards a linear general type-2 fuzzy logic based approach for computing with words, Soft Comp. 17 (2013) 2203–2222.
[40] A. Bilgin, H. Hagras, A. Malibari, M.J. Alhaddad, D. Alghazzawi, An experience based linear general type-2 fuzzy logic approach for computing with words, in: 2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2013, pp. 1–8.
[41] A. Bilgin, H. Hagras, A. Malibari, M.J. Alhaddad, D. Alghazzawi, A general type-2 fuzzy logic approach for adaptive modeling of perceptions for computing with words, in: UKCI, IEEE, 2012, pp. 1–8..
[42] A. Bilgin, H. Hagras, A. Malibari, M.J. Alhaddad, D. Alghazzawi, Towards a general type-2 fuzzy logic approach for computing with words using linear adjectives, in: FUZZ-IEEE, 2012, pp. 1–8..
[43] J.M. Mendel, The perceptual computer: the past, up to the present, and into the future, Informatik-Spektrum 41 (2018) 15–26.
[44] D. Dubois, H. Prade, Rough fuzzy sets and fuzzy rough sets, Int. J. General System 17 (1990) 191–209.
[45] T.Y. Lin, Granular computing: Fuzzy logic and rough sets, in: Computing with Words in Information/Intelligent Systems 1, Springer, 1999, pp. 183–200..
[46] T. Young, C.-J. Liau, Granular computing and rough sets, in: Data mining and knowledge discovery handbook, Springer, 2005, pp. 535–561..
[47] Z. Pawlak, Rough sets, International journal of computer & information sciences 11 (1982) 341–356.
[48] Z. Pawlak, J. Grzymala-Busse, R. Slowinski, W. Ziarko, Rough sets, Communications of the ACM 38 (1995) 88–95.
[49] J.M. Mendel, D. Wu, Determining interval type-2 fuzzy set models for words using data collected from one subject: Person fous, in: 2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2014, pp. 768–775.
[50] P. Gupta, Perceptual computing for power management, linguistic optimization and decision making, PhD Thesis (2019)..
[51] H. Bustince, E. Barrenechea, M. Pagola, J. Fernandez, Z. Xu, B. Bedregal, J. Montero, H. Hagras, F. Herrera, B. De Baets, A historical account of types of fuzzy sets and their relationships, IEEE Transactions on Fuzzy Systems 24 (2015) 179–194.
[52] J.M. Loomis, S.J. Lederman, Tactual perception, Handbook of perception and human performances 2 (1986) 2.
[53] V. Bruce, P.R. Green, M.A. Georgeson, Visual perception: Physiology, psychology, & ecology, Psychology Press, 2003.
[54] H. Becker, Computing with words and machine learning in medical diagnostics, Information Sciences 134 (2001) 53–69.
[55] C. Kahraman, I. Kaya, S. Çebi, Renewable energy system selection based on computing with words, International Journal of Computational Intelligence Systems 3 (2010) 461–473.
[56] J.M. Mendel, D. Wu, Challenges for perceptual computer applications and how they were overcome, IEEE computational intelligence magazine 7 (2012) 36–47.
[57] P.K. Gupta, J.A. Perez, Enhanced type-2 wang-mendel approach, To appear (2022)..
[58] N. Díaz-Rodríguez, A. Lamas, J. Sanchez, G. Franchi, I. Donadello, S. Tabik, D. Filliat, P. Cruz, R. Montes, F. Herrera, Explainable neural-symbolic learning (x-nesyl) methodology to fuse deep learning representations with expert knowledge graphs: The monumai cultural heritage use case, Information Fusion 79 (2022) 58–83.
[59] K. Intanssov, Intuitionistic fuzzy set, Fuzzy Sets and Systems 20 (1986) 87–96.
[60] K. Atanassov, Review and new results on intuitionistic fuzzy sets, preprint IM-MFAIS-1-88, sofia 5 (1988) l..
[61] K.T. Atanassov, More on intuitionistic fuzzy sets, Fuzzy sets and systems 33 (1989) 37–45.

[62] K.T. Atanassov, Intuitionistic fuzzy sets: Theory and applications (studies in fuzziness and soft computing), 35, Physica-Verlag, Heidelberg, 1999.

Prashant Kumar Gupta is currently working as Artificial Intelligence Researcher at German Research Center for Artificial Intelligence (DFKI GmbH), Saarbrücken, Saarland, Germany. He received the B.Tech. and M.Tech. degrees in 2008 and 2012, respectively, both from Guru Gobind Singh Indraprastha University, India and the PhD degree in 2019 from South Asian University, India. He has contributed research publications in various journals and core A-ranked conferences such as IEEE Transactions on Fuzzy Systems, Information Sciences, Fuzzy Sets and Systems, Applied Soft Computing, Granular Computing, FUZZ- IEEE, IEEE-IJCNN, and IEEE SMC. He is serving as a reviewer in various SCI and SCIE journals of international repute like Information Sciences, Elsevier, Expert Systems with Applications, Elsevier, Computers & Industrial Engineering, Elsevier, Knowledge-Based Systems, Elsevier, Applied soft computing, Elsevier, Electronic Commerce Research and Applications, Elsevier, Egyptian Informatics Journal, Elsevier, Complex & Intelligent Systems, Springer, etc. His research interests include fuzzy systems, computing with words, artificial intelligence, explainable artificial intelligence, etc.

Javier Andreu-Perez (Senior Member, IEEE) received his PhD degree in Intelligent Systems in 2012 from Lancaster University (United Kingdom). He is currently a Senior Lecturer and the Chair of the Centre for Computational Intelligence, Smart Health Technologies Group, University of Essex, U.K. He has published his research in prestigious journals from IEEE, Nature, Springer, and Elsevier publishers, mostly in top artificial intelligence and neuroscience journals. His work in artificial intelligence and biomedical engineering has attracted nearly 3000 citations. His primary expertise is in employing and developing novel artificial intelligence methods for neuro/bioengineering, health, and industrial informatics. His research interests include deep learning, fuzzy systems, and human-centred artificial intelligence, particularly developing novel methods focusing on trustworthy and explainable AI to work along with humans. He has also been recipient of distinguished individual fellowships, such as the Talentia Senior Fellowship and Invitational Fellowship for Japan Society for the Promotion of Science (JSPS). He has been the primary investigator of projects funded by the U.K. research councils, international organizations, and corporations. He also contributes as an Associate Editor-in-Chief of the Journal Neurocomputing (Elsevier) and other editorials from other prestigious journals in computational intelligence and emerging technologies.