A Computing with Words path from fuzzy logic to natural language *

Claudio Moraga Enric Trillas

European Centre for Soft Computing, 33600 Mieres, Asturias, Spain
{claudio.moraga, enric.trillas}(at) softcomputing.es

Abstract *– Linguistic variables are reviewed and some aspects of the original definition revisited, showing that they provide a first formal frame to do computing with words in a "narrow sense". A two-levels hierarchy of languages is discussed in order to move to simple cases of computing with words in natural language, supported by linguistic variables.*

Keywords – Computing with words, linguistic variables, meaning modifiers

1 Introduction

In the late 90's, L.A. Zadeh introduced the idea of *computing with words* [1]. Much efforts have been done ever since to face the challenge and to develop the required proper concepts, models and methods to turn the bright initial idea into a sound area of research. Fuzzy logic seems predestined to be an appropriate formal frame to support the above efforts. Among other things, because fuzzy if-then rule bases may be understood as a specification of (the model of) a system in a language close to natural language and each rule, as the specification of a computation based on words, even though at the moment of doing real computations, this is done at the level of numbers. At this time, fuzzy if-then rule bases may be considered as a metaphor of what we would like to achieve by *really* computing with words. In this sense, what is currently known as fuzzy logic may be considered as a particular case of computing with words.

This paper shows, that in the context of fuzzy linguistic variables there are already processes that are representative of computing with words, albeit possibly in a "narrow sense". Furthermore it will be shown that it is possible to build a hierarchical relationship between words based on linguistic variables at the level of fuzzy logic and "words" – (actually short sentences)- at the level of natural language; to move from linguistic variables to "linguistic expressions" and from linguistic modifiers to "meaning modifiers". In analogy to the fact that linguistic modifiers, when applied to linguistic variables constitute possibly the simplest case of computing with words, it will be shown that "meaning modifiers", when applied to "linguistic expressions" may lead to non-trivial linguistic results of non-negligible complexity, in what may be thought of, as computing with "words".

The rest of the paper provides first a set of needed definitions to continue with a discussion about computing with words in fuzzy logic. An extension to computing with "words" in natural language is analyzed in the closing section.

* This work has been partially supported by the Foundation for the Advancement of Soft Computing (Asturias, Spain) and CICYT (Spain) under project TIN2008-06890-C02-01

2 Definitions

Definition 1 [1]:
Computing with words and perceptions, or CWP for short, is a mode of computing in which the objects of computation are words, propositions and perceptions described in a natural language.

Definition 2:
Let Ω be a set of (non-ambiguous) words that are used in a given context. The elements of Ω are considered to be pairs (representation, meaning). Representations may be taken from different syntactic domains, but meanings, from a single semantic domain. Moreover, let Γ be a finite set of functions $\{\gamma_1, \gamma_2, \ldots, \gamma_k\}$, with

$$\gamma_i : \Omega^n \rightarrow \Omega$$

The functions $\{\gamma_1, \gamma_2, \ldots, \gamma_k\}$ must be *designed* in such a way, that applied to words of Ω will produce reasonable words of Ω for the given context.

Definition 3:
In this paper when speaking of computing with words, the *objects* of computation are specified in definition 1. The *agents* of computation are functions taken from Γ, as stated in definition 2.

Examples: Let Ω be a subset of English.
If γ_1 is an appropriate "association" function. Then
γ_1(parent, children) = family
If γ_2 is a synonym function. Then
γ_2(pretty) = beautiful
If γ_3 is an antonym function. Then
γ_3(near) = far

These examples show that since the functions in Γ represent some "linguistic transformation", their interpretation should be taken take in account in the line of language precisiation (*à la* Zadeh).

At this stage, it should be mentioned that not under the scope of "computing with words", but simply with the goal of designing good fuzzy if-then rules, work has been done to select or build the most convenient operations connecting premises and the conclusion [2], [3]. In the context of this paper, the given references correspond to a basic contribution in the design of the most convenient functions for the corresponding Γ.

