Fuzzy Linguistic Summaries: Where Are We, Where Can We Go?

Bernadette Bouchon-Meunier, Gilles Moyse

To cite this version:

Bernadette Bouchon-Meunier, Gilles Moyse. Fuzzy Linguistic Summaries: Where Are We, Where Can We Go?. 2012 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr), Mar 2012, New-York, United States. pp.317-324, 2012, <10.1109/CIFEr.2012.6327810>. <hal-00932854>

HAL Id: hal-00932854

https://hal.archives-ouvertes.fr/hal-00932854

Submitted on 5 Jan 2015
Fuzzy Linguistic Summaries: Where Are We, Where Can We Go?

Bernadette Bouchon-Meunier, Fellow, IEEE, and Gilles Moyse

Abstract— Along with the increase of the amount of data stored and to be analyzed, different techniques of data analysis have been developed over the years. One of them, the linguistic summary, aims at summing up large volume of data into simple sentences.

In this paper, we present an overview of two main streams of research, namely fuzzy logic based systems and natural language generation, covering the methods designed to work with numerical data, time series, or simple labels (enumerations). We focus on the former stream and we give some hints to go further on fuzzy quantifiers.

I. INTRODUCTION

VARIOUS techniques of linguistic data summaries have been developed over the past years in order to cope with the exponential quantity of data created and stored. The financial sector is especially concerned with this large amount of data since it needs to process and sum up information quickly and accurately in order to make the right decision on time.

In this paper, we discuss linguistic representations of numerical input data. It does not take into account tasks of text, images or videos summarization. First, we propose an overview of existing approaches to linguistic data summarization. Then we focus on the use of quantifiers for fuzzy linguistic summaries. Finally, we present some results obtained using these quantifiers.

In the first part, we present the two main options that have been developed and used in order to perform such linguistic summaries: the first one is based on fuzzy logic, the second one on Natural Language Generation (NLG) systems. This presentation is supported by several examples and financial ones more specifically. We underline the fact that, even though the two approaches share the same concerns and goals, they share their techniques.

In the second part, we propose an analysis of solutions to represent and manage fuzzy linguistic quantifiers, as a first step to a more general attempt to move closer fuzzy logic based summarization and natural language, or at least to obtain fuzzy logic summaries closer to expert linguistic descriptions.

In the third part, these quantifiers are implemented and used in the context of an online bookseller database. We calculate the truth values of marketing sentences, and thus demonstrate how linguistic summaries can be used in a business environment.

II. COMPARATIVE STUDY OF EXISTING APPROACHES

In this part, we briefly recall the main classes of solutions which have been implemented to address the summary generation question.

The different techniques presented here are linked to systems which have been implemented, most of the time regarding financial issue. Table I summarizes these systems, their principles and main applications.

There are two main ways for automatic production of linguistic summaries, one using fuzzy logic tools, the other adopting Natural Language Generation (NLG) techniques. [1] suggests they both could learn from each other. These two branches are successively presented below.

A. Protoform

On the one hand, the fuzzy logic community proposes tools using protoforms (based on Yager’s and Zadeh’s proposals [2], [3]) like “QX are A”, where Q is a quantifier, such as “most”, A is a possible value of an attribute, such as “high”.

Few stocks have performed well this year” is an instance of a sentence based on this protoform.

They have been extended in [4] and [5] with “QBX are A” (ex: “Most of the green stocks have performed well this year”), “QX are A and B” (ex: “Some stocks are eco related and have performed well this year”). For its part, [6] also added “X are C1 and C2 and C3…” (ex: “Some stocks are eco related and have performed well this year and are start ups and have received large amounts of capital”).

Other protoforms are introduced, based on fuzzy dependencies, like “most stocks in the same kind of sector have performed the same way this year” in [7], [8], or based on gradual rules like “the higher the benefit, the higher the dividend” in [8], [9].

Systems based on these protoforms have been implemented, like FQuery [10], [11], Quantirius [12] and Summary SQL [8]. Some of them have been used to generate financial summaries, like [5] which allows the evaluation of sentences like “Slowly decreasing trends that took most of the time are of a large variability” or Quantirius which was used in [12] to summarize a shares quotations database.

