This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2014.2328011, IEEE Transactions on Fuzzy Systems

Linguistic Descriptions for Automatic Generation of Textual Short-Term Weather Forecasts on Real Prediction Data

A. Ramos-Soto, A. Bugarín, S. Barro, and J. Taboada

***Abstract*—We present in this paper an application which automatically generates textual short-term weather forecasts for every municipality in Galicia (NW Spain), using the real data provided by the Galician Meteorology Agency (MeteoGalicia). This solution combines in an innovative way computing with perceptions techniques and strategies for linguistic description of data together with a natural language generation (NLG) system. The application, named GALiWeather, extracts relevant information from weather forecast input data and encodes it into intermediate descriptions using linguistic variables and temporal references. These descriptions are later translated into natural language texts by the natural language generation system. The obtained forecast results have been thoroughly validated by an expert meteorologist from MeteoGalicia using a quality assessment methodology which covers two key dimensions of a text: the accuracy of its content and the correctness of its form. Following this validation GALiWeather will be released as a real service offering custom forecasts for a wide public.**

***Index Terms*—linguistic descriptions of data, natural language generation, computing with perceptions, open data**

I. Introduction

In recent years, governments and agencies from many countries have increasingly focused efforts on improving the accessibility of their citizens to public data, i.e., all the data that public bodies in a given country produce, collect or pay for, which is widely known as the Open Data paradigm [1]. These resources, which come from many different fields of knowledge, offer a high potential for re-use in new products and services. This scenario has been described very graphically with the following statement "data is the new oil for the digital age" [2].

However, there is still a significant gap between the resources offered by public institutions and the necessities of their potential consumers. One reason is that the publishing bodies are usually focused on the availability of their datasets rather than on providing tools or means for accessing and processing them. This often results in extensive catalogues of heterogeneous data which have almost no direct value for the potential consumers of that data.

Besides a lack of standardization, there is also a lack of tools and services which allow a better access and comprehension of the raw data provided by the public institutions. An interesting and illustrative example of this kind of services can be found in meteorology, where meteorological agencies offer both raw data and also several types of information pieces (such as forecasts, reports or meteorological warnings) that are elaborated by meteorologists from these raw data.

Artificial Intelligence provides us with tools which allow us to process and understand this massive availability of huge quantities of data. Originally, this objective has been assumed by the knowledge discovery in databases (KDD) field, but more specifically by its core stage, the data mining field [3], which assembles several tasks such as classification, association, clustering, trend analysis or summarization [4]. Summarization is of particular interest, since it abstracts data into useful information at different levels and dimensions. The abstracted information can adopt many forms, although the most common services come in the form of web-based visualization tools. However, other approaches taken by research fields such as natural language generation (NLG) or soft computing offer solutions to convert and summarize data into textual information which can be easily consumed by human users.

The creation of automatic textual summaries of data is a task which originally started within the NLG field. Several NLG approaches which generated summaries of data include ANA [5], which generated summaries of stock market activity; LFS [6], which generated summaries of statistical data; SUMGEN [7], which generated summaries of events in a battle simulation; TEMSIS [8], which generated summaries of environmental data; TREND [9], which generated summaries of historical weather data; and, more recently, BabyTalk [10], which generates medical reports for neonatal intensive care data. However, the most successful NLG systems for data summarization, at least in terms of public impact and usefulness, generate automatic textual weather forecasts from numerical prediction data. A few systems, such as FoG [11], MultiMeteo [12] and SUMTIME-MOUSAM [13], [14], have been used by meteorological agencies to automatically produce public weather forecasts.

At the same time, within the fuzzy logic and soft computing field, the paradigm of computing with words (CWW) [15], and its later evolution computing with perceptions (CWP)

A. Ramos-Soto, A. Bugarín and S. Barro are with the Research Centre on Information Technologies (CiTIUS), University of Santiago de Compostela, Spain (e-mail: alejandro.ramos@usc.es, alberto.bugarin.diz@usc.es, senen.barro@usc.es). J. Taboada is with MeteoGalicia, Santiago de Compostela, Spain (e-mail: juan.taboada@meteogalicia.es).

This work was supported by the Spanish Ministry for Economy and Competitiveness under grant TIN2011-29827-C02-02. It was also supported in part by the European Regional Development Fund (ERDF/FEDER) under the project CN2012/151 of the Galician Ministry of Education.

A. Ramos-Soto is supported by the Spanish Ministry for Economy and Competitiveness (FPI Fellowship Program).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2014.2328011, IEEE Transactions on Fuzzy Systems

[16], [17], made their appearance in the 1990s. As opposed to other classical approaches, these paradigms involve a fusion of natural languages and computation with linguistic variables [16]. Although many new approaches based on CWW have emerged, one of the most promising tools is linguistic data summarization [18], [19], which employs fuzzy quantified propositions to obtain linguistic summaries involving one variable (as in "Most of the dogs are brown" or "A few trees are tall") or more than one variable (as in "Some of the brown dogs are heavy" or "Most of the tall trees are very old"). Since then, linguistic summarization from CWW has been applied in several practical cases and, with the appearance of CWP, some authors have started to refer to linguistic summaries as linguistic descriptions of data (LDD) [20], which understand linguistic summaries as a tool to describe human perceptions. For reasons of clarity, we will use in this paper the term linguistic descriptions of data. Examples of fields of application of linguistic description approaches include descriptions of the patient inflow in health centers [21], domestic electric consumption reports [22], human activity based on mobile phone accelerometers [23] or human gait quality [24]. Other approaches use more complex expressions involving relationships among different attributes (in economic data [25], in sales data [26] or the analysis of investment fund quotations [27]).

Most of these approaches are very strongly dependent on the field of application and the users' needs of information. A more general approach which is able to construct different kinds of linguistic descriptions regardless of the application domain is still an open challenge in this field. Nevertheless, steps in this direction have been taken by providing general criteria on how to structure quantified sentences in order to obtain more complex descriptions [28] or on how to build and evaluate linguistic descriptions [29], [30], [31].

Another open challenge is the relationship between linguistic descriptions in CWP and NLG. Until now, both have followed separate paths, although it remains clear that both can contribute to each other in a substantial way [26].

With both linguistic descriptions from CWP and textual summaries from NLG as inspiration, we present in this paper GALiWeather [32], an application which automatically generates short-term weather forecasts in the form of natural language texts for the Galician Meteorological Agency (MeteoGalicia) [33]. This solution employs in an innovative way a LDD computational method combined with a NLG system in order to solve a real life information need, as opposed to other approaches which only present test use cases and do not address the whole problem of adapting their solutions to real final user needs and demands. For this, the use of fuzzy procedures through linguistic variables and quantifiers allows the application to model imprecise concepts included in the linguistic descriptions. Furthermore, the quality of these descriptions, which are generated as natural language texts by the NLG system, has been assessed by an expert meteorologist in two key dimensions, verifying that the textual forecasts are both correct and properly expressed.

The next section introduces the context in which this solution has been devised. In Section III a formal description of the forecast input data and the linguistic description computational method is provided, followed by an extensive overview of the NLG system. Section IV addresses the validation process and results obtained for our application. Section V contains some insights about a methodological conceptualization of our approach and finally in Section VI we present the most relevant conclusions.

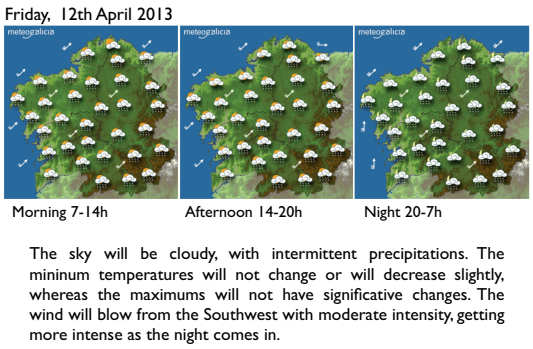


Fig. 1. Example of a real weather forecast for 12th April, 2013 for Galicia, published at [33].

II. Short-term web forecasts for Galicia

The operative weather forecasting offered by the Galician (NW Spain) Meteorology Agency through its website (MeteoGalicia [33]) consisted until now of a global description of the short-term meteorological trend (Fig. 1). This service has been recently improved in order to provide visitors with symbolic forecasts for each of the 315 municipalities in Galicia, thus improving its quality and allowing users to obtain more precise information about specific locations of the Galician geography.

Figure 2 shows the current web application for consulting municipality forecasts [33], which has been graphically divided in blocks for an easier explanation. Block 1 contains a shortcut list to the seven most important municipalities in Galicia, which allows a direct access to their forecast data (the user can select a favorite municipality, which is loaded by default in posterior visits). Block 2 allows the user to search for the rest of the municipalities, which are grouped according to the Galician province they belong to. It also allows to add to the shortcut list in Block 1 the selected municipality. The short-term forecast is shown in Block 3, which offers symbolic data for wind and sky state and numeric data for temperatures for four days, including morning, afternoon and night each day. Block 4 shows the mid-term forecast for several days and includes a global comment about the weather in Galicia in general, which consequently remains the same for every municipality.

