Available online at www.sciencedirect.com

ScienceDirect

Fuzzy Sets and Systems 443 (2022) 173–202

FUZZY sets and systems

www.elsevier.com/locate/fss

Meta-heuristics for generation of linguistic descriptions of weather data: Experimental comparison of two approaches

Andrea Cascallar-Fuentes *, Alejandro Ramos-Soto, Alberto Bugarín

Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela, Spain

Received 2 September 2019; received in revised form 17 September 2021; accepted 22 February 2022
Available online 2 March 2022

Abstract

In this paper we experimentally assess, from both algorithmic and pragmatic perspectives, the adequacy of linguistic descriptions of real data generated by two metaheuristics: simulated annealing and genetic algorithm meta-heuristics. The type of descriptions we consider are fuzzy quantified statements (both Zadeh's type-1 and type-2) involving three well-known quantification models (Zadeh's scalar and fuzzy and Delgado's GD). We conducted an empirical validation using real observation and prediction meteorological data, where both automatic (metrics-based) and manual (human experts-based) assessment on the adequacy of the generated descriptions was assessed. Results indicate that, overall, the genetic approach performs better than simulated annealing in terms of quality of the obtained descriptions and time execution. Significance of this outperforming depends on the type of meteorological data and the quantification model selected. Tests of statistical significance point out that for type-1 descriptions no significant differences exist between the two meta-heuristics in the prediction case. For type-2 descriptions, significant differences exist for Delgado's GD model for both types of data. For Zadeh's scalar and fuzzy quantification significance depends on the type of data (observation or prediction). Globally, outperforming of the genetic approach over simulated annealing i) is significant in 4 out of 12 scenarios considered (all of them type-2), and ii) is not significant in the other 8 out 12 scenarios (all type-1 and two type-2). Also human expert assessment on the adequacy of the descriptions was conducted, showing that both meta-heuristics behave similarly for type-1 descriptions, while genetic algorithms produce more suitable type-2 linguistic descriptions.

Keywords: Linguistic descriptions of data; Data-to-text systems; Computing with words; Natural language generation

1. Introduction

Nowadays, analysis and interpretation of data in professional settings is becoming increasingly difficult, as the amounts of data to be processed grow exponentially and exceed the capabilities of human experts. In these circumstances, computational methods and systems that can perform such tasks are in high demand.

* Corresponding author.
E-mail addresses: andrea.cascallar.fuentes@usc.es (A. Cascallar-Fuentes), alejandro.ramos@usc.es (A. Ramos-Soto), alberto.bugarin.diz@usc.es (A. Bugarín).

https://doi.org/10.1016/j.fss.2022.02.016

Within the natural language generation (NLG) field, data-to-text (D2T) systems [1] automatically generate texts from large numerical or symbolic data sets, providing human-friendly comprehensible information that could not be produced otherwise. D2T systems include: i) a data analysis stage where the relevant information is extracted from data and ii) a generation stage where information is interpreted and conveyed to the user in natural language.

Many data-to-text systems have been developed for different application domains based on the techniques described above [2]. For example, the BT45 system [3] generates reports from the collected data from babies in an Intensive Care Unit. The meteorology field has also been a recurrent case study with some remarkable systems [4–8].

Also in the fuzzy logic field several approaches were proposed to generate data descriptions using linguistic terms. Seminal work by Zadeh introduced the computing with words paradigm [9–11], where computations are performed on linguistic terms modeled as fuzzy sets, and its evolution, computing with perceptions [12,13]. Within these frameworks, one approach is linguistic descriptions of data (LDD) [14,15] which summarize in a linguistic form one or more numerical variables and their values, using the general notion of protoform [16] and more specifically, fuzzy quantified sentences which can follow several structure types (e.g. "In some locations the temperature is low").

Linguistic descriptions may lack in general the expressiveness of real Natural Language texts, but are nonetheless useful information items that can be used as high-level input to NLG systems in general and D2T in particular. For example, GALiWeather [17] is a meteorological D2T system that uses LDD to extract linguistic information, which is then verbalized under two different output languages. In [18] an NLG system which also uses LDD to generate reports for saving energy at home is proposed.