ISBN: 978-989-95079-6-8

3 Computing with words in fuzzy logic

One of the key concepts in the development of fuzzy logic is that of a *linguistic variable*, introduced by L.A. Zadeh in 1975 [4]. The value set of a linguistic variable is a set of words –"linguistic terms"– denoting predicates or linguistic labels, and the syntactic domain of those words is constituted by appropriate convex fuzzy sets. "*… in general, a linguistic variable is associated with two rules: (1) a syntactic rule, which may have the form of a grammar for generating the names of the values of the variable; and (2) a semantic rule which defines an algorithmic procedure for computing the meaning of each value"* (See section 2 of Part II of [4])

In what follows, this idea is extended to include a formal grammar and a data base. The grammar will control possible transformations on the fuzzy sets representing the linguistic terms of a linguistic variable. The effect of linguistic modifiers (originally called "hedges of type I" [5]) as well as the generation of antonyms of linguistic terms (which was studied much later [6], [7], [8]) may, for instance, be formalized in terms of productions of a grammar. The data base, on the other hand, is meant to contain the meanings associated to the linguistic terms and their considered possible modifications through the grammar.

It is easy to see that the grammar and the data base have a functionality which is subsumed by that of the functions presented in Definition 2.

For the representation of linguistic terms as fuzzy sets, if no further information is available, a trapezoidal shape is appropriate since an agreement among users about the subset of the universe where the predicate is indeed valid, defines the core of the fuzzy set and the agreement about the subset of the universe where the predicate is not at all valid, defines its co-support. Our human experience with predicates indicates that the transitions between co-support and core are continuous and monotone. Therefore, if no further information is available, linear transitions leading to a piecewise linear convex normalized fuzzy set is an appropriate first choice. If the core contains only one point, then the trapezium reduces to a triangle. Notice that the above analysis does not require the trapezium (triangle) to be symmetric.

Trapezoidal fuzzy sets are simple to represent as an ordered quadruple of the ordinates of their four corners (t_1, t_2, t_3, t_4). The interval between the first and the fourth corner represents the support of the fuzzy set. (If non-normalized trapezoidal fuzzy sets are considered, then a preceding scaling factor is needed.)

Another basic assumption, normally associated to linguistic variables, is that all linguistic terms of the variable constitute a partition of unity, *i.e.*, at every point of the universe of discourse, the membership degrees of the linguistic terms add up to 1.

Definition 4:

A linguistic modifier is a unary function in the set of fuzzy sets representing linguistic terms. A linguistic modifier LM is *compressing* if for any fuzzy set A representing a linguistic term, $LM(A) \subset A$, and is *expanding* if $A \subset LM(A)$. There are however linguistic modifiers that are neither compressing nor expanding.

In what follows, to keep the notation as simple as possible, the label of a linguistic term will also be used to denote the membership function of the fuzzy set representing it.

Let $T := (t_1, t_2, t_3, t_4)$

$LM(T) := ((t_1+t_2)/2,\ t_2,\ t_3,\ (t_3+t_4)/2)$ is a compressing modifier and in what follows will be given the meaning *more_strictly*(T).

$LM(T) := (t_1, (t_1+t_2)/2, (t_3+t_4)/2, t_4)$ is an expanding modifier and in what follows will be given the meaning *roughly*(T).

Lemma 1:

Let $T := (t_1, t_2, t_3, t_4)$

Define $T_{left} := ½(t_1, (t_1+ t_2)/2, (t_1+ t_2)/2, t_2)$

$T_{right} := ½(t_3, (t_3+ t_4)/2, (t_3+ t_4)/2, t_4)$

Then:

$$T = T_{left} + more\_strictly(T) + T_{right}$$

Proof: By adding point-to-point the piecewise linear segments.