This increase in the quantity of available numerical-symbolic data has a main downside, which resides in the lack of natural language forecasts which describe this set of data. This issue makes forecasts harder to understand, since users need to look at every symbol and detect which phenomena are relevant and when they will occur, whereas natural language descriptions directly provide all this information. In the case of a mid-term forecast, its uncertainty allows the inclusion of a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2014.2328011, IEEE Transactions on Fuzzy Systems

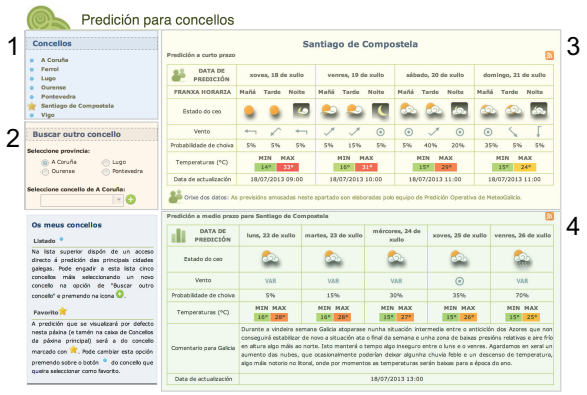


Fig. 2. Short-term and mid-term municipality forecast web application for Galicia [33].

global description, which is written by a meteorologist. However, for short-term forecasts, which are much more accurate, the meteorological diversity causes that several meteorological phenomena may occur at the same time in different areas. Thus, to issue daily textual forecasts upon 315 municipalities is not feasible.

In order to address this issue we have developed an application which, from short-term data, generates linguistic descriptions which highlight meteorological phenomena considered important by an expert meteorologist. The style and contents of the natural language linguistic descriptions for each location are similar to the general one presented in Fig. 1.

III. Application Description

The solution we have devised employs numerical-symbolic forecast data and additional expert information to generate the final output textual weather forecasts in two separate tasks. The first task converts the numerical-symbolic input data into linguistic descriptions (encoded in an intermediate language). These descriptions are created through a computational method which abstracts data values into linguistic labels dealing with uncertainty and temporal references. In the second stage, a NLG system translates the intermediate codes into a natural language forecast for one of the available final output natural languages, which is ready for human consumption. A general schema of this process is shown in Fig. 3.

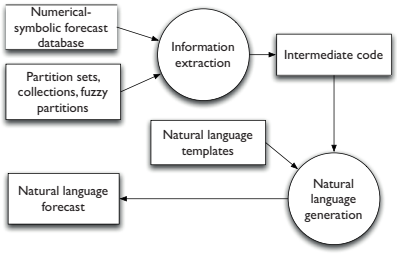


Fig. 3. General schema of the application architecture.

A. Input weather forecast data characterization

MeteoGalicia's database offers a dataset which covers all the 315 Galician municipalities and includes forecast data associated to several items in a four-day temporal window. This data is heterogeneous in its nature and includes values in degrees Celsius and weather symbols represented by codes. For instance, the meteorologists have characterized the sky state phenomena as 21 numerical codes (values in the interval [101,121]) and the wind phenomena as 34 numerical codes (values associated to a given intensity and direction in the interval [299,332]). These numerical codes are used to display graphical symbols in the forecast website. Figure 4 shows an example of a real short-term forecast data series.

Formally, each municipality M has an associated forecast data series set $FD_M = \{SS_M, W_M, TMAX_M, TMIN_M\}$, which includes data series for the input variables considered: sky state (SS_M), wind (W_M) and maximum ($TMAX_M$) and minimum ($TMIN_M$) temperatures. For clarity reasons, without loss of generality, we will consider a single municipality data series in the explanations that follow ($FD_M = FD$). Each data series element in FD is characterized in what follows:

- **Sky state** (SS). It provides three numerical codes per day (morning, afternoon, night) about two meteorological variables of interest, namely cloud coverage and precipitation. From a formal point of view, $SS = \{ss_1, \ldots, ss_i, \ldots, ss_{12}\}$, where $ss_i \in [101, 121] \forall ss_i \in SS$. Each code in the interval $[101, 121]$ has a specific sky state meaning (for example, 111 means "covered with rain").
- **Wind** (W). It provides three numerical codes per day about the wind intensity and direction. $W = \{w_1, ..., w_i, ..., w_{12}\}$, where $w_i \in [299, 332] \forall w_i \in W$. Each code in the interval $[299, 332]$ has an associated wind direction and intensity (for instance, 317 means "strong wind from the North").
- **Temperature** ($TMAX$ and $TMIN$). Maximum and minimum forecasted temperatures are given in degrees Celsius with a resolution of 1 degree and one value per day:
  - $TMAX = \{tmax_1, tmax_2, tmax_3, tmax_4\}$, where $tmax_i \in [-60°C, 60°C] \forall tmax_i \in TMAX$.
  - $TMIN = \{tmin_1, tmin_2, tmin_3, tmin_4\}$, where $tmin_i \in [-60°C, 60°C] \forall tmin_i \in TMIN$.

For each forecast data series FD, our application obtains linguistic descriptions about seven forecast variables, namely cloud coverage, precipitation, wind, maximum and

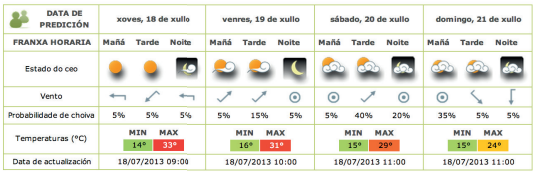


Fig. 4. Real example of a data source for a given location used in the generation of the automatic weather forecasts.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2014.2328011, IEEE Transactions on Fuzzy Systems

minimum temperature variation and maximum and minimum temperature climatic behavior [1]. For this, we have devised a computational method divided in several linguistic description generation operators.

B. First stage: Linguistic description generation method

The first stage of our application obtains a linguistic description for every variable, which consists in sets of linguistic labels and temporal references which contain the relevant information extracted from the raw data. This process, as it can be seen in Fig. 5, consists of providing each linguistic description operator with its corresponding data and expert knowledge (in the form of crisp and fuzzy partition sets and numeric categories) in order to generate the intermediate linguistic descriptions. Each operator is formally described in what follows.

1) Cloud coverage fuzzy operators: Two different fuzzy operators are used in the linguistic description generation of the cloud coverage variable. The first one provides a chronological description, while the second one provides a short-term global description when the previous description is not appropriate.

1) Chronological description fuzzy operator.
   - **Input**:
     - Sky state data series $SS = \{ss_1, \ldots, ss_i, \ldots, ss_{12}\}$.
     - A temporal fuzzy linguistic partition $CCT = \{cct_1, \ldots, cct_j, \ldots, cct_n\}$, where each temporal linguistic term cct_j has an associated fuzzy membership function $\mu_{cct_j} : \mathbb{N} \rightarrow [0, 1]$. For our application, $CCT = \{BEGINNING, HALF, END\}$ (Fig. 6).
     - A cloud coverage linguistic variable, defined as a set of precipitation categories $CCL = \{ccl_1, \ldots, ccl_k, \ldots, ccl_m\}$. Each linguistic term $ccl_k \in CCL$ has an associated crisp membership function $\mu_{ccl_k} : \mathbb{N} \rightarrow \{0, 1\}$, defined as:

$$\mu_{ccl_k}(ss_i) = \begin{cases} 1 & \text{if } ss_i \in ccl_k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

[1]It measures the difference between the forecasted temperatures and the temperature climatic mean, defined as the average for the previous 30 years in a given month.

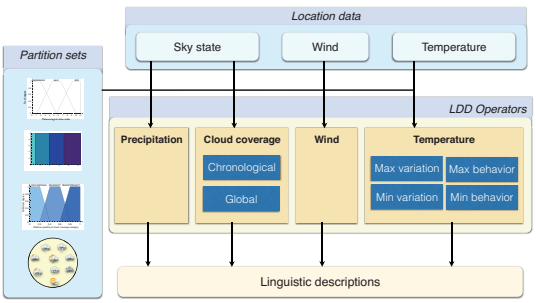


Fig. 5. Global schema of the linguistic description generation method.

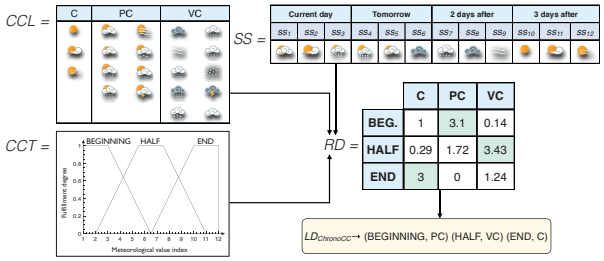


Fig. 6. Chronological description fuzzy operator definitions and process example.