Approaches for LDD are mostly based on type-1 and type-2 quantified sentences [19–23]. Type-1 descriptions have the following structure: "Q Y are S" (e.g. "In many locations the temperature is low") where Q is a quantifier (*many* in the example), Y is a linguistic variable (*temperature* in the example) and S a linguistic value of Y (*low* in the example). Type-2 descriptions have the following structure: "Q KY are S" (e.g. "In many locations *in the North* the temperature is low"), where an additional qualifier K is included (*in the North* in the example). However, in other domains, for instance, in time series description [24–26], health [21,27], process management [28], elder care [29] or probabilistic uncertainty consideration [30]; other protoforms different from quantified propositions are proposed.

The problem of finding an appropriate LDD on a given data set can be considered as a search problem where candidate protoforms need to be generated and assessed on the data set. In many applications, the data set size together with the number of linguistic elements in the protoform (variables, values,...) make it unfeasible to explore all possible combinations. Because of this, some authors have proposed approaches based on heuristic or meta-heuristic (genetic algorithms) search techniques for finding relevant sentences (according to predefined criteria) in those cases where the number of possible descriptions is so large that the search space cannot be fully explored efficiently. Heuristics are usually problem-dependent search techniques which guarantee finding a good solution in a reasonable amount of time, generally worse than the optimal. Heuristic strategies were proposed, for instance, for time series description [31] or the meteorological prediction generation in natural language [17]. On the other hand, a meta-heuristic is a high-level problem-independent algorithm which optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. In [32] a multi-objective evolutionary algorithm was proposed for the times series description with the aim of generating the shortest and most precise summary that covered the time dimension. Also a genetic algorithm was proposed in [33] for generating linguistic summaries about operational and financial healthcare data. Another bio-inspired approach for linguistic summaries generation is the Ant Colony Optimization model proposed in [34].

In general, the problem of generating LDD in contexts where the search space and/or the data size become too large did not receive much attention in the literature. To the best of our knowledge, most approaches (both heuristic and meta-heuristic) rely on a single algorithm, and no comparisons with other alternatives are made in terms of performance or relevance of the generated sentences. Furthermore, no significance analysis were reported in the comparison of the proposed algorithms.

In this work, we are focused on generating linguistic descriptions of weather data using quantified sentences. The data we worked with described the weather situation in Galicia (NW Spain) with data provided by the Galician Meteorological Agency, MeteoGalicia [35]. As in the approaches mentioned before, the generation of all the possible LDD is not feasible in this case. Therefore, we followed a meta-heuristic search strategy for generating LDD, by defining two different search algorithms. On one hand, following [36] we designed a genetic algorithm. On the other hand, after analyzing different meta-heuristic algorithms, we also designed a solution based on Simulated Annealing

[37] as an alternative, since this meta-heuristic was reported to achieve better results than other algorithms for several tasks [38,39].

We also performed a comparison of the two approaches, both considering the relevance of the LDD they obtain as well as the algorithms' performance. Also, tests were conducted to assess the statistical significance of the experimental results obtained. To the best of our knowledge, the problem presented in this paper is not tackled in the literature in these wide terms: model definitions, testing and significance analysis.

This paper is structured as follows: in Section 2 the context of the problem is described. In Section 3 we present the protoforms we developed and the two meta-heuristic algorithms we designed for obtaining linguistic descriptions of meteorological observation and prediction data. In Section 4 we present the experimental comparison between our two models in terms of their performance with different metrics and the significant differences between them. This paper finishes with concluding remarks presented in Section 5.

2. Problem context

A meteorological situation is defined by complex numerical models that are not easily understandable, mainly for non-expert users. Furthermore, the characteristics of these models could even make their manual analysis a challenging task for experts in the field, taking into account the high number of variables/values usually described or the extension of the territory under consideration. Even the usage of maps may not be intuitive due to the amount and different shapes of the icons used for representing the weather situations. In the example presented in Fig. 1, maps show a meteorological real-state of Galicia for three variables: sky state, wind and temperature. Both the sky and wind icons are the standard ones used by the Galician Meteorology Agency (MeteoGalicia) whereas in the temperature maps the possible values are: VL (very low), L (low), N (normal), H (high), VH (very high) printed in different colors from red, associated to high temperatures, to dark blue, associated to low temperature values. Despite generating one map per variable, understanding what they represent and producing a description of the weather situation is not easy, especially for the wind variable, whose values graphic representation are not easily distinguishable.