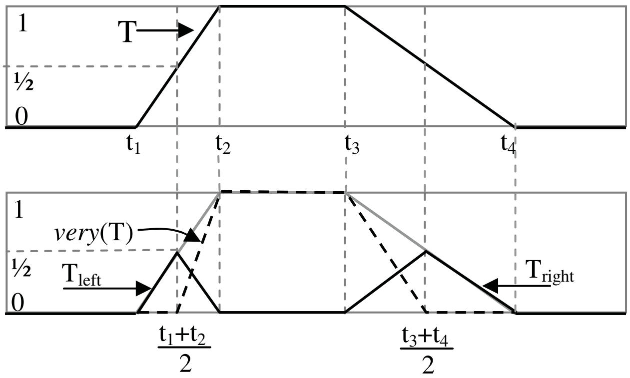

Figure 1**:** The relationship among the fuzzy sets T, *more_strictly*(T), T_{left} and T_{right}

It is easy to see that for all $x < t_3$, T_{right} takes the value 0. Moreover, since $(t_1 + t_2)/2$ is equidistant of t_1 and t_2, then at this point, T takes the value ½. (See Figure 1). Moreover between t_2 and t_3, both T_{left} and T_{right} have membership degree 0, meanwhile T and *more_strictly*(T) have membership degree 1, therefore in this interval the claim trivially holds. Table 1 summarizes the relevant membership values at t_1, $(t_1 + t_2)/2$ and t_2.

Table 1: relevant membership values

	t_1	$(t_1 + t_2)/2$	t_2
T	0	½	1
more_strictly(T)	0	0	1
T_{left}	0	½	0
more_strictly(T) + T_{left}	0	½	1

In the interval $[t_1, t_2]$, both *more_strictly*(T) and T_{left} are piecewise linear; therefore their sum will also be piecewise linear. In the subinterval $[t_1, (t_1+t_2)/2]$, since the membership value of *more_strictly*(T) is 0, the sum will be given by T_{left}, which has a representation as a straight line segment with slope $1/(t_2-t_1)$. In the subinterval $[(t_1+t_2)/2, t_2]$, both *more_strictly*(T) and T_{left} have a representation as straight line segments, therefore also their sum. It is simple to see

ISBN: 978-989-95079-6-8

from Figure 1, that the sum segment runs from the point $\langle (t_1+t_2)/2, 1/2 \rangle$ to the point $\langle t_2, 1 \rangle$, with slope $1/(t_2-t_1)$. Since both sum segments share a common point and have the same slope, they constitute a single straight line segment with slope $1/(t_2-t_1)$ in the whole interval $[t_1, t_2]$. This corresponds with the left part of T. A similar analysis will prove that in $[t_3, t_4]$ the sum of *more_strictly*(T) plus T_{right} returns the corresponding part of T. For reasons that will become clear below, no attempt will be done here to associate a meaning to T_{right} and T_{left}.

Corollary 1.1:
Assume that T is a rectangular trapezium. Without loss of generality let $T := (t_1, t_2, t_3, t_3)$. Then :

$T = T_{left} +$ *more_strictly*(T)

since T_{right} has zero-support.

Example 1:
Consider a linguistic variable with three linguistic terms building a partition of unity, as shown in Figure 2(a). Furthermore, let it be assumed that some experiments done using this representation –(model)- of the variable, made advisable a refinement of the middle linguistic term, and that this is done by applying Lemma 1 (with $t_2 = t_3$). This is shown in Figure 2(b). Solving the superposition of the small triangles with the respective trapeziums leads to a new model of the linguistic variable, as shown in Figure 2(c).

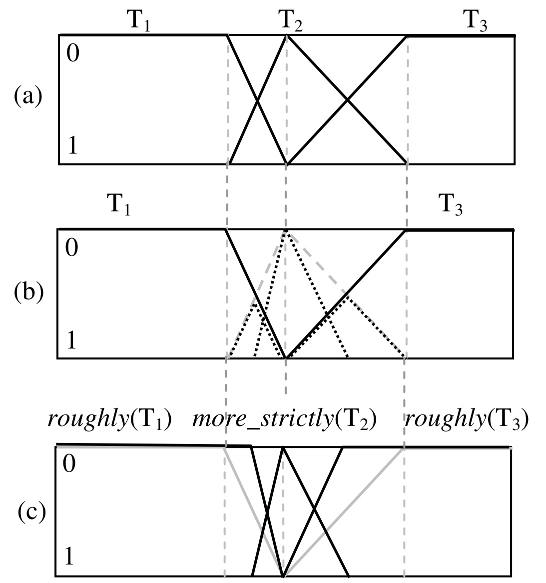

Figure 2**:** (a) Initial model of the linguistic variable
(b) Effect of applying Lemma 1 to T_2
(c) The resulting new model for the linguistic variable