In our application, $CCL = \{C, PC, VC\}$ ("clear", "partly cloudy", "very cloudy"), as shown in Fig. 6.

   - **Procedure**. This operator provides the most appropriate cloud coverage linguistic term ccl_k for each temporal subdivision cct_j. A relevance degree is calculated for each pair of cloud coverage and temporal labels and the label pairs with the highest degree are then selected (one per temporal label):
     - Relevance degree matrix RD, where each value $RD_{j,k}$ determines the importance a cloud coverage linguistic term ccl_k has within a temporal sub period cct_j: $RD_{j,k} = \sum_{i=1}^{|SS|} \mu_{ccl_k}(ss_i) * \mu_{cct_j}(i)$
     - Set of the most appropriate cloud coverage label for each temporal label, ordered by the temporal partition index j: $CCTL = \{(cct_j, ccl_k) | RD_{j,k} = max(RD_j)\}$
   - **Output**. A chronological cloud coverage linguistic description as an intermediate code characterized by the following concatenation:

$$LD_{ChronoCC} \rightarrow (cct_1, ccl_k) \ldots (cct_n, ccl_k)$$

Figure 6 shows the definitions of both linguistic variables for our application and an example of the chronological cloud coverage linguistic description process. This description is provided only if the following experimental condition is fulfilled: $\forall (cct_j, ccl_k) \in CCTL, RD_{j,k} \geq 3$. This condition ensures that every cct_j has an associated predominant cloud coverage type ccl_k, while maintaining tolerance to the appearance of other cloud coverage categories in SS. Otherwise, the linguistic description generated by the second operator is provided.

2) Global quantification description fuzzy operator. This operator provides a global description of the cloud coverage state for the whole short-term period.
   - **Input**:
     - Sky state data series $SS = \{ss_1, \ldots, ss_i, \ldots, ss_{12}\}$.
     - A cloud coverage predominance linguistic label $CCQ = \{ccq_1, \ldots, ccq_j, \ldots, ccq_n\}$, where each linguistic term ccq_j has an associated fuzzy quantifier $\mu_{ccq_j} : [0, 1] \rightarrow [0, 1]$. In our case, $CCQ = \{OCCASIONAL, RELEVANT, PREDOMINANT\}$ (Fig. 7).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2014.2328011, IEEE Transactions on Fuzzy Systems

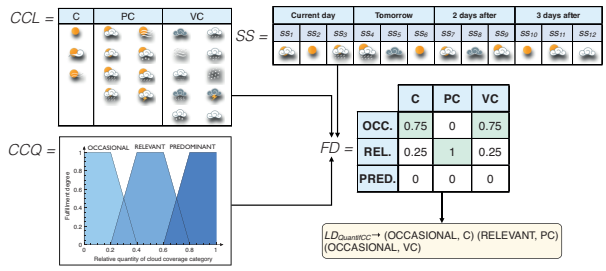


Fig. 7. Global quantification description fuzzy operator definitions and process example.

  - A cloud coverage linguistic variable CCL, as defined in the previous operator.
- **Procedure**. This operator quantifies the occurrence of the different cloud coverage categories ccl_k using Zadeh's quantification model [34]:
  - Fuzzy fulfillment degree matrix FD, where $FD_{j,k} = \mu_{ccq_j}(\sum_{i=1}^{|SS|} \frac{\mu_{ccl_k}(ss_i)}{|SS|})$
  - Set of cloud coverage label and quantifier label pairs with the highest fulfillment degree: $CCQL = \{(ccq_j, ccl_k) | FD_{j,k} = \max_l FD_{l,k}\}$, where j is minimum.
- **Output**. A cloud coverage linguistic description as an intermediate code characterized by the following concatenation:

$$LD_{QuantifCC} \rightarrow (ccq_j, ccl_1) \dots (ccq_j, ccl_m)$$

Figure 7 shows the definition of the fuzzy quantifiers μ_{ccq_j} and an example of this linguistic description process.

2) Precipitation episode extractor operator: This operator extracts precipitation episodes from the sky state values. These periods are classified according to the kind of precipitations detected:

- **Input**:
  - Sky state data series $SS = \{ss_1, \dots, ss_i, \dots, ss_{12}\}$.
  - A precipitation linguistic variable, defined as a set of precipitation categories $PV = \{pv_1, \dots, pv_j, \dots, pv_n\}$, where each linguistic term pv_j has an associated crisp membership function $\mu_{pv_j} : \mathbb{N} \rightarrow \{0, 1\}$, where μ_{pv_j} is defined identically as μ_{ccl_k} in expression (1).
- **Procedure**. This operator extracts an ordered set of precipitation episodes $PE = \{pe_1, \dots, pe_k, \dots, pe_m\}$, where each episode is characterized as $pe_k = \{START, END, LABELS\}$. The algorithm in Fig. 8 describes how the precipitation operator extracts the relevant episodes from SS:
- **Output**. A precipitation linguistic description for each precipitation episode pe_k as an intermediate code characterized by the following concatenation of terms:

$$LD_{Precipitation_k} \rightarrow START_k\ END_k\ LABELS_k$$

procedure PRECIPITATIONEPISODEEXTRACTOR(SS,PV)
  $PE \leftarrow \{\}$
  $pe_k \leftarrow \emptyset$
  while $i < |SS|$ **do**
    $active\_period \leftarrow False$
    for all $pv_j \in PV$ **do**
      if $\mu_{pv_j}(ss_i) = 1$ **then**
        if $pe_k \neq \emptyset$ **then**
          $pe_k.LABELS \leftarrow pe_k.LABELS \cup pv_j$
        else
          $pe_k \leftarrow \{START, END, LABELS\}$
          $pe_k.START \leftarrow i$
          $pe_k.LABELS \leftarrow \{pv_j\}$
          $PE \leftarrow PE \cup pe_k$
        $active\_period \leftarrow True$
        break
    if $\neg active\_period\ \&\ pe_k \neq \emptyset$ **then**
      $pe_k.END \leftarrow i - 1$
      $pe_k \leftarrow \emptyset$
    $i \leftarrow i + 1$
  if $active\_period\ \&\ pe_k \neq \emptyset$ **then**
    $pe_k.END \leftarrow |SS|$
  return PE

Fig. 8. Precipitation episode extractor procedure.

In this case, $PL = \{I, P, SN, ST, H\}$ ("intermittent","persistent","snow","storm","hail") is defined for precipitation (although "intermittent" and "persistent" are not explicitly included in the final natural language forecasts, as required by the meteorologists). Figure 9 shows the definition of PL and provides a graphical example of the precipitation linguistic description generation process.

3) Wind operator: It follows a similar strategy to the precipitation operator, although in this case it does not convert the original values into labels.

- **Input**:
  - Wind data series $W = \{w_1, \dots, w_i, \dots, w_{12}\}$.
  - A numeric interval $AW = [aw_a, aw_b] | AW \subset [299, 332]$ (as indicated in Section III-A), which specifies the relevant wind values to be extracted by the operator. In our application, $AW = [317, 332]$. This interval corresponds to strong and very strong winds, which are the only relevant wind conditions to be included in the descriptions according to the meteorologists.
- **Procedure**. This operator extracts an ordered set of wind episodes $WE = \{we_1, \dots, we_k, \dots, we_m\}$, where each

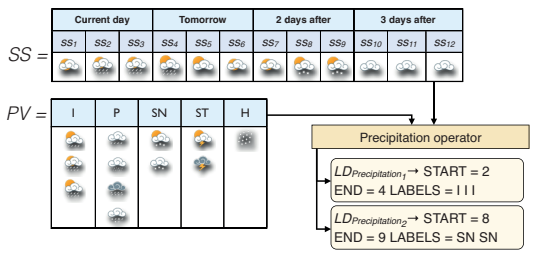


Fig. 9. Schema of the precipitation operator method with the current meteorological phenomena categories for precipitation and its associated labels.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2014.2328011, IEEE Transactions on Fuzzy Systems

episode is characterized as $we_k = \{START_k, END_k, SYMBOLS_k\}$. The algorithm in Fig. 10 describes how the wind operator extracts the relevant episodes from W.

- **Output**. A wind linguistic description for each wind episode we_j as an intermediate code characterized by the following concatenation: $LD_{Wind_k} \rightarrow START_k\ END_k\ SYMBOLS_k$. For example, if there is a period of strong wind within W, we could obtain a linguistic description such as "START=2 END=4 LABELS=322,322,322", meaning "from tonight ($i = 2$) until tomorrow afternoon ($i = 4$) there will be strong wind from the southwest ($w_i = 322$)".