MeteoGalicia provides, among others, two different related types of data: real-time data from 313 Galician municipalities (Fig. 2) and numerical forecast data from a 4 kilometers grid which covers the entire Galician territory and its adjacent regions, a total of 3,363 locations (Fig. 3).

These services have very different data update frequencies: the observation service offers data collected in real time with a very high update frequency, every 10 minutes, so the effectiveness of the search method is very important (i.e., it should produce very high-scored descriptions in near real time). On the other side, prediction data has a 12-hour update frequency. Therefore, in this case the time requirements are not as demanding as in the observation case, and, therefore, the effectiveness of the search method is not so relevant.

From the data point of view, the main difference between real-time observation data and numerical forecast data is the data sets size, since real-time observation data is available from 313 locations whereas the prediction data provides information from a total of 3,363 locations (one order of magnitude larger). This is a critical feature as the size of the data set affects the efficiency of the linguistic description search process since evaluating the fulfillment degree of a description involves evaluating, for each location, the degree to which it meets a given condition i.e. the computational cost of evaluating a quantified sentence increases with the number of locations to be checked.

Using a greedy approach, the amount of data to be processed in the generation consumes approximately 8 hours in the observation case and more than 2 days in the prediction one. Since this approach is excessively time consuming, we defined a time frame requisite for each case to ensure the time frame of our approaches is shorter than the data update frequency.

Besides, each data source provides the information under a different format: while the observation data is provided as a JSON format, the prediction data is presented in a netCDF file so each data is processed differently in order to generate a common input to our approach.

The set of described variables and their data type are also different. We select three present variables (sky state, wind and temperature) in both sources. Both sky and temperature do not need a transformation since the sky state is represented with the same 42 codes and the temperature is provided as a numerical value. However, the wind variable did not follow the same format. In the observation case, this variable is represented with 34 codes in the range [299, 322] with defined direction and speed.

(a) Sky state (b) Wind (c) Temperature

Fig. 1. Meteorological state. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

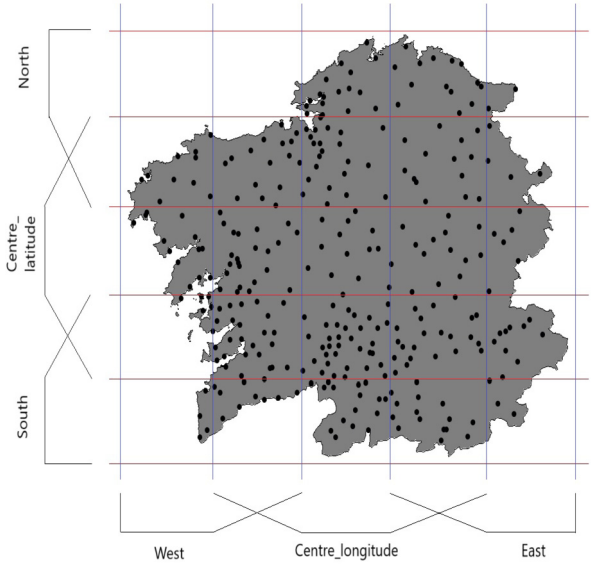


Fig. 2. Described locations for the weather observation service provided by MeteoGalicia (one observation point per municipality).

In the prediction case this variable is described by two different variables: *dir*, which represents the direction of the wind in grades and *mod*, which represents the wind speed in m/s. So a preprocessing step is performed in order to obtain the corresponding code.

In both cases, we aim to generate descriptions that cover the following meteorological variables: state of sky, wind and temperature. Some of these concepts do not have a crisp definition that establishes precisely their limits according to how both users and experts use them. Therefore, the need arises to use techniques based on fuzzy logic to define from them fuzzy linguistic variables that allow us to generate imprecise expressions.