An analysis of the process shows that T_2 changed into *more_strictly*(T_2). Furthermore, the core of T_1 was expanded by one half of the support corresponding to the right wing of the trapezoidal representation. From Definition 4 follows that it represents *roughly*(T_1). Similarly, T_3 turns into *roughly*(T_3). In summary, a new representation of the linguistic variable, which preserves the partition of unity, was obtained through a refinement process by using appropriate linguistic modifiers. This represents (possibly the simplest) elementary operations in the context of computing with words.

Case study 1:
In the context of linguistic variables with structure of a partition of unity, *two* consecutive linguistic terms T_j and T_{j+1} will be considered, which will be refined under Lemma 1.

Let $T_j := (t_1, t_2, t_3, t_4)$
and $T_{j+1} := (t_3, t_4, t_5, t_6)$
Lemma 1 generates:

more_strictly(T_j) $:= ((t_1+t_2)/2, t_2, t_3, (t_3+t_4)/2)$
$(T_j)_{right} := ½(t_3, (t_3+t_4)/2, (t_3+t_4)/2, t_4)$
more_strictly(T_{j+1}) $:= ((t_3+t_4)/2, t_4, t_5, (t_5+t_6)/2)$
$(T_{j+1})_{left} := ½(t_3, (t_3+t_4)/2, (t_3+t_4)/2, t_4)$

(Lemma 1 also generates $(T_j)_{left}$ and $(T_{j+1})_{right}$, which are however not in the focus of this case study).
Notice that $(T_j)_{right} = (T_{j+1})_{left}$ therefore
if $T' = (T_j)_{right} + (T_{j+1})_{left}$ then

$T' := (t_3, (t_3+t_4)/2, (t_3+t_4)/2, t_4)$

It is easy to see that T' is an isosceles triangle. To T' the following *specification of meaning* can be associated:

T' = *between*(*more_strictly*(T_j), *more_strictly*(T_{j+1}))

and, the other way around, the operation *between* is defined as the structure of T'. See Figures 3(a) and 3(b).

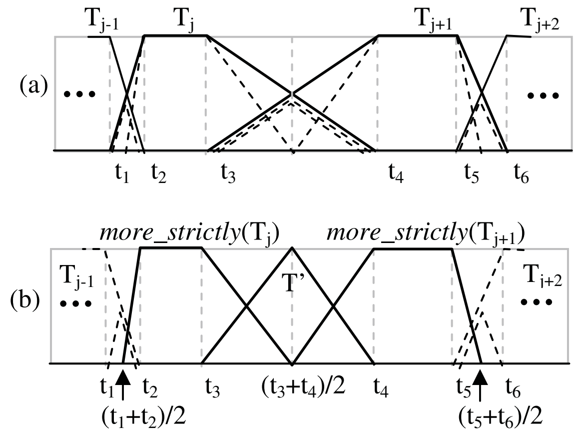

Figure 3**:** (a) Two neighbor linguistic terms and their refinement with Lemma 1 (dotted).
(b) Effect of the superposition of the overlapping side triangles generated by the refinement.

From T' = *between*(*more_strictly*(T_j), *more_strictly* (T_{j+1})) it may be concluded that there exists a $\gamma \in \Gamma$ such that

$\gamma =$ *between* $\circ$ (*more_strictly* $\times$ *more_strictly*)

and $T' = \gamma(T_j, T_{j+1})$

Notice that if all linguistic terms of a linguistic variable are refined with Lemma 1, then all "new" linguistic terms may be obtained with the composition of *between* and the Cartesian product *more_strictly* $\times$ *more_strictly*. A refinement of all linguistic terms after Lemma 1, may be related to wavelets [9], but almost duplicates the number of linguistic terms. Partial refinements, however, were studied in [10] under the scope of a metasemantics.