B. Concept Hierarchies

Conceptual trees [13], implemented in the SaintEtiQ system [14–16], also use fuzzy logic tools, but present results differently. They are not displayed as natural language sentences but as a hierarchy of concepts, where the most general one is the root, and the most detailed ones are the leaves. Another option, presenting results in ISA hierarchies, is introduced in [17], but does not seem to have been developed as extensively as SaintEtiQ has.
C. Alternative Approaches

Other techniques have been developed to find more concise representations of data. Nevertheless, they do not yield real linguistic sentences, and, as far as we know, have not been implemented as completely as the two aforementioned. For instance, fuzzy rule extraction, described in [18], where fuzzy relations between data can be established, provides a kind of non-linguistic summarization considering that the similar data can be removed (and then summarized). The same remark applies to generalization, like in [19]. Here, the data is summed up by generalizing it, i.e., finding categories that encompass it. In that way, this approach is closer to clustering than to linguistic summarization.

D. Natural Language Generation approach

A second type of summaries is obtained through Natural Language Generation (NLG) methods. The main differences in terms of yielded results, is text generation, more sophisticated and longer (more than one sentence) with NLG, and data processing part, less detailed with NLG. For instance, the data selection rules are usually hard-coded or based on expert systems, hence less flexible than the fuzzy logic techniques.

Such systems are, for instance, Yseop [20], SumTime Mousam [21], SumTime Turbine [22] EasyText [23], FOG.

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>SYSTEMS AND THEIR APPLICATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fuzzy logic based systems</strong></td>
<td></td>
</tr>
<tr>
<td>FQuery</td>
<td>- Fuzzy summarization, allowing direct drawing of fuzzy membership functions, and fuzzy computations over the data contained in the Access database.</td>
</tr>
<tr>
<td>Quantirius</td>
<td>- &quot;Interactive system supporting the mining and the assessment of linguistic summaries in a database.&quot;</td>
</tr>
<tr>
<td>Summary SQL</td>
<td>- Query language aimed at finding fuzzy functional dependencies and gradual functional dependencies.</td>
</tr>
<tr>
<td>SaintEtiQ</td>
<td>- Summarizes information and returns a concept tree, with the most general concept at its root, and the most detailed ones in the leaves.</td>
</tr>
<tr>
<td>Custom systems</td>
<td>- &quot;Truth values obtained for extended form summaries&quot;</td>
</tr>
<tr>
<td></td>
<td>- Production of a natural language report composed by sentences extracted from the model</td>
</tr>
<tr>
<td><strong>NLG systems</strong></td>
<td></td>
</tr>
<tr>
<td>SumTime Mousam</td>
<td>- Generic techniques to produce textual summaries of time series data</td>
</tr>
<tr>
<td>SumTime Turbine</td>
<td></td>
</tr>
<tr>
<td>EasyText</td>
<td>- Numerical data processing to generate specific analytical comments</td>
</tr>
<tr>
<td>FOG</td>
<td>- Bilingual (English and French) report generator</td>
</tr>
<tr>
<td>BT-45</td>
<td>- The BabyTalk project aims at providing automatic generation of textual summaries.</td>
</tr>
</tbody>
</table>
and B. Let be a finite set of data, A and B fuzzy or 2. prepared data analysis, 3. sentences generation. Systems [24], and BT-45, part of the Baby-Talk project [25].

As detailed below, NLG summary generation consists in three steps: 1. important data selection/preparation, 2. prepared data analysis, 3. sentences generation. Systems differ in the attention they pay to one step rather than another, mostly depending on the domain, as detailed hereunder.

Yseop generates financial consulting documents based on legal financial reports like a balance sheet and a profit and loss account. However, and since it is a commercial system, no details are given, except that the input data is processed using an expert system (steps 1 and 2) and the resulting text is generated using standard NLG techniques (step 3). It also seems to work in an integrated way, generating summaries directly into the clients’ software according to their online demos.

For their part, SumTime Mousam [21], SumTime Turbine [22] and BT-45 [25] apply to time-series data input. They pre-process the input using segmentation or rapid change detection (step 1, as detailed in [26]) so as to obtain an event list. Based on this event list, these systems create high-level abstraction. This step (step 2) is very dependent on the domain, as the three examples detail. SumTime Mousam aims at generating weather forecast reports, so it focuses on large segments of data, adding data for time and wind. SumTime Turbine, on his side, generates turbine log summaries. In this area, the interesting data for the engineers are the co-occurring events in the turbine based on different measure channels, so the system will focus on them. Lastly, in BT-45, it makes more sense to link the events using one of the three relations “cases”, “includes” and “associates”. In each case, this event analysis and high-level abstraction production is based on static hard-coded procedures and domain-dependant rules, processed through different kind of expert systems.