2.1. Sky state

This variable describes the state of the sky based on two dimensions: cloud coverage and rainfall. MeteoGalicia's meteorologists labeled the values of this variable with 42 codes, which are integer numbers. Half of these values (21 integer numbers in the range [101, 121]) are used to describe the day situation (e.g. code 101 means "clear sky"), using terms such as, for example, "cloudy", "rainy" or "storm with few clouds". Also the same 21 different situations

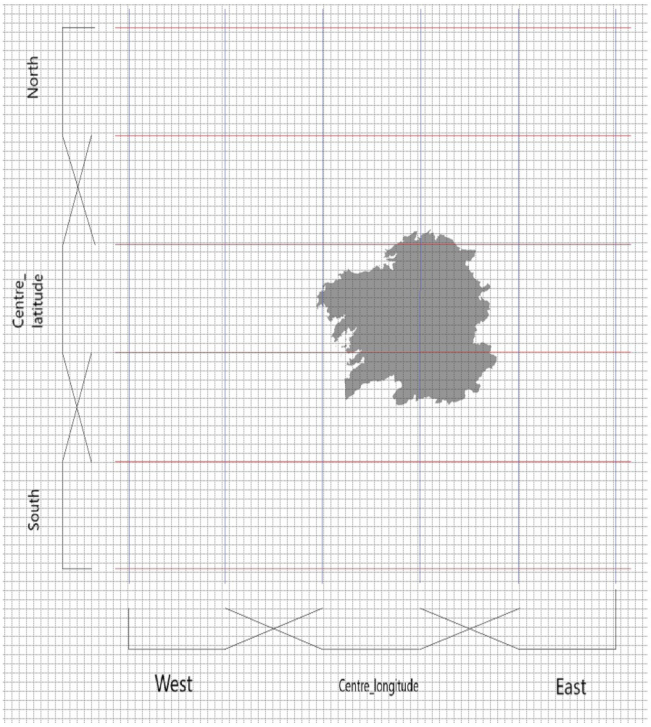


Fig. 3. Described locations for the weather prediction service provided by MeteoGalicia. Each point is the intersection of the grid lines (being each cell 4 km. x 4 km. size).

are described for the night, which are coded with 21 integer numbers in the range [201, 221] are used (e.g. 211 means "night with clear sky").

2.2. Wind

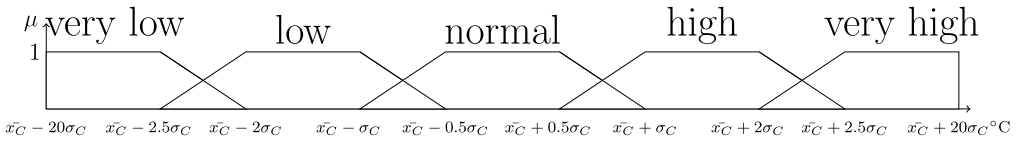
This is a variable that comprises the wind direction and speed, and is labeled with 34 codes, which are integer numbers in the range [299, 332]. Meteorologists consider eight wind directions (N, S, E, W, NW, NE, SE, SW) combined with four wind speed values (weak, moderate, strong, very strong). Also the calm and variable direction situations are considered. For instance, 305 code means "South direction and weak speed".

2.3. Temperature

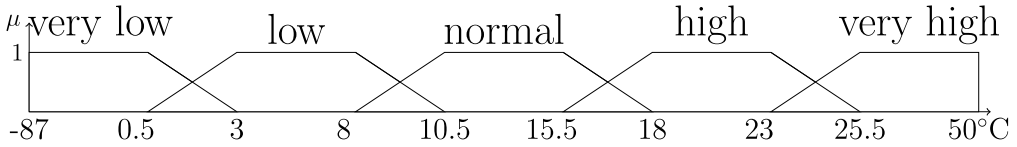
This is a numerical variable that represents the temperature in degrees Celsius. We model this variable through a fuzzy linguistic variable with five linguistic values: {"very low", "low", "normal", "high", "very high"}, which are numerically defined as fuzzy sets. According to the meteorologists criteria, information related to temperature is always provided taking a reference value for comparison (e.g., normal temperatures are those that are similar to the reference one). This reference value is defined differently for observation and prediction, because of the availability of historical data.

For the observation case, historical data are available for each point and each month of the year. Therefore, we take as reference the historical average temperature for the last twenty years (this is called the "climatic" temperature) $\bar{x}_C$, and its standard deviation, σ_C and define the linguistic terms as indicated in Fig. 4a. For instance, the label "normal" is defined with the trapezoid with support $[\bar{x}_C - \sigma_C, \bar{x}_C + \sigma_C]$ and core $[\bar{x}_C - 0.5\sigma_C, \bar{x}_C + 0.5\sigma_C]$. So, for a specific point with, for example, $\bar{x}_C = 14.2$ and $\sigma_C = 4.8$, its "normal" label has support [9.4, 19] and core [11.8, 16.6].