ISBN: 978-989-95079-6-8

Before studying a next case, two elementary operations will be introduced.

Definition 5:
Let $T_j := (t_1, t_2, t_3, t_4)$
Then, the operation *less than or equal to*, abbreviated "LE" is defined for all x in the universe of discourse, as:

$$(LE(T_j))(x) = \begin{cases} 1 & \text{if} \quad x \leq t_3 \\ T_j & \text{if} \quad x > t_3 \end{cases}$$

Similarly, the operation *more than or equal to*, abbreviated "ME" is defined as:

$$(ME(T_j))(x) = \begin{cases} 1 & \text{if} \quad x \geq t_2 \\ T_j & \text{if} \quad x < t_2 \end{cases}$$

Case study 2:
Consider a linguistic variable with more than 3 linguistic terms and refine some linguistic term in the middle. What will be the effect upon its non-refined neighbors?

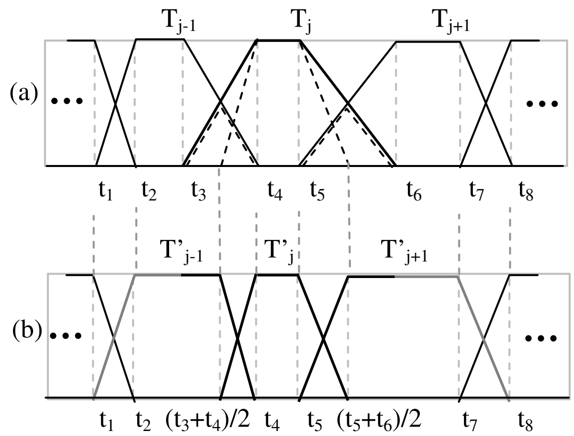

Figure 4 **:** (a) A linguistic variable with the prepared refinement of the j-th linguistic term.
(b) Effect of refining T_j upon its neighbors.

Figure 4 shows the linguistic terms before and after refining the j-th term. From Lemma 1 follows that:

$T'_j = more\_strictly(T_j)$

An interpretation of T'_{j+1} is not straight forward. Fig. 4(b) shows however that at the left hand side, the core of T'_{j+1} increased by one half of the support of the left wing, *i.e.*, by $\frac{1}{2}(t_6-t_5)$, with respect to the original core of T_{j+1}. Notice that this is the same (partial) effect caused by the operation *roughly*(T_{j+1}); however, the right hand side of T'_{j+1} does not change as compared to T_{j+1}. This may be given a formal expression as follows, for all x in the universe of discourse:

$(T'_{j+1})(x) = minimum(roughly(T_{j+1}), LE(T_{j+1}))$

or "in words",

$T_{j+1} = roughly(T_{j+1})$ *and* $LE(T_{j+1})$

where *and* is realized with the operation *minimum*.

Similarly,

$(T'_{j-1})(x) = minimum(roughly(T_{j-1}), ME(T_{j-1}))$

or "in words",

$T_{j-1} = roughly(T_{j-1})$ *and* $ME(T_{j-1})$

Example 1 and the two above cases show that at the lowest level, that of terms of linguistic variables, transformations may be formalized, which represent the possibly simplest cases of computing with words in narrow sense.

4 First steps towards natural language

When working with formal languages, "words" are build with "symbols" of an alphabet. Quotations are being used, since the "symbols" may be words of a reference language and, consequently the "words" will (normally) be (meaningful) sentences of the reference language. This model of a hierarchy of languages will be used in an attempt to transfer or imitate the elementary operations discussed above to short statements in natural language. These "words" will be written in square brackets. Furthermore the notation "linguistic statement" instead of linguistic variable, and "meaning modifier" instead of linguistic modifier will be used. Finally it becomes apparent that "words" will no longer be just pairs (representation, meaning), but at least triples (representation, structure, meaning). Representation is concerned with the rules of writing, meanwhile the structure refers to the grammatical ordering of the reference words to support a meaning.