Then comes the text creation step (step 3). The three systems use a micro-planner, dedicated to mark phrases to be elided since they are redundant from one line of data to another, and a realization step which applies a reduction based on the micro-planner analysis. For instance, in SumTime Mousam, the two phrases “Wind backing SW by mid afternoon” and “Wind backing S by midnight” extracted from the data are replaced with “Wind backing SW by mid afternoon and S by midnight”.

EasyText [23] partly uses the same steps for its summary generation, but focuses more on the linguistic creation part (steps 2 and 3). The data extraction part (step 1) relies on a simple expert system, created from the rules explained by the analysts during the design phase. The linguistic step is extensively explained in the paper. The first action is “document structuring” which aims at generating a conceptual tree (step2, also used in SumTime and BT-45 systems). The tree here describes rhetorical relations between the semantic content of the selected values. The “tactical component” which consists of micro/macro planner and a surface realizer (like the realization step in SumTime and BT-45 systems, but more detailed), is based on a G-TAG [27] system. It is dedicated to the “segmentation of the text into sentences and linear ordering of these sentences, the choice of discourse connectives and other lexical items, the syntactic constructions within sentences, aggregation operations, referring expressions, semantic and syntactic parallelism, etc.” The surface realizer links sentences with each other.

III. PROTOFORMS AND FUZZY LINGUISTIC QUANTIFIERS

In this section, we focus on linguistic summaries using protoforms and more precisely on quantifiers (the Q in “QX are A” or “QBX are A” presented earlier). In this section, we show that more complex quantifiers than the classic “many” or “few” seem easy to use in natural language and can be represented in a fuzzy setting and that there exists a variety of treatments of fuzzy summaries still to explore.

A. Principles

The quantifiers enjoy a privileged position in the field of fuzzy linguistic summarizers, since they are responsible for aggregating the data, and thus summarizing it. They have been formalized by Zadeh in a fuzzy setting [3] on the basis of the cardinality of fuzzy sets evaluated by means of the σ-count (sum of the membership values over a fuzzy set). It is used for absolute quantifiers such as “more than 10”, or the fuzzy proportion based on the conditional σ-count of a fuzzy set for propositional quantifiers such as “around a third”. This seminal work has been continued by A. L. Ralescu [28], Yager [29] who studied quantifiers in the light of fuzzy summaries [4], and D. Ralescu who defined a fuzzy cardinality providing a different view of quantifiers [30]. Liu and Kerre have proposed an overview of fuzzy quantifiers in [31], [32].

The interest of using a fuzzy set based representation of linguistic quantifiers consists in two aspects. The first one is the approximate meaning of quantifiers, not associated with crisp boundaries or amounts, but roughly understood in natural language.

The second one is the need to take account of the variability of these quantifiers according to the context. For instance, the meaning of the relative quantifier “few” is different if you speak of children in a class or in the world: “few children have understood the question” may correspond to 1 over one million. A fuzzy representation of quantifiers answers both concerns.

More formal works on quantifiers have been presented, regarding a fuzzy version of generalized quantifiers [33], a logical approach of fuzzy quantifiers [34] or a quantifier generation approach [35].

B. Measure of Validity

Let \( U = \{ u_1, ..., u_n \} \) be a finite set of data, A and B fuzzy or non fuzzy subsets of \( U \) with respective membership functions \( f_A \) and \( f_B \). Let \( Q \), with membership function \( f_Q \), be a fuzzy quantifier in the sense of [3], i.e. which “denote the collection of quantifiers in natural languages whose representative elements are: several, most, much, not many, close to five, approximately ten, frequently, etc.”. It can be
the sequence of dates associated with data. For the sake of simplicity, let us only consider the influence of the specificity of $A$ in considering elements of $U$ which "really" belong to $A$, instead of all of them: we use an $\alpha$-cut of $A$, for a threshold $\alpha$ preferably at least equal to 0.5, i.e. consider:

\[
t(S) = f_Q \left( \sum_{i=1}^{r} f_A(u_i) \right)
\]

(8)

2) Instead of the previous positive real-valued cardinality, we can use an integer-valued one [30] which seems more natural to evaluate the number of elements satisfying $A$. We then obtain:

\[
t(S) = f_Q \left( nCard(A) \right)
\]