For the prediction case, the historical data is not available, although historical daily data are accessible. Therefore, we generate a data set which covers all months of a year and we took as reference the average of the temperatures in this data set $\bar{x}_D$, and its standard deviation, σ_D for defining the linguistic terms as indicated in Fig. 4b. For instance,



(a) Observation use case.



(b) Prediction use case ($\bar{x}_D = 13°C$ and $\sigma_D = 5°C$).

Fig. 4. Definition of the linguistic terms of temperature.

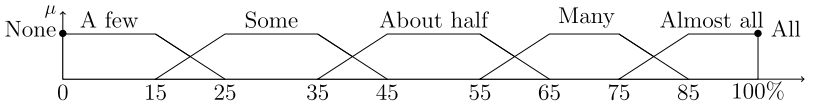


Fig. 5. Observation and prediction quantifiers definition for the whole territorial coverage (percentage of locations in the map).

the label "normal" is defined with the trapezoid with support [8, 18] and core [10.5, 15.5], since $\bar{x}_D = 13°C$ and $\sigma_D = 5°C$ for the data set available.

2.4. Quantifiers

As we mentioned above, LDD include sentences as "In some locations the wind has North direction and moderate speed", so quantifiers are necessary to count the relative amount (percentage) of locations in the map (territorial coverage) that satisfy a given meteorological condition. We define seven fuzzy quantifiers {"None", "A few", "Some", "About half", "Many", "Almost all", "All"} defined as trapezoidal fuzzy sets as shown in Fig. 5.

Besides, from their definition we define the concept of "coverage" [40] which represents the percentage of locations it covers, having "None" the minimum because it covers 0% of the territory whereas the maximum is assigned to "All", which covers 100%.

2.5. Geographical descriptors

For type-2 sentences we add a geographical qualifier. This allows us to describe smaller regions instead of the whole territory as in type-1 descriptions. For instance, "In some locations in the North the temperature is low".

We define 9 linguistic term descriptors (North, South, East, West, Center, North-east, North-west, South-east and South-west), using longitude and latitude as the universe of discourse. Fig. 6 shows the location of the territory regarding its meridians and parallels. These descriptors are defined with different values in observation (Fig. 7) and in prediction (Fig. 8). In the Figs. 2 and 3 the descriptors are represented in order to show graphically the differences between the definitions. The reference values are adapted to cover only the Galician territory in the observation, but in prediction, since forecast data covers a larger extension, the descriptors cover larger pieces of territory.

The descriptors can be classified as simple or composite depending on the number of dimensions they are defined upon (one or two). Composite descriptors are defined as AND combinations of two one-dimensional descriptors for longitude and latitude. For instance, "Center" is defined by combining "Center_latitude" and "Center_longitude". When evaluating the position of a point in for "Center", "Center_latitude" and "Center_longitude" are evaluated independently and their result is combined using the minimum. The rest of the composite geographical descriptors (Northeast, Northwest, Southeast, Southwest) are defined in the same way.