Example 3:
Consider the "word" [*The weather is improving*], where the linguistic statement has the value set {hardly a bit, slightly, steadily, strongly}. How could the meaning of this "word" be emphasized?

i) *Use* the "word" consistently within the value subset {steadily, strongly}

ii) Design a meaning modifier MM_1 which integrates explicitly in the "word" the desired predicate from the value set, preserving the structure:

MM_1 (*steadily*, [*The weather is improving*]) =
[*The weather is steadily improving*]

It may be observed that this modifier changes the representation of the "word", preserves the grammar of the "word" and emphasizes its meaning.

iii) Design a meaning modifier MM_2 which integrates an additional reference word, taken from a pre-defined "subset of emphasizers" of the corresponding natural language, preserving the structure, to produce an enhancement:

MM_2 (*indeed*, [*The weather is improving*]) =
[*The weather is indeed improving*]

Notice that it is only claimed that both [*The weather is steadily improving*] and [*The weather is indeed improving*]

ISBN: 978-989-95079-6-8

emphasize the meaning of [*The weather is improving*], but not that they are synonyms.

Example 4:
Consider the "word" [*Yesterday it rained*]. A value set for the linguistic statement may be thought of as $D_1 \times D_2 \times D_3$, where D_1 = {in the morning, at midday, in the afternoon, in the evening, at night}, D_2 = {a couple of minutes, several hours, the whole day} and D_3 = {a few drops, moderately, cats-and-dogs}. Notice that D_1 and D_2 may be associated both to *yesterday* and to *rain*, meanwhile D_3 is clearly related only to *rain*.

i) It is requested that [*Yesterday it rained*] be emphasized. The *rained*-component may be emphasized in a way similar to that discussed in example 3. Can *yesterday* be emphasized? At the level of natural language, it is possible to design a 2-place meaning modifier MM_1 giving:

MM_1([*December the 1st*], [*Yesterday it rained*]) = [*Yesterday December the 1st it rained*],

which may be considered as a possible way of emphasizing *yesterday*.

ii) In the context of [*Yesterday it rained*] D_1, D_2 and D_3 may be seen as value sets of the linguistic variables "when", "how_long" and "how_much" (at the reference level), respectively. As linguistic variables, they have grammars that, among other things, may change number, shape and distribution of the fuzzy sets representing the corresponding linguistic terms, as well as a database keeping track of their meanings. Assume that the corresponding grammars are G_1, G_2 and G_3. Let $r_{2,i}$ and $r_{3,j}$ represent the i-th and j-th rules of G_2 and G_3, respectively, such that:

$r_{2,i}(D_2)$ = {a couple of minutes, *half an hour*, several hours, the whole day}

$r_{3,j}(D_3)$ = {a few drops, moderately, *a lot*, cats-and-dogs}

Some elements of the new domains may have the same labels as in the original domains, but will possibly be different since at least some of them are shifted and compressed, because in both cases, a new linguistic term was introduced. (Notice here the importance of the data base with meanings).
Then it is possible to use linguistic modifiers at the low level to build meaning modifiers MM (at the upper level) to generate new, richer "words" as, for instance,

MM[*Yesterday it rained*] =
[*Yesterday it rained quite a lot about half an hour almost at midday*]

It may be observed that the meaning modifier reordered the domains as $r_{3,j}(D_3) \times r_{2,i}(D_2) \times D_1$, made use of the new generated linguistic terms "half an hour" and "a lot", applied the linguistic modifiers "quite" (to "a lot") "about" (to "half an hour") and "almost" (to "at midday"). Fuzzy logic offers sound methods to perform all these reference level operations, which put together, formally specify the meaning modifier.