(9)

with:

\[
nCard(A) = \begin{cases} 
0 & \text{if } A = \emptyset \\
j & \text{if } A \neq \emptyset \text{ and } \mu_j \geq 0.5 \\
j-1 & \text{if } A \neq \emptyset \text{ and } \mu_j < 0.5
\end{cases}
\]

(10)

and $j = \max \{ i \mid 1 \leq i \leq n, \mu_{i+1} + \mu_{i} > 1 \}$

3) Having in mind to compare the fuzzy cardinality of $A$ and the given quantifier $Q$, we can also think of a comparison between two fuzzy sets and choose an appropriate measure of comparison $C$, for instance a resemblance or a similarity in the framework of [37]. We propose to use the following:

\[
t(S) = C(Q, fCard(A))
\]

(11)

for instance:

\[
t(S) = M(Q \cap fCard(A))
\]

(12)

for a fuzzy set measure $M$. Such a quantity presents the advantage of providing a more accurate evaluation of the compatibility between the proposed quantifier $Q$ and the actual fuzzy cardinality $fCard(A)$. This advantage is of great importance in the case where we consider quantifiers more complex than the classic ones such as "more" or "few".

D. The Case of Temporal Quantifiers

The quantifiers used in linguistic summaries are mainly referring to a number of elements or occurrences, in the case where the absolute cardinality is used, or to ratios in the case where proportional cardinalities are taken into account. Various other quantifiers can be thought of, for instance those dealing with time such as "often" or "seldom". In this section we consider the case where $U = \{u_1, \ldots, u_n\}$ is a temporal sequence of data.

We can consider that there is a one-to-one mapping between the number of elements of $U$ satisfying property $A$ and the frequency of property $A$ in $U$. Then "most" can be roughly associated with "often" and "few" with "seldom" and we can consider at first glance that there is no difference between a general management of quantifiers as explained in the previous sections or the specific management of temporal quantifiers.

Nevertheless, it must be noted that the specificity of temporal quantifiers is essentially due to the fact that the $u_i$'s are ordered and occur at given moments. Let us call $V = \{v_1, \ldots, v_n\}$ the sequence of dates associated with data of $U = \{u_1, \ldots, u_n\}$. For the sake of simplicity, let us only consider regular moments, independent of the universal time, like Day 1, Day 2, ...
The remaining problem in such a framework is the complexity of some interesting quantifiers. For example, a linguistic quantifier such as “from time to time” cannot be managed by means of the only cardinality of fuzzy sets. A summary for a given period like “from time to time, the sales are high” takes into account that few of the sales are high (or seldom, the sales are high) and, in addition, these high sales are not grouped in one interval of V, but they are somewhat dispersed over V.

We are facing what we can call complex quantifiers. To some extent, we can consider that we are in a kind of branching quantification [38] of the form:

\[ S : Q_1 u_i's \& Q_2 u_i's \text{ are } A \]

with for this example, \( Q_1 = "\text{seldom}" \) and \( Q_2 = "\text{in a dispersed way}" \).

A quantifier such as \( Q_2 \) does not involve any notion of cardinality, but a notion of dispersion, which can be identified through an entropy, for instance.

IV. EXPERIMENTS WITH ONLINE BOOKSELLER DATA

A. Experimental Data

1) Collected Data

The quantifiers discussed above have been experimented on a database used to support the activity of an online bookseller.

The database contains information about the books and their rankings. Weekly, monthly and yearly rankings are computed based on the book sales. They list the 100 best selling books over the considered time lapse.

With this data, we want to highlight a link between the sales of some kind of books and the period in the year, considering diet books.

2) Data Pre-Processing

To figure so, we tag in the database all the books linked to the “diet” topic (actually those whose title contains “diet”, “get slimmer”, “lose weight”, etc.). Then, we calculate a score per month for each book tagged as “diet”. The book in \( r \)th position receives the score of \( 101-r \), for \( r = 1...100 \). A book not present in the ranking receives 0. This way, we take into account both the number of books in the ranking and their positions.

For instance, if two books about diet are listed in the April 2002 rankings, the first with the rank 10 (i.e. the 10th best sale in April 2002), and the second with the rank 73 (73rd best sale), then the score for April 2002 will be \( 101-10+101-73 = 119 \). Then, we calculate the highest score for the current year, and we normalize (divide by the maximum) each month’s score, in order to have values in \([0, 1]\).