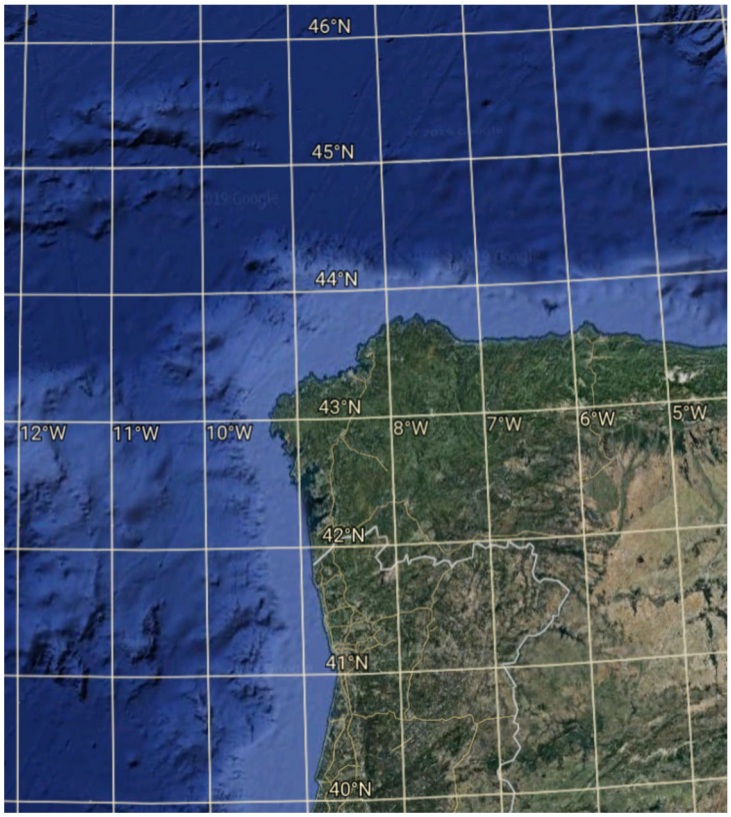
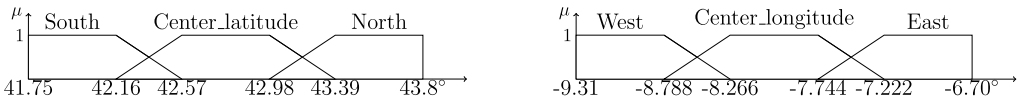
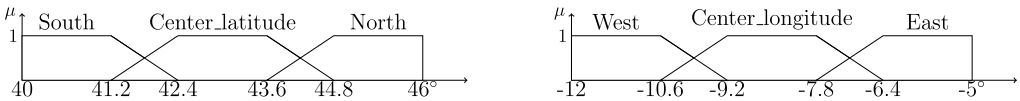


Fig. 6. Reference coordinates from Google Earth [41].



(a) Latitude definition. (b) Longitude definition.

Fig. 7. Geographical descriptors in the observation use case.



(a) Latitude definition. (b) Longitude definition.

Fig. 8. Geographical descriptors in the prediction use case.

3. Materials and methods

In the observation case we construct a data set composed of 20 records between July 2017 and January 2019. The data was randomly collected to construct a data set that contains data from different times of the day. For prediction, we extract one random case randomly from each month in 2018, resulting in a data set composed of 12 cases.

Table 1 shows all the possible values for the previously defined linguistic variables. In the style guide of Meteo-Galicia, the sky state variable is composed by two parts: night, which is optional, and day. When a night situation is described we use the *"night with"* + *value of column day* structure whereas when a day situation is described the night

Table 1
Linguistic variables values.

Q	G		A			
% locations	latitude	longitude	sky state		wind		temperature
			night	day	direction	speed	
None	Null	Null		Null	Null		Null
A few	North	West		Clear sky	Calm		Very low
Some	Center_latitude	Center_longitude		High clouds	Variable		Low
About half	South	East		Clouds and clear	North	Low	Normal
Many				Very cloudy	South	Moderate	High
Almost all				Covered sky	East	Strong	Very high
All				Fog	West	Very strong	
				Rain shower	North-east		
				Rain shower 75%	North-west		
				Snow shower	South-east		
			[Night with]	Dew	South-west		
				Rain			
				Snow			
				Storm			
				Mist			
				Fog banks			
				Mid-level clouds			
				Light rain			
				Light showers			
				Storm with few clouds			
				Sleet			
				Hail			

Table 2
Template to describe each meteorological variable in the generated descriptions.

Linguistic variable	Template
Sky state	the sky state is <value>
Wind	wind direction is <direction_value> and its speed is <speed_value>
Temperature	the temperature is <value>

part is omitted. For instance, code 201 means “night with clear sky” whilst code 101 means “clear sky”. Nevertheless, a similar situation during the day is represented simply with code 101 (“clear sky”)

Besides, the wind variable also is composed by two parts: direction and speed, *value of column direction + value of column speed*, except in three special cases which cannot be composed by the defined direction and speed: no data where the direction and speed value is “Null”, calm where their value is “Calm” and variable where the value is “Variable”.

Based on all these possible values for the linguistic variables, 632,030 different sentences can be generated. As indicated before, type-1 and type-2 quantified sentences are generated.