Analysis of the meaning modification
[*quite a lot about half an hour almost at midday*]:

Recall that the value set for [*Yesterday in rained*] was specified as the Cartesian product of three domains. Therefore, its "value" is a triple, whose components are possibly modified linguistic terms within the prevailing versions of the corresponding domains. Therefore, each one of the linguistic terms may be analyzed separatedly and the required linguistic modifiers, *designed* to satisfy the requirements of the use of the language.

i) Design of *quite*, to produce the modified linguistic term *quite a lot*.
It should be noticed that *quite* is a rather complicated linguistic modifier, since it is very context dependent: in American English it seems to be used mostly to emphasize, in a way similar to *more_strictly*, meanwhile in British English it may also be ironically used to mean exactly the contrary. Furthermore, *quite* is also used as an idiom, like in "*quite* a few", possibly meaning the same as "*quite* a lot". In the case under analysis however, it seems that *quite* is intended to be used to suggest rather more precisely, what is expressed by "a lot" alone. If this is the case, then a compressing linguistic modifier should be used, but with possibly a smaller amount of compression than in the case of *more_strictly*.
A general analysis leading to a proposal for a prototype of *quite* is shown in Figure 5.

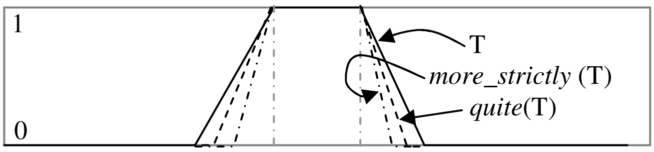

Figure 5: A possible prototype for *quite* as a linguistic modifier, which is less compressing than *more_strictly*

Observations on the *use* of *quite* in the context where it will be applied and in similar contexts will be needed to properly tune this modifier. Notice that it may well be the case that the contexts at the reference level and at the natural language level pose different constraints for the design of the modifier.

ii) Design of *about*, to produce the modified linguistic term *about half an hour*.
In the use of the language, *about* half an hour, appears to have a meaning close to that of *around* half an hour or *roughly* half an hour. Experiences with numerical approximation with fuzzy numbers indicate that *around* is an expanding modifier that preserves the core –(which in this case contains only one point, since the fuzzy set is triangular)– and increases the support. It seems reasonable to extend this modifier to trapezoidal fuzzy sets in the same way. On the other hand, *roughly*, as defined earlier, expands the core and preserves the support. If the meaning of *about* half an hour is indeed close to that of *around* half an hour and *roughly* half an hour, then the corresponding modified fuzzy sets should exhibit a high degree of "family resemblance" [11] and this can be used as an additional guiding constraint to design *about*.
A proposal for a prototype of *about* is illustrated in Figure 8 with respect to some abstract T (to more clearly show the effect of the (known) linguistic modifiers and of the proposed prototype).

ISBN: 978-989-95079-6-8

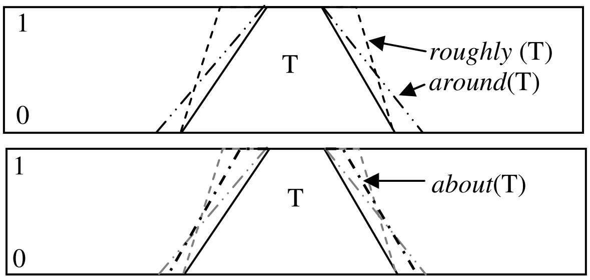

Figure 6: A possible prototype for *about*, for some abstract T, as related to *roughly* and *around*

As in the former case, observations on the *use* of the modifier in the target context will be needed to properly tune it.

iii) Design of *almost*, to produce the modified linguistic term *almost at midday*.
The use of the language, *e.g.* "the glass is *almost* empty" or "Peter was *almost* crazy with so many problems", indicates that this is a one-sided shifting and possibly expanding modifier, which applied to a fuzzy set representation of a linguistic term, displaces and possibly expands the fuzzy set towards the "less than" side. (See Figure 7). It is simple to see that *almost at midday* and *around midday* will have some "family resemblance", but to a clear weaker degree than the one discussed in the former paragraph.