Figures 1 and 2 present the resulting data. Figure 1 shows the normalized score from 2002 to 2011. Figure 2 illustrates the sum per month of these scores. They both show that there seems to exist a link between the season and the good ranking of the books about diets in the best seller lists. Now, we show how the different fuzzy quantifiers presented earlier enable us to evaluate summaries presenting this seasonality.

B. Using Fuzzy Summaries

1) Summary Definition

At first sight, figure 1 shows that the scores of the Diet books are not very good. We represent this fact with the sentence “Less than half of the scores for the diet books are good”. On the other hand, the good scores are attained during the first semester of the year, and we can even say that the best ones happen in Spring. We exemplify this with the sentence “Around half of the good scores for the diet books are in Spring”.

The first sentence matches the protoform in (1) “Q u_i’s are A”, the second matches (2) “Q A u_i’s are B”.

In the first case, \( Q \) is “Less than half”, \( u_i \)'s are the normalized scores of the diet books and \( A \) is “good scores”. In the second, \( Q \) is “Around half”, \( A \) is “good scores”, \( u_i \)'s are the normalized scores of the diet books and \( B \) is “Spring”.

Beside the computation of the two sentences truth values, we study the following two properties:

- **Property 1 – Non contradiction**: When a sentence has a high truth value, then the opposite sentence has a low one. For instance, if the sentence “Less than half of the scores are good” has a high truth value, then “More than half of the scores are good” must have a low value.
Property 2 – Double negation: Using the negation of 2 or any even number of parameters in the sentence must give approximately the same truth value. For instance, if “Less than half of the scores are good” have a high truth value, then “More than half of the scores are bad” should have a high truth value as well.

These properties clearly depend on the definitions of quantifiers \( Q \), as well as variables \( A \) and \( B \).

2) Linguistic Variables and Quantifiers

In order to evaluate these sentences, we define several linguistic variables. The first one is “Score”, with three modalities, “Good”, “Average”, or “Bad”. Figure 3 shows this partition. The score used is the normalized one, thus in \([0, 1]\).

The second linguistic variable is “Calendar”, which determines the membership degree of a given month to a given time in the year. For the purpose of this paper, we keep “Spring” and “Autumn”, as Figure 4 illustrates. This linguistic variable obviously depends on the cultural area where the summary is calculated. In countries near the equator, “Spring” and “Autumn” are not relevant. “Winter” as well happens from December to March in the northern hemisphere, whereas it is summer at that time in the southern one (we do not use this value in this article, but it could be included as well in the Calendar linguistic variable). This fact simply illustrates the fuzzy logic’s ability of capturing the cultural specificities.

Lastly, we also define three quantifiers: “Less than half”, “Around half” and “More than half”, as displayed in Figure 5. These quantifiers are relative ones, which means they do not take an absolute value as an input, but a relative one, in \([0, 1]\).

3) Computation of the Truth Values

Here, we compare the different methods introduced in the previous section. Data are processed using a 5-step method to compute the truth value. We study all the relevant combination configuring 3 values for the \( \alpha \)-cut (see (8)), 3 cardinalities (“Zadeh”, which is as standard \( \sigma \)-count as used in (3), “nCard” as in (10) and “fCard” as in (6)) and 2 methods (“Zadeh” as in (3) and (4) or “Similarity” as in (12)). These parameters, as well as the quantifiers and linguistic variables are used in the order given in Figure 6. Tables II and III present the results of the truth calculation for the different possible values.

C. Result Analysis

The objective is to check the truth of “Less than a half of the scores for the diet books are good”. Actually, this sentence is not true. Indeed, when we look closely at Figure 1, we can see that, for the few first years, the sentence is true, but it is not for the remaining ones. And this is confirmed by the results. With \( \alpha = 0 \), the sentence scores are between 0.11 and 0.32 (Table II, column 7, rows 1-3), meaning it is quite false. But it is not zero either, hence explaining our first impression. However, the sentence “Around half of the scores for the diet books are good” is around 0.7 (Table II, column 8, rows 1-2 – the Similarity result is ignored here, since it is much too low compare to the others), confirming the second look at the graph. The truth values for the other sentence “Around half of the good scores for the diet books are in Spring” confirms our first guess, since it equals 1 for every \( \alpha \)-cut (Table III, column 9). The sentence “Less than half of the good scores for the diet books are in Autumn” (Table III, column 14) is very true as well. More interesting are columns 12 and 13 in Table 3, since they show that bad scores happen in Autumn, “a bit more than 50%”.