This approach is focused on the content determination and realization stages of the usual NLG pipeline [1]. For each approach, Fig. 9 shows the execution stages with their inputs and outputs, described as follows:

- Description components: the input of this stage is the meteorological data with the format described in Section 2 and the predefined data to generate the linguistic variable. The output of this step is the defined variables (Q, G, S, W, T).
- Description generation: in the generation process, a description is represented as a tuple of elements. For instance, “all North 101 303 normal” is the intermediate representation of the description “In all locations in the North the sky is clear, the wind has East direction and low speed and the temperature is normal”. In this step we generate new descriptions as candidates solution where each of their components is selected randomly.

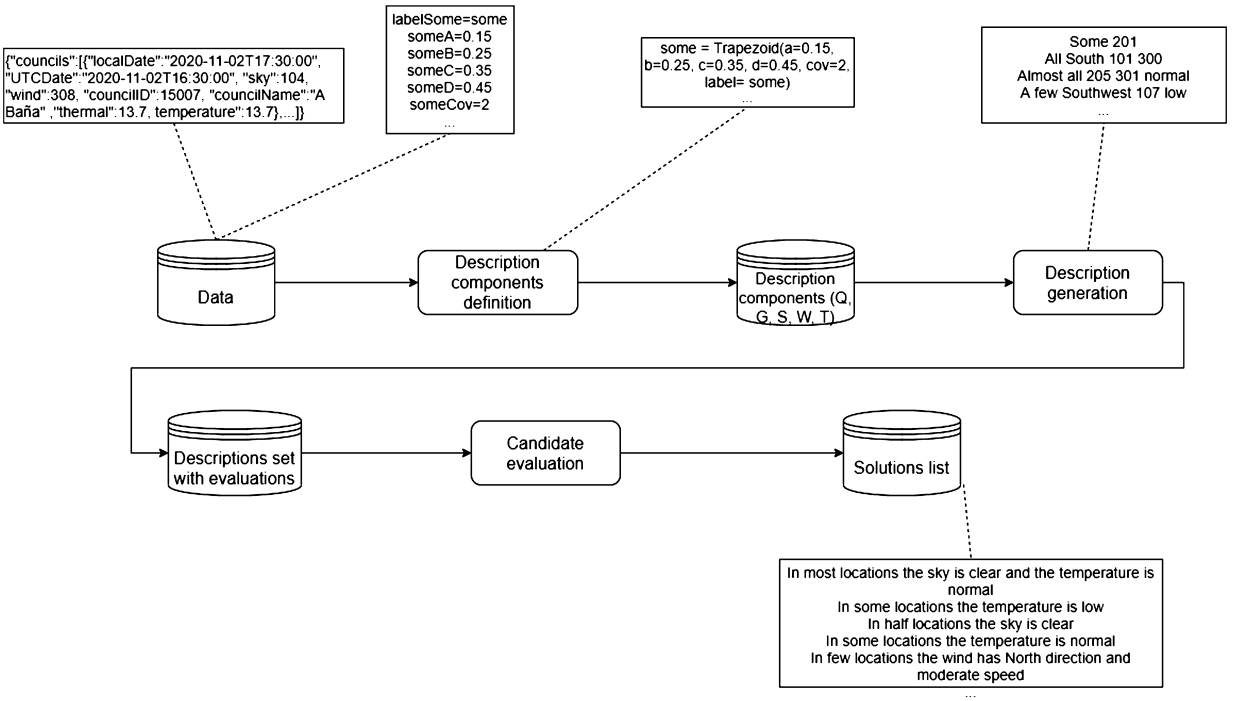


Fig. 9. Description of the experimentation stages with their inputs and outputs.

- Candidate evaluation: in this stage we evaluate the candidate description to accept, and add it to the solutions list if accepted. We store all the generated candidates in a set of candidates so the first evaluation is checking if a new candidate has already been evaluated. If a candidate has been previously generated it is discarded to avoid evaluating already assessed descriptions. The output of this stage is the set of accepted solutions verbalizing them with their corresponding, type-1 or type-2, structure through the templates defined in Table 2.

3.1. Descriptions assessment

Once we generate a sentence, we need to assess how descriptive it is in terms of the meteorological situation described by the data. For this we have considered a combination of three different criteria: the truth value, the coverage, and the length of the sentence, as described in what follows.