4) Temperature operator: This operator generates a linguistic description which reflects the temperature trend for the 4-day period and also obtains information about the climatic behavior of the forecasted temperatures. Thus, four variables are considered: maximum and minimum temperature variations and maximum and minimum climatic behavior.

- **Input**:
  - Maximum temperature data series $TMAX = \{tmax_1, tmax_2, tmax_3, tmax_4\}$.
  - Minimum temperature data series $TMIN = \{tmin_1, tmin_2, tmin_3, tmin_4\}$.
  - A temperature variation linguistic variable, defined as $TV = \{tv_1, \ldots, tv_j, \ldots, tv_n\}$, where each linguistic term $tv_j \in TV$ has an associated crisp membership function $\mu_{tv_j} : \mathbb{R} \rightarrow \{0, 1\}$. In our application, $TV = \{ED, ND, MD, SD, WC, SI, MI, NI, EI\}$ ("extreme decrease", "notable decrease", "moderate decrease", "slight decrease", "without changes", "slight increase", ..., "extreme increase").
  - A temperature climatic behavior linguistic variable, defined as $TC = \{tc_1, \ldots, tc_j, \ldots, tc_n\}$, where each linguistic term $tc_j \in TC$ has an associated crisp membership function $\mu_{tc_j} : \mathbb{R} \rightarrow \{0, 1\}$. In our case, $TC = \{VL, L, N, H, VH\}$ ("very low", "low", "normal", "high", "very high").
- **Procedure**. This operator provides the linguistic terms with the highest membership degree from TV and TC for the four temperature variables considered:
  - Temperature variation: for maxima $TMAXV = tv_j|\mu_{tv_j}(tmax_{|TMAX|} - tmax_1) = 1$, and minima $TMINV = tv_j|\mu_{tv_j}(tmin_{|TMIN|} - tmin_1) = 1$.
  - Temperature climatic behavior: for maxima $TMAXC = tc_j|\mu_{tc_j}(\sum_{i=1}^{|TMAX|} \frac{tmax_i}{|TMAX|}) = 1$, and minima $TMINC = tc_j|\mu_{tc_j}(\sum_{i=1}^{|TMIN|} \frac{tmin_i}{|TMIN|}) = 1$.
- **Output**. A temperature linguistic description as an intermediate code characterized by the following term concatenation:

$$LD_{Temperature} \rightarrow TMINC\ TMAXC\ TMINV\ TMAXV$$

The definition of TV and a graphical example of the temperature operator are shown in Figure 11. As for TC, its associated crisp membership functions μ_{tc_j} are not shown in this example, since they vary for each municipality.

procedure WINDEPISODEEXTRACTOR(W, AW)
  $WE \leftarrow \{\}$
  $we_k \leftarrow \emptyset$
  while $i < |W|$ **do**
    $active\_period \leftarrow False$
    if $w_i \in AW$ **then**
      if $we_k \neq \emptyset$ **then**
        $we_k.SYMBOLS \leftarrow we_k.SYMBOLS \cup w_i$
      else
        $we_k \leftarrow \{START, END, LABELS\}$
        $we_k.START \leftarrow i$
        $we_k.SYMBOLS \leftarrow \{w_i\}$
        $WE \leftarrow WE \cup we_k$
      $active\_period \leftarrow True$
    if $\neg active\_period\ \&\ we_k \neq \emptyset$ **then**
      $we_k.END \leftarrow i - 1$
      $we_k \leftarrow \emptyset$
    $i \leftarrow i + 1$
  if $active\_period\ \&\ we_k \neq \emptyset$ **then**
    $we_k.END \leftarrow |W|$
  return WE

Fig. 10. Wind episode extractor algorithm.

C. Second stage: Natural language generation

The natural language generation (NLG) stage of this application consists of a domain-specific system which, following standard NLG techniques, has also been divided into different modules for each variable, so that changes in one of them do not affect the rest of the system. From a global perspective, each of these modules receives the intermediate linguistic description generated by their corresponding operator, parses it and generates the final textual forecast for its associated variable.

If we delve deeper into the natural language generation stage structure, the complexity of the final natural language descriptions is a factor which has determined the design and implementation approach we have followed. This includes evaluation criteria applicable to linguistic descriptions [29] such as the description length, but also NLG systems design methodologies as in [35] and [36].

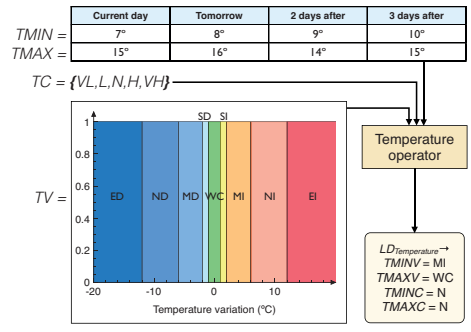


Fig. 11. Schema of the temperature operator, with the current definition of the temperature variation partition and its associated labels.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2014.2328011, IEEE Transactions on Fuzzy Systems

Thus, since the quantity of information in the descriptions is variable and the diversity of situations for each variable to be included ranges from simple to more complex, we have adopted two different NLG solutions. On one hand, we have defined templates in structured text files which contain generic natural language sentences for the simpler variables (cloud coverage, temperatures and wind). On the other hand, we have designed and implemented the generation of natural language sentences for precipitation inspired by standard NLG methodologies [35], [36].

1) Template-based NLG approach: This approach has been devised as a solution for variables whose corresponding natural language sentences have rather static structure and length, such as temperatures or cloud coverage. For example, a textual forecast for temperatures usually includes information about variation of maxima and minima and their climate behavior, and the only elements that differ from one forecast to another are the labels assigned to the variations and the behavior, whereas the syntactic structure and length of the forecasts remain the same.

In this context, structured text files, such as XML, allow to model and build templates of natural language sentences, where static text can be mixed with other elements, such as variables or optional texts within a sentence. We have taken advantage of this flexibility by designing templates for temperature, cloud coverage and wind variables. These templates are included in a document which also contains natural language label sets for variables, time expressions or other kind of language-dependent text resources. Figure 12 shows parts of a template document (in this case for English language), whose structure (Fig. 13) is comprised of the following elements:

- **Variable templates**, which include the generic natural language forecast structures for several variables, such as cloud coverage or temperature.
- **Label sets**, which contain the natural language vocabulary and expressions used to fill in the variable elements. They are the natural language equivalent to the crisp and fuzzy partition sets used in the linguistic description extraction stage. For example, in Fig. 11 the temperature variation labels in TV correspond to the label identifiers

```
<?xml version="1.0" encoding="UTF-8"?>
<language id="EN" name="English">
  <template id="T" name="Temperature">
    <case id="1">
      <part>
        <static>The temperatures</static>
          <space/>
          <static>will be</static>
          <space/>
        <option id="1">
          <variable id="norT"></variable>
          <space/>
          <static>for this period of the year,</static>
        </option>
        <option id="2">
          <variable id="minT"></variable>
          <space/>
          <static>for the minimums and</static>
          <space/>
          <variable id="maxT"></variable>
          <space/>
          <static>for the maximums compared
            to the expected for this time of the year,
          </static>
        </option>
        <space/>
      </part>
```

a) Temperature template sample

```
<labelset name="C" value="Cloud Coverage">
  <label name="C">clear</label>
  <label name="CL">cloudy</label>
  <label name="O">covered</label>
  <label name="F">foggy</label>
</labelset>
<labelset name="R" value="Rain">
  <label name="I">intermittent</label>
  <label name="P">persistent</label>
  <label name="SN">of snow</label>
  <label name="ST">stormy</label>
  <label name="H">of hail</label>
</labelset>
<labelset name="V" value="Temperature Variation">
  <label name="ED">in extreme decrease</label>
  <label name="ND">in notable decrease</label>
  <label name="MD">in moderate decrease</label>
  <label name="SD">in slight decrease</label>
  <label name="WC">without changes</label>
  <label name="SI">in slight increase</label>
  <label name="MI">in moderate increase</label>
  <label name="NI">in notable increase</label>
  <label name="EI">in extreme increase</label>
</labelset>
```

b) Cloud coverage, precipitation and temperature variation label sets

Fig. 12. Temperature template sample and label sets from the English language template document.

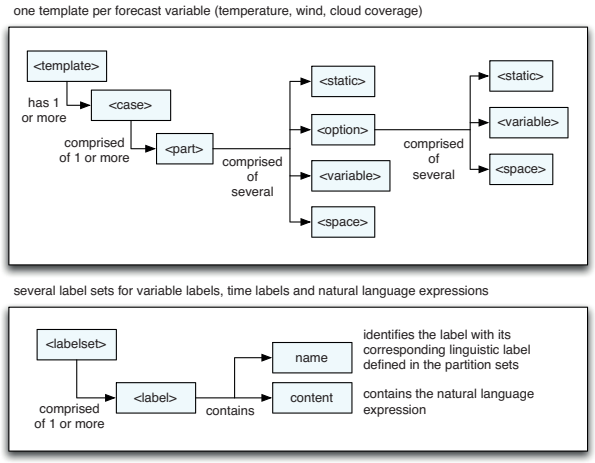


Fig. 13. Schema of the structure of a NLG template file, which contains generic sentences and label sets.

in the temperature variation label set in Fig. 12.