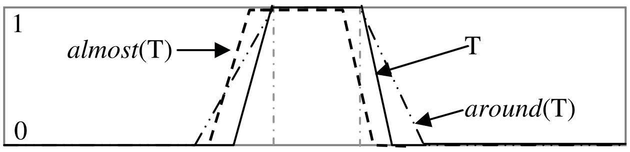

Figure 7: A possible prototype for *almost*, as compared to *around*, applied to some abstract T

Observations on the *use* of all above modifiers in the actual target context, including experimentation at the level of the language, will be needed to ensure their empirical correctness. See, for instance [12], [13], [14].

5 Conclusions

Within the formalism of linguistic variables in fuzzy logic, linguistic modifiers represent a very simple case of computing with words. Some new elementary operations have been defined, that allow new non trivial instances of low level computing with words. The paper shows that it is possible to extend these ideas to "linguistic statements" (short sentences) and "meaning modifiers" in natural language. The "linguistic statements" become the new "words" (as is the case of programming languages when considered as context free formal languages). This is done however keeping a solid base on linguistic terms and linguistic modifiers. The hierarchy of languages suggests that the more about computing with words is learnt in fuzzy logic, the more can be projected to natural languages to do computing with "words".

Acknowledgement

The authors thank the reviewers for their constructive criticism and the suggestions for the improvement of the paper.

References

[1] L.A. Zadeh, From computing with numbers to computing with words. From manipulation of measurements to manipulation of perceptions. *IEEE Trans. Circuits and Systems* **45** (1): 105-119, 1999

[2] C. Moraga, A. Pradera and E. Trillas, Evolutionary tuning of fuzzy if-then rules at the level of operations: A proposal. *Proc. III Congr. Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados MAEB'2003*, 530-537. Press University of Oviedo, 2003

[3] E. Trillas, On some features of fuzzy logic. *Proc. 27th Annual Meeting of the North American Fuzzy Information Processing Society NAFIPS.* 2008

[4] L.A. Zadeh, The concept of linguistic variable and its application to approximate reasoning. *Information Sciences* **8**: 199-249; 301-357; **9**: 43-80, 1975

[5] L.A. Zadeh, A fuzzy-set-theoretic interpretation of hedges. Memorandum M-335, Electronics Research Laboratory, University of California, Berkeley, 1972

[6] E. Trillas and T. Riera, Towards a representation of synonyms and antonyms by fuzzy sets. *Busefal* V:30-45, 1981

[7] E. Trillas, S. Cubillo and E. Castiñeira, On antonymy from fuzzy logic. *Proc. 9th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge Based Systems (IPMU'00)*, 79-84, 2000

[8] E. Trillas , C. Moraga, S. Guadarrama, S. Cubillo and E. Castiñeira, Computing with Antonyms. *Studies in Fuzziness and Soft Computing* **217**: 133-153, Springer, 2007

[9] A. Rodriguez de Soto, Building a hierarchical set of fuzzy rules using Wavelet Theory. *Proc. 9th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge Based Systems (IPMU'00)*, 1764-1769, Madrid, 2000.

[10] C. Moraga, A metasemantics to refine fuzzy if-then rules. *Proc. IEEE Int. Symp. on Multiple-valued Logic*, 148-153. IEEE-CS-Press, 2004

[11] E. Trillas, C. Moraga and A. Sobrino, On "family resemblances" with fuzzy sets. *Proc. IFSA'2009*, IFSA-Press, 2009

[12] J. Hellendorn, Reasoning with Fuzzy Logic. Dissertation, Delft University of Technology, The Netherlands, 1990

[13] V. Novak, A horizon shifting model of linguistic hedges for approximate reasoning, *Proceedings of the 5th IEEE International Conference on Fuzzy Systems,* vol. 1, 423–427, IEEE-CS-Press, 1996

[14] P.J. Macvicar-Whelen, Fuzzy sets, the concept of height, and the hedge 'very'. *IEEE Trans. On Systems, Man, and Cybernetics*, SMC-8, 507-511, 1978

ISBN: 978-989-95079-6-8