Several other remarks can be made on the basis on these results. Further investigation must be carried out in order to examine their generality.
−\(\alpha\)-cut effect: we confirm an expected result with the \(\alpha\)-cut, that increasing values for \(\alpha\) leads to lower truth values with increasing quantifiers, and lower values with decreasing ones. Another point with \(\alpha\)-cut is the different evolution of the result using different methods and cardinalities: they all decrease or increase the same way depending on \(\alpha\) and the type of quantifiers, but at different pace.

−Property 1 is satisfied except for Similarity. For instance, columns 4, 5 and 6 in Table II and III, or columns 14, 15 and 16 in Table 3.

−Property 2 depends on the definition of the linguistic variables. For instance, columns 5 and 9 in Table III comply with this property, but here the negation of “\(< 50\%\)” is “\(\approx 50\%\)”.

−Cardinalities. Ralescu cardinality is not impacted the same way \(\sigma\)-count is. Indeed, the former is an integer one, thus leading to “threshold” effects, whereas the latter, being real, immediately shows the changes.

−Similarity: as it is used here, the similarity between the Ralescu’s fuzzy cardinality and the tested quantifier is not efficient, since the fuzzy cardinality is not normal, i.e. its highest value is not 1. Worst, it is usually quite low, around 0.5. Hence, even when it totally belongs to the quantifier, the result of its intersection is quite low, leading to a low truth value. Nevertheless, this method should be investigated further on, since it does seem to provide different result than Zadeh’s. For instance, “Less than 50\% are bad” (Table II, column 4, rows 1 and 3) returns 1.00 using Zadeh’s method, and 0.14 using Similarity. On the other hand, “Around 50\% are good” (Table II, column 8, rows 1 and 3) gives 0.76 with Zadeh and 0.18 with Similarity. It suggests that they do not provide the same interpretation, over the same data, leading to further research on them.

V. CONCLUSION

We have proposed an overview of linguistic summarization, presenting the main streams of a symbolic representation and management of numerical data, which can be crisp or fuzzy. We have pointed out that fuzzy approaches bring solutions to the imprecision of quantification and the use of subjective qualification of data. Nevertheless, the protoforms used in a fuzzy setting are still far from a natural language description of data.

We then presented some ways of processing data using fuzzy experimental quantifiers. The results are really promising, for all kinds of summaries, and deserve to be further examined. We plan to go deeper about several topics

\[\begin{array}{cccccccccccccccc}
\hline
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\
\hline
\alpha\text{-cut} & \text{Method} & \text{Cardinality} & \text{Method} & \text{Cardinality} & \text{Method} & \text{Cardinality} & \text{Method} & \text{Cardinality} & \text{Method} & \text{Cardinality} & \text{Method} & \text{Cardinality} & \text{Method} & \text{Cardinality} \\
\hline
0 & Z & \sigma & 1.0 & 0.00 & 0.00 & 0.00 & 1.00 & 0.00 & 0.00 & 0.62 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0 & Z & N & 1.0 & 0.00 & 0.00 & 0.00 & 1.00 & 0.00 & 0.00 & 0.62 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.5 & Z & \sigma & 1.0 & 0.00 & 0.00 & 0.00 & 1.00 & 0.00 & 0.00 & 0.62 & 0.41 & 0.59 & 1.00 & 0.00 & 0.00 \\
0.5 & Z & N & 1.0 & 0.00 & 0.00 & 0.00 & 1.00 & 0.00 & 0.00 & 0.62 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.8 & Z & \sigma & 1.0 & 0.00 & 0.00 & 0.00 & 1.00 & 0.00 & 0.00 & 0.62 & 0.41 & 0.59 & 1.00 & 0.00 & 0.00 \\
0.8 & Z & N & 1.0 & 0.00 & 0.00 & 0.00 & 1.00 & 0.00 & 0.00 & 0.62 & 0.41 & 0.59 & 1.00 & 0.00 & 0.00 \\
\hline
\end{array}\]

The “Zadeh” method here is the one described in (4)
like testing other cardinalities (FE-Count, Sugeno, Choquet), using other quantifiers (like Glöckner’s [35]) especially the time-related ones, which could be useful with the time series, studying linguistic variables (rules to determine their relevance, automatic generation), understanding the links between alpha-cuts and cardinalities, and between Similarity and Zadeh methods.

ACKNOWLEDGMENT

The authors express their thanks to Marie-Jeanne Lesot for her valuable comments on the present version of the paper.

REFERENCES