The template documents for the supported languages are loaded into structured objects within the application. Once the intermediate codes for the NLG template-based variables have been obtained, each NLG module (one per meteorological variable) parses its corresponding code and executes expert rules incorporated into the implementation code, so that according to certain detectable events in the intermediate language, different cases and options can be selected. Then, the template variables are filled with the natural language labels which correspond to the linguistic labels found in the intermediate code. Finally, the NLG template structures are translated into a natural language forecast text through the concatenation of the text values of each of their elements.

2) Precipitation NLG approach: The previous NLG approach is not suitable for variables such as precipitation, where several episodes can occur within a forecast term. This can lead to the generation of several natural language sentences which, although may reflect faithfully the meteorological data, are repetitive and tedious to read. Since the purpose of building linguistic descriptions in natural language is to provide users with textual information which should be easy to read and to understand, another NLG approach is required in order to achieve this goal.

Based on the concepts of a NLG system architecture described in [35] and [36], we have designed and developed a NLG module for precipitation which addresses redundancy or length excess in the obtained descriptions.

In [36], a NLG system is depicted as a six stage task, where one subtask is performed per stage. However, some of these subtasks may be merged or might not even be necessary, depending on the NLG requirements. Consequently, we have adapted some of these subtasks for the precipitation NLG module: content determination, sentence aggregation, lexicalization and linguistic realization. Others such as document planning were not considered, since in our case the NLG complexity is aimed at a sentence level. This process is summarized in Figure 14.

Content determination is defined in [36] as the process

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2014.2328011, IEEE Transactions on Fuzzy Systems

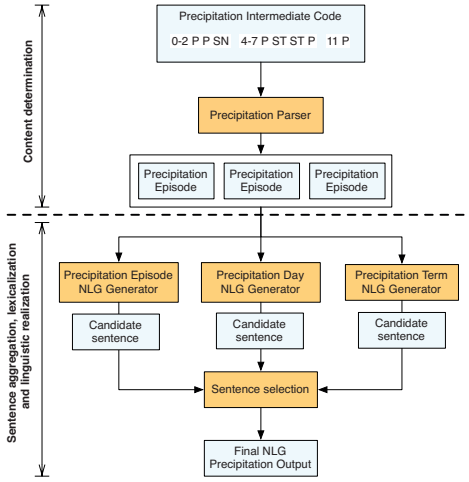


Fig. 14. Schema of the NLG approach for the precipitation variable.

which decides what information should be communicated in the text. This is done by creating a set of data objects (messages) which contain the filtered and summarized data. In our method, this task is partially performed in the linguistic description stage by the precipitation operator, which extracts the relevant data from the raw data and converts it into an intermediate language. The remaining task is to convert the intermediate code into data objects, which is done by the precipitation NLG module parser. As a result, a list of precipitation episodes, whose structure is shown in Fig. 15, is created and used by the subsequent natural language generation subtasks.

The precipitation data object structure in Fig. 15 shows that a precipitation episode has a duration (which can range from a single instant to the whole term). Furthermore, it might have associated nuances, which are subintervals within the episode in which the precipitation can be of different nature than rain (of snow, of hail or stormy).

The next NLG subtask we have adopted in our approach is sentence aggregation, which consists in grouping messages into sentences. We have contemplated three different ways of aggregating the precipitation episodes: by episodes, by days and whole-term aggregation. Consequently, we have created three different submodules which perform not only sentence aggregation, but also lexicalization and linguistic realization.

Lexicalization, which is the process of deciding which specific words and phrases should be chosen to express the concepts and relations in the messages, employs label sets defined in the NLG templates described in the previous approach.

Linguistic realization produces a text which is syntactically, morphologically and orthographically correct. Our precipitation approach obtains three candidate natural language precipitation sentences which describe the same input meteorological data set. The final output sentence for precipitation will be the shortest of the three, since we want to ensure that the obtained natural language forecasts remain as concise and brief as possible [29].

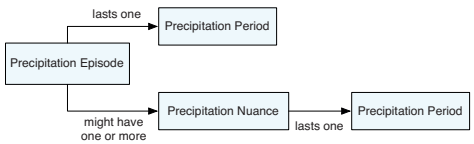


Fig. 15. Precipitation data object structure for the NLG stage.

D. Implementation details

This application has been developed in the cross-platform coding language Python, with the use of libraries for mathematical and fuzzy calculations (*numpy*, *pyfuzzy*) or text pattern recognition by grammars (*pyparsing*). The current implementation supports both Linux and Windows systems. The initially supported languages include Spanish and Galician. English was also included for research and scientific exposure purposes.

IV. Validation and Results

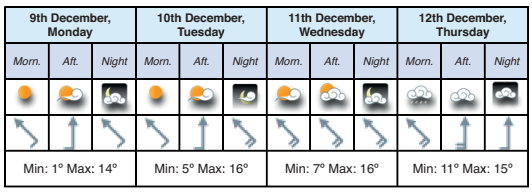
In this section we address the validation process for GALiWeather, which consists in an exhaustive expert-based revision and quality assessment of a set of automatically generated text forecasts obtained by the application. For this, we briefly discuss the state of the art in validation methodologies for both NLG and LDD fields and, based on these approaches, we explain in detail the validation methodology we have followed and its associated results. For illustration purposes, we present beforehand three examples of linguistic descriptions from the validation set obtained with the application.

A. Examples of automatic weather forecasts

Although the short-term prediction data series are limited to 32 values, the number of phenomena which must be considered and its temporal variability ensures a high richness in the obtained linguistic descriptions. As a proof of this richness, we present in this section the following examples covering several meteorological situations.

The example shown in Fig. 16 includes real forecast data for the town of Pontevedra, issued the 9th of December by MeteoGalicia. This case shows how GALiWeather performs in common meteorological situations, where the weather changes progressively.

The examples shown in Fig. 17 and Fig. 18 present unusual and odd meteorological conditions, which were generated



There will be clear skies at the beginning and towards the middle of the term, although at the end they will be very cloudy. We expect precipitations on Thursday morning. The temperatures will be normal for the minimums and high for the maximums for this period of the year, with minimums in notable increase and maximums without changes.

Fig. 16. Linguistic description forecast obtained with the application using real forecast data for Pontevedra, 9th of December, 2013.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2014.2328011, IEEE Transactions on Fuzzy Systems

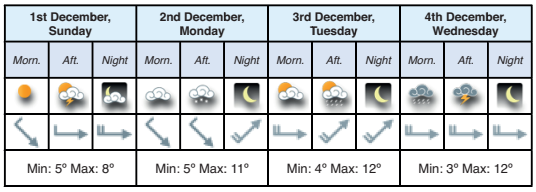


Fig. 17. Linguistic description forecast obtained with the application using synthetic data.

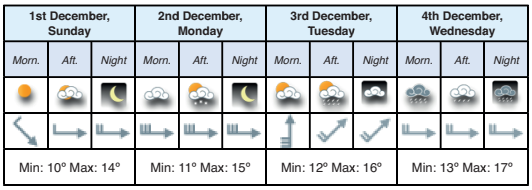


Fig. 18. Linguistic description forecast obtained with the application using synthetic data.

using synthetic data forecasts. These cases were created to test the application robustness under uncommon situations. Both examples include several meteorological phenomena, such as snow, storm, strong winds and temperature variations. Furthermore, each example shows a different precipitation sentence which aggregates the precipitation periods in a different way, as described in Section III-C.

B. Validation methodology

Validating automatic natural language generated texts is still an open challenge, even within the NLG field [37]. Several validation approaches do exist, both human and automatic, although in general, the human validation by experts is considered the most reliable [38], [39]. Consequently, the vast majority of NLG systems are validated using expert assessment, which usually implies answering questions about different aspects of the output texts. In the case of the LDD field several criteria have been proposed for evaluating and measuring the quality of the linguistic descriptions objectively [29], but they are not applicable in every approach and the information they provide is very limited compared to that of an expert, besides the fact that many LDD approaches do not reach the NLG stage and are not subject to a full validation process.

MeteoGalicia's meteorologists have provided support for a human expert validation of the results, which has allowed us to refine the proposed solution in a way that ensures it works under realistic conditions and cases. For this, we have performed the following validation process:

1) **Dataset collection creation**. A collection of 45 forecast datasets was created by the meteorologists. This collection includes synthetic and real forecast data, which covers common as well as unusual meteorologic scenarios, similar to the ones presented in the examples in Section IV-A.
2) **Natural language forecast automatic generation**. From this collection of forecast datasets, 45 automatically generated natural language forecasts were obtained.
3) **Polishing stage**. These 45 natural language forecasts generated by our application were evaluated by a meteorologist who assessed their quality taking into account their most relevant aspects and dimensions of interest. This initial evaluation was made to obtain preliminary conclusions and polish our approach in those aspects which needed to be improved.
4) **Natural language forecast automatic generation**. Once the changes to our approach were implemented, new 45 automatically generated language forecasts have been obtained from the original collection of forecast datasets.
5) **Validation stage**. We have requested the expert to assess the new 45 automatically generated natural language forecasts. As opposed to the results from the polishing stage, which served to identify certain issues and potential improvements, the results of this stage allow to discern if the improvements in our approach are effective and, more importantly, if our application meets the expert's requirements and is consequently prepared to be released as a public service.

In order to assess the quality of the automatically generated forecasts, we have provided the expert meteorologist with a questionnaire which follows the approach presented in [40]. This questionnaire covers three key dimensions about the generated weather forecasts, as shown in Fig. 19:

- Relevance: Does the forecast include all the kind of information the expert would include?
- Truthfulness: Does the included information in the forecast reflect the numeric-symbolic forecast correctly?
- Manner: Does the forecast express the information properly? Is it well formatted?

These three dimensions are directly classified into two

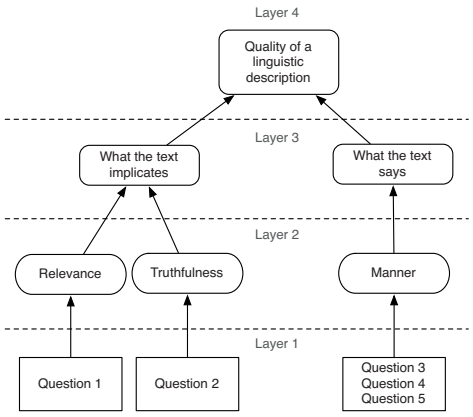


Fig. 19. Schema of the validation composition.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2014.2328011, IEEE Transactions on Fuzzy Systems

higher level categories, "what the text implicates" and "what the text says", which altogether determine the quality of the generated forecast. More specifically, the questionnaire we propose consists of five questions which deal in more depth with the previous three dimensions:

- **Question 1**: "Indicate in which degree you identify the type of results expressed as the type of results expressed by yourself: a) For sky coverage b) For precipitations c) For wind d) For temperatures".
  This question determines the grade in which an expert identifies the generated forecast with the ones he creates. For reasons of precision, and in order to identify more specific issues in each forecast variable, Question 1 was divided into four subquestions, one for each forecast variable.
- **Question 2**: "Do you agree with the provided descriptions? a) For sky coverage b) For precipitations c) For wind d) For temperatures".
  This question considers the degree of truthfulness of the generated description, this is, the degree in which the content of the forecast reflects faithfully the information within the numeric-symbolic forecast data. Similar to Question 1, Question 2 is divided into four subquestions. With the ratings of Questions 1 and 2, we obtain the partial rating of the forecast related to "what the text implicates".
- **Question 3**: "Indicate in which degree the vocabulary is used correctly".
  This question evaluates if the vocabulary from the meteorology domain is used properly.
- **Question 4**: "Indicate in which degree the content is correctly grouped to facilitate the comprehension of the description".
  This question evaluates if the information in the natural language description is properly grouped and not repetitive.
- **Question 5**: "Indicate in which degree the format of the report, including the punctuation, is the most adequate". Question 5 considers aspects related to the forecast text presentation, such as punctuation. With the ratings of Questions 3, 4 and 5 we obtain the partial rating "what the text says".

Each of these questions must be answered as a number in a 1-5 scale (from 1 "very negative" to 5 "very positive"). Thus, in order to calculate the global score for the collection of automatically generated forecasts, we follow the global aggregation schema defined in expression (2). Following this quality measure approach, the quality Q of an automatically generated natural language weather forecast S_i is defined as the arithmetic mean of the two dimensions in Layer 3 (Fig. 19):

$$Q_{S_i} = \frac{\frac{\overline{p_1}+\overline{p_2}}{2} + \frac{p_3+p_4+p_5}{3}}{2} \quad (2)$$

The terms $\overline{p_1}$ and $\overline{p_2}$ correspond to the average score of the subquestions a, b, c and d for Question 1 and Question 2, respectively. The remaining terms, p_3, p_4 and p_5 are the scores for Questions 3, 4 and 5. As 2 shows, the average of $\overline{p_1}$ and $\overline{p_2}$ ("what the text implicates") and the average of p_3, p_4 and p_5 ("what the text says") determine the quality of a forecast. Thus, the global quality score GQ for our collection of automatically generated natural language forecasts is obtained as the average of the validation cases quality score: $GQ = \sum_{i=1}^{n} \frac{Q_{S_i}}{n}$, where $n = 45$ in our case.

C. Results

One expert meteorologist answered the proposed questionnaire for the initial 45 automatically generated forecasts. Table I shows that, in general, the meteorologist's assessment about the content of the forecasts was very positive for the initial test (with an average global score (GQ) of 4.35 out of 5 and a deviation of 0.22). In this sense, the expert identified the content and language of the generated forecasts with the ones he would provide in a high degree. However, from each individual question score we could extract additional conclusions which, in general, implied that there was room for improvement, especially on Question 4 and on some variables from Question 1 and 2. This was due to several repetitive sentences produced by the NLG stage in some of the variables (especially precipitation) and to some expressions which were not appropriate for some variables.

Based on the results obtained for the polishing stage, we have improved the NLG modules to address the issues found in our first approach and a validation test has been performed by the meteorologist with new 45 automatically generated natural language forecasts. With an average score of 4.83 out of 5 and a deviation of 0.18 (as Table II shows), the quality increase is substantial. In particular, the results in Question 1 show that the expert fully identifies the automatically generated forecasts as if they were produced manually by him. The fact that both content and language from the automatic forecasts are almost

TABLE I
POLISHING STAGE QUESTIONNAIRE SCORE

Questions	Average score	Standard deviation
Q. 1 (a-d)	(3.6 3.93 5 4)	(0.45 0.75 0 0.57)
Q. 2 (a-d)	(4.04 4.44 5 4.86)	(0.36 0.5 0 0.34)
Q. 3	5	0
Q. 4	3.64	0.77
Q. 5	4.26	0.49
GQ	**4.35**	**0.22**

TABLE II
VALIDATION QUESTIONNAIRE SCORE

Questions	Average score	Standard deviation
Q. 1 (a-d)	(5 5 5 5)	(0 0 0 0)
Q. 2 (a-d)	(4.97 4.53 5 5)	(0.14 0.5 0 0)
Q. 3	5	0
Q. 4	4.64	0.48
Q. 5	4.53	0.50
GQ	**4.83**	**0.18**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2014.2328011, IEEE Transactions on Fuzzy Systems

indistinguishable from those that an expert would produce are the most important among the several quality aspects which can be measured for a NLG approach. The remaining Questions also show increased scores compared to the first assessment.

V. Application conceptualization

The solution we have presented addresses a specific practical problem by solving the need for providing 315 daily short term weather forecasts, which otherwise would not be possible to produce if they were manually created by a single meteorologist. As a consequence, the NLG stage is problem-oriented and is mostly not reusable. In spite of this, we want to stress the role that linguistic descriptions of data (LDD) techniques can play as a generic toolset which can be applied to many domains and give some insights into the generic methodology we are following for this LDD approach. For example, our application includes highly configurable linguistic description operators, which allow data series of any length and linguistic variables (implemented as fuzzy or crisp membership functions) with any number of labels as input. In fact, most of the changes made to improve the application during the whole development process were made to the linguistic variable definitions used by these operators (some of which are shown in Fig. 20) rather than to the operators themselves.

From our point of view, the main purpose of creating linguistic description solutions is to provide users with descriptions which make use of easily understandable familiar concepts found in natural language, imprecise and ambiguous in their nature. These concepts are usually modeled by employing some of the theoretical tools provided by the Computation With Perceptions field, such as fuzzy quantifiers, linguistic variables and others. However, the fact that these descriptions include linguistic terms neither implies they are actually expressed in natural language nor means they should be, as it occurs in NLG systems. In fact, both research fields seem rather complementary, in such a way that LDD provides tools for extracting the most relevant information in the form of (imprecise) linguistic terms, which then are used as an input to a NLG system to produce well-constructed sentences which are ready for human consumption. This is the approach we have followed in our solution, where LDD operators create input descriptions for an independent NLG system which generates natural language forecasts.

```
<partition name="V">
  <CrispInterval a="-100.0" b="-12.0" name="ED" mode="LeftClosed"/>
  <CrispInterval a="-12.0" b="-6.0" name="ND" mode="LeftClosed"/>
  <CrispInterval a="-6.0" b="-2.0" name="MD" mode="LeftClosed"/>
  <CrispInterval a="-2.0" b="-1.0" name="SD" mode="LeftClosed"/>
  <CrispInterval a="-1.0" b="1.0" name="WC" mode="Closed"/>
  <CrispInterval a="1.0" b="2.0" name="SI" mode="RightClosed"/>
  <CrispInterval a="2.0" b="6.0" name="MI" mode="RightClosed"/>
  <CrispInterval a="6.0" b="12.0" name="NI" mode="RightClosed"/>
  <CrispInterval a="12.0" b="100.0" name="EI" mode="RightClosed"/>
</partition>
<partition name="T">
  <CrispInterval a="-50.0" b="-2.0" name="VL" mode="LeftClosed"/>
  <CrispInterval a="-2.0" b="-1.0" name="L" mode="LeftClosed"/>
  <CrispInterval a="-1.0" b="1.0" name="N" mode="Closed"/>
  <CrispInterval a="1.0" b="2.0" name="H" mode="RightClosed"/>
  <CrispInterval a="2.0" b="50.0" name="VH" mode="RightClosed"/>
</partition>
```

Fig. 20. Crisp partition sets for temperature variation and climatic behavior as defined in the configuration document.

With a clearer view of which aspects LDD, in our opinion, should cover, we can abstract the basic elements which serve as pillars for a general LDD methodology. Many of the approaches described in the literature (e.g. those referred to in Section I) share several elements in common that can be taken into account for a flexible and reusable methodology for generating linguistic descriptions approaches:

- **Operators**. Operators extract information from raw data, converting numeric measurements into structures composed of linguistic terms. Originally, linguistic descriptions were conceived as quantified sentences, which resulted from applying fuzzy quantification models to data series. Therefore, many of the existing approaches use some kind of fuzzy quantification to obtain descriptions over one or several variables. For example, we can apply Zadeh's or other quantification models to produce a summary like "Most days of the month were dry" (in the case of rain data time series). However, many other operators which extract different pieces of information can be defined and implemented [29], such as: evaluation of a fuzzy label over the data series (e.g. "Most of the temperatures were high in March"), search of data sequences fulfilling a given fuzzy label (e.g. "Energy consumption was low between days 3 and 10"), search of increasing or decreasing patterns (e.g. "There was a slight increase of valve pressure during the morning"), search of pitches in the dataset or of oscillation patterns (e.g. "The system got unstable between 10:00 and 10:30"), event-counting operators (e.g. "There were too many high pitches within the last hours") or summarizing operators based on temporal/spatial hierarchies (e.g. "The month was hot but the first week was cold"). For instance, for our LDD approach we have created highly configurable operators for each weather variable, according to the type of information that we needed to extract. These operators can be applied straight-forwardly to other variables by just replacing the partition sets for the current variables with partition sets for the new ones.
- **Use of temporal/spatial hierarchies**. In the majority of cases, the numeric data series have an associated temporal and/or spatial component. This allows to arrange the data in hierarchies, which are usually defined by the experts in the application field. For example, in a temperature data series which covers one year, with one measurement per day, we can define a temporal hierarchy which would group the individual days in months, the months in seasons and so on. This considerably improves the exploitation of the available data, allowing to extract richer and more complex information. In our case, we have employed a time hierarchy which divides the short-term forecast temporal window into three subperiods for cloud coverage.
- **Operator compositions**. Operators can be considered as the core primitives or atomic logical units of a framework which generates linguistic descriptions. These units can

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2014.2328011, IEEE Transactions on Fuzzy Systems

be combined in order to build more complex descriptions, depending on the requirements of the specific linguistic description problem. Therefore, means for mixing their outputs should be taken into account as additional elements in our framework. Our LDD approach does not make use of this concept, since the linguistic descriptions we obtain for each variable are independent.

- **Evaluation criteria**. The raw output of linguistic description approaches usually consists of several candidate descriptions which must be filtered according to some pre-defined criteria, in order to ensure the quality and truthfulness of the selected final summary. Again, every specific problem needs its own set of adapted criteria, but also some general objective and reusable evaluation criteria, such as the description length, truth or fulfillment degree, data coverage, ambiguity, etc. should be used [29]. In our case, we have employed the aggregation of fulfillment degrees of each fuzzy subperiod with respect to each cloud coverage label to obtain the best cloud coverage for each subperiod. Furthermore, we have also used the length of descriptions in order to discriminate the final precipitation text forecast.

Although all of these are concepts and notions taken from experience, we believe the main value of this methodology lies in the operators as the building blocks of the LDD approaches. If a collection of well-tested both in quality and usefulness operators for linguistic description of data is gathered, the viability of a generic LDD framework to create domain-specific approaches is highly ensured. In order to achieve this, we propose a feedback process which combines bottom-up and top-down approaches. On one hand, we believe that the best way to ensure the usefulness of the operators is to generalize specific solutions taken from real life problems and test them in other contexts. On the other hand, intuition-based operators can also be proposed and tested to check whether the information they produce is relevant to the experts. This loop which goes from concrete to abstract and then vice versa would help to improve in a correct direction the general LDD framework.

VI. Conclusions and Future Work

We have presented GALiWeather, an application which obtains textual short-term weather forecasts for the 315 municipalities in Galicia, using the real data provided by MeteoGalicia. As opposed to other linguistic descriptions approaches, this solution is based on an applied development in a realistic application, whose definition and structure is inspired by the linguistic descriptions research field by using both fuzzy and crisp operators which extract relevant information, and also by the natural language generation field.

Furthermore, the automatically generated textual forecasts were thoroughly evaluated by a meteorologist in order to assess the quality of their contents and to check whether his expert knowledge was included correctly. The obtained results show that the textual forecasts fulfill the expert's requirements in a very high degree (4.83 out of 5). GALiWeather is to be released as a real service in a very near future, since the application fully meets the meteorologists' requirements. The automatic linguistic descriptions will be displayed as a new information service at MeteoGalicia's website [33].

The main value of GALiWeather resides in its ability to cover and support a service of high interest for a wide number of users, which can only be provided by generating descriptions of data in an automatic manner, due to the high number of textual forecasts (315 in this case) which must be obtained.

In a longer term we are considering other application fields in which linguistic descriptions will prove useful. Among them, we have identified linguistic descriptions on information and decision support environmental systems as a promising research line, where not only linguistic descriptions for single location data are interesting, but also descriptions which geographically aggregate data in order to provide region-wide information. This is a complex challenge which will include the description of data in both time and space dimensions. This will lead us to develop a general model which can be applied to application fields in other areas.

Acknowledgments

The authors would like to thank the editors and referees for their comments and suggestions, which have led to a substantial improvement in the paper quality. We would also like to thank CITIUS and MeteoGalicia for their support and for providing personal and material means for the development of this application.

References

[1] E. Comission. (2011, December) Open data. an engine for innovation, growth and transparent governance. [Online]. Available: http://www.ipex.eu/IPEXL-WEB/dossier/document/COM20110882.do

[2] N. Kroes. (2012, March) Speech/12/149 digital agenda and open data from crisis of trust to open governing. Presentation of the Action Plan of the Slovak Republic in favour of Open Democracy. [Online]. Available: http://europa.eu/rapid/press-release_SPEECH-12-149_en.htm

[3] U. Fayyad, G. Piatetsky-shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI Magazine*, vol. 17, pp. 37–54, 1996.

[4] A. Fu, "Data mining," *Potentials, IEEE*, vol. 16, no. 4, pp. 18–20, 1997.

[5] K. Kukich, "Design and implementation of a knowledge-based report generator," in *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics (ACL-1983)*, 1983, pp. 145–150.

[6] L. Iordanskaja, M. Kim, R. Kittredge, B. Lavoie, and A. Polguère, "Generation of extended bilingual statistical reports," in *Proceedings of the 14th International Conference on Computational Linguistics (COLING-1992)*, vol. 3, 1992, pp. 1019–1023.

[7] M. T. Maybury, "Generating summaries from event data," *Information Processing And Management*, vol. 31, no. 5, pp. 735 – 751, 1995.

[8] S. Busemann and H. Horacek, "Generating air-quality reports from environmental data," in *DFKI Workshop on Natural Language Generation, DFKI Document D-97-06*, S. Busemann, T. Becker, and W. Finkler, Eds., 1997.

[9] S. Boyd, "Trend: A system for generating intelligent descriptions of time-series data," in *Proceedings of the IEEE International Conference on Intelligent Processing Systems (ICIPS-1998*, 1998.

[10] F. Portet, E. Reiter, A. Gatt, J. Hunter, S. Sripada, Y. Freer, and C. Sykes, "Automatic generation of textual summaries from neonatal intensive care data," *Artif. Intell.*, vol. 173, no. 7-8, pp. 789–816, May 2009.

[11] E. Goldberg, N. Driedger, and R. Kittredge, "Using natural-language processing to produce weather forecasts," *IEEE Expert*, vol. 9, no. 2, pp. 45–53, 1994.

[12] J. Coch, "Interactive generation and knowledge administration in multimeteo," in *Proceedings of the Ninth International Workshop on Natural-Language Generation (INLG-1996)*, 1998, pp. 300–303.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2014.2328011, IEEE Transactions on Fuzzy Systems

[13] E. Reiter, S. Sripada, J. Hunter, and I. Davy, "Choosing words in computer-generated weather forecasts," *Artificial Intelligence*, vol. 167, pp. 137–169, 2005.

[14] S. Sripada, E. Reiter, and I. Davy, "Sumtimemousam: Configurable marine weather forecast generator," *Expert Update*, vol. 6, no. 3, pp. 4–10, 2003.

[15] L. A. Zadeh, "Fuzzy logic = computing with words," *Fuzzy Systems, IEEE Transactions on*, vol. 4, no. 2, pp. 103–111, 1996.

[16] ——, "From computing with numbers to computing with words : From manipulation of measurements to manipulation of perceptions," in *Intelligent Systems and Soft Computing: Prospects, Tools and Applications*. Springer-Verlag, 2000, pp. 3–40.

[17] J. Mendel, "The perceptual computer: an architecture for computing with words," in *Fuzzy Systems, 2001. The 10th IEEE International Conference on*, vol. 1, 2001, pp. 35–38.

[18] R. R. Yager, K. M. Ford, and A. J. Cañas, "An approach to the linguistic summarization of data," in *Uncertainty in Knowledge Bases, 3rd International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 90, Paris, France, July 2-6, 1990, Proceedings*, ser. Lecture Notes in Computer Science, B. Bouchon-Meunier, R. R. Yager, and L. A. Zadeh, Eds., vol. 521. Springer, 1990, pp. 456–468.

[19] J. Kacprzyk and S. Zadrozny, "Linguistic database summaries and their protoforms: towards natural language based knowledge discovery tools," *Inf. Sci. Inf. Comput. Sci.*, vol. 173, no. 4, pp. 281–304, 2005.

[20] G. Trivino and M. Sugeno, "Towards linguistic descriptions of phenomena," *International Journal of Approximate Reasoning*, vol. 54, no. 1, pp. 22 – 34, January 2013.

[21] R. Castillo-Ortega, N. Marín, and D. Sánchez, "A fuzzy approach to the linguistic summarization of time series," *Multiple-Valued Logic and Soft Computing*, pp. 157–182, 2011.

[22] A. van der Heide and G. Trivino, "Automatic generated linguistic summaries of energy consumption data," in *Proceedings of 9th ISDA Conference*, 2009, pp. 553–559.

[23] D. Sanchez-Valdes, L. Eciolaza, and G. Trivino, "Linguistic description of human activity based on mobile phone's accelerometers," in *Ambient Assisted Living and Home Care*, ser. Lecture Notes in Computer Science, J. Bravo, R. Hervás, and M. Rodríguez, Eds. Springer Berlin Heidelberg, 2012, vol. 7657, pp. 346–353.

[24] A. Alvarez-Alvarez and G. Trivino, "Linguistic description of the human gait quality," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 13 – 23, 2013.

[25] I. Kobayashi and N. Okumura, "Verbalizing time-series data: With an example of stock price trends," in *Proceedings IFSA/EUSFLAT Conf.*, 2009, pp. 234–239.

[26] J. Kacprzyk, "Computing with words is an implementable paradigm: Fuzzy queries, linguistic data summaries, and natural-language generation," *IEEE Trans. Fuzzy Systems*, pp. 451–472, 2010.

[27] J. Kacprzyk and A. Wilbik, "Using fuzzy linguistic summaries for the comparison of time series: an application to the analysis of investment fund quotations," in *Proceedings IFSA/EUSFLAT Conf. 2009*, 2009, pp. 1321–1326.

[28] J. Kacprzyk and S. Zadrozny, "Linguistic data summarization: A high scalability through the use of natural language?" *Scalable Fuzzy Algorithms for Data Management and Analysis: Methods and Design*, pp. 214–237, 2010.

[29] F. Díaz-Hermida, A. Ramos-Soto, and A. Bugarín, "On the role of fuzzy quantified statements in linguistic summarization," in *Proceedings of 11th International Conference on. Intelligent Systems Design and Applications (ISDA)*, 2011, pp. 166–171.

[30] R. Castillo-Ortega, N. Marín, D. Sánchez, and A. Tettamanzi, "Quality assessment in linguistic summaries of data," in *Advances in Computational Intelligence*, ser. Communications in Computer and Information Science, S. Greco, B. Bouchon-Meunier, G. Coletti, M. Fedrizzi, B. Matarazzo, and R. Yager, Eds. Springer Berlin Heidelberg, 2012, vol. 298, pp. 285–294.

[31] C. Menendez and G. Trivino, "Selection of the best suitable sentences in linguistic descriptions of data," in *Advances in Computational Intelligence*, ser. Communications in Computer and Information Science, S. Greco, B. Bouchon-Meunier, G. Coletti, M. Fedrizzi, B. Matarazzo, and R. Yager, Eds. Springer Berlin Heidelberg, 2012, vol. 298, pp. 295–304.

[32] GALiWeather reference website. [Online]. Available: http://citius.usc.es/transferencia/demostradores-tecnoloxicos/GALiWeather

[33] MeteoGalicia website. [Online]. Available: www.meteogalicia.es

[34] L. A. Zadeh, "A computational theory of dispositions," *International Journal of Intelligent Systems*, vol. 2, no. 1, pp. 39–63, 1987.

[35] E. Reiter and R. Dale, *Building Natural Language Generation Systems*. Cambridge University Press, 2000.

[36] ——, "Building applied natural language generation systems," *Journal of Natural-Language Engineering*, no. 3, pp. 57–87, 1997.

[37] E. Reiter, "Task-based evaluation of nlg systems: control vs real-world context," in *Proceedings of the UCNLG+Eval: Language Generation and Evaluation Workshop*, ser. UCNLG+EVAL '11, 2011, pp. 28–32.

[38] A. Belz and E. Reiter, "Comparing automatic and human evaluation of nlg systems," in *In Proc. EACL'06*, 2006, pp. 313–320.

[39] R. Sambaraju, E. Reiter, R. Logie, A. McKinlay, C. McVittie, A. Gatt, and C. Sykes, "What is in a text and what does it do: qualitative evaluations of an nlg system – the bt-nurse – using content analysis and discourse analysis," in *Proceedings of the 13th European Workshop on Natural Language Generation*, ser. ENLG '11. Association for Computational Linguistics, 2011, pp. 22–31.

[40] L. Eciolaza, M. Pereira-Fariña, and G. Trivino, "Automatic linguistic reporting in driving simulation environments," *Applied Soft Computing*, vol. 13, no. 9, pp. 3956 – 3967, 2013.

Alejandro Ramos received the M.S.c. degree in computer science from the University of Santiago de Compostela (USC), Spain, in 2011. He is currently a Ph.D. student at its Research Centre on Information Technologies (CiTIUS). His research interests include Linguistic Descriptions of Data and Natural Language Generation.

Alberto J. Bugarín received the Ph.D. degree in physics from the University of Santiago de Compostela (USC), Spain, in 1994. He is currently a Full Professor at its Research Centre on Information Technologies (CiTIUS). His research interests mainly focus on Linguistic Data Description of Data using Natural Language Generation, Machine Learning techniques for fuzzy knowledge bases discovery and Fuzzy Temporal knowledge representation and reasoning. On these and related topics and their applications he has published more than 150 scientific refereed papers and participated in more than 40 R+D projects and contracts.

Senén Barro received the Ph.D. in physics with distinction from the University of Santiago de Compostela (USC), Spain, in 1988. He is Professor in the area of Computer Science and Artificial Intelligence. He was head of the Computer and Electronic Department of the University of Santiago de Compostela from 1993 to 2002, and the rector of this university from 2002 to 2010. Since May 2008 he is the president of RedEmprendia, which is made of 24 European and Latin American universities, focused on transfer on R&D, innovation and entrepreneurship. He founded the USC Intelligent Systems Group, which he also directs, and which currently has more than 40 members and is one of the first Artificial Intelligence groups in Spain. He has been editor or author of seven books and author of more than 200 scientific articles. He has also been member of organizing, scientific and publishing committees of international conferences and journals.

Juan Taboada received the Ph.D. Degree in physics from the University of Santiago de Compostela (USC), Spain, in 1999, after a two-year stage in the University of Paris VI. He currently leads the operational weather forecast department in MeteoGalicia, the Galician (NW Spain) Meteorological Agency. His research areas include climate variability and change, and seasonal and weather forecasting.