3.1.1. Truth value

We calculate the truth value of the quantified sentences ($\mu \in [0, 1]$) using three widely used quantification models: Zadeh's scalar quantification model [42], Delgado's GD quantification method [43] and Zadeh's fuzzy cardinality method (ZS) [44].

The selection of these methods is based on a previous study [45] where we compared the behavior of a set of quantification models and the features of the defined quantifiers to ensure the suitability of the methods for this evaluation.

Zadeh's quantification method Equation (1) shows the calculation for type-1 sentences involving a map with n observation/prediction locations with a quantifier Q, a set of linguistic values S for each linguistic variable and for each point in the map $x_i,\ i = 1, ..., n$.

$$Z_Q(S) = Q\left(\frac{P(S)}{|X|}\right) \tag{1}$$

where Q is a linguistic quantifier and $P(S) = \sum_{i=1}^{n} S(x_i)$ the Zadeh's scalar cardinality $\sum -count$ also called power [43], as defined in [42].

Equation (2) describes the calculation for type-2. Here, K represents the linguistic qualifier of the type-2 quantified statements.

$$Z_Q(S/K) = Q\left(P(S/K)\right) = Q\left(\frac{P(S \cap K)}{P(K)}\right) \tag{2}$$

Delgado's GD method The GD method is a quantification model of the so-called G-family. The evaluation of a type-1 quantified sentence with n observation/prediction locations with a quantifier Q, a set of linguistic values S for each linguistic variable and for each point in the map $x_i,\ i = 1, ..., n$ is as follows:

$$GD_Q(S) = \sum_{i=0}^{n} ED(S, i) \times Q\left(\frac{i}{n}\right) \tag{3}$$

where $ED(S, k) = b_k - b_{k+1}$ is a fuzzy cardinality of the E family [43] with $b_0 = 1$ and $b_{n+1} = 0$, being b_k the kth largest value of belongingness of an element to the fuzzy set S.

The generalization of the GD method for type-2 descriptions is defined as follows:

$$GD_Q(S/K) = \sum_{c \in CR(S/K)} ER(S/K, c) \times Q(c), \tag{4}$$

where K represents the linguistic qualifier of the type-2 quantified statements, ER is the fuzzy cardinality used by this method for this evaluation [43]:

$$ER(S/K, c) = \sum_{c = C(S/K, \alpha_i)} (\alpha_i - \alpha_{i+1})\ \forall c \in CR(S/K) \tag{5}$$

where

$$C(S/K, \alpha_i) = \frac{|(S \cap K)_{\alpha_i}|}{|K_{\alpha_i}|} \tag{6}$$

and CR is the set of crisp representatives of the relative cardinality defined by $\alpha - cuts$:

$$CR(S/K) = \{\frac{|(S \cap K)_{\alpha}|}{|K_{\alpha}|} \text{ with } \alpha\ \in\ M(S/K)\}$$

where $M(S/K)$ is the set of representative $\alpha - cut$ levels of S with respect to K, defined as $M(S/K) = \{\alpha \in (0, 1]\ | \exists x_i \in X \text{ with } (S \cap K)(x_i) = \alpha\ or\ K(x_i) = \alpha\}$.

Zadeh's fuzzy cardinality based method (ZS) The evaluation for type-1 quantified statements is:

$$ZS_Q(S) = \max_{k \in \{0, ..., n\}} \min\left(Z(S, k), Q\left(\frac{k}{n}\right)\right) \tag{7}$$

where

$$Z(S, k) = \begin{cases} 0 & \text{if } \nexists \alpha\ |\ |S_\alpha| = k \\ sup\{\alpha\ |\ |S_\alpha| = k\} & \text{otherwise} \end{cases} \tag{8}$$

The evaluation of type-2 descriptions is as follows:

$$ZS_Q(S/K) = \max_{c \in CR(S/K)} \min\left(ES(S/K, c), Q(c)\right) \tag{9}$$

where *ES(S/K, c)* is the relative cardinality of S with respect to K, defined as:

$$ES(S/K, c) = max\{\alpha \in M(S/K)\ |\ c = \frac{|(S \cap K)_{\alpha}|}{|K_{\alpha}|}\}\ \forall c \in CR(S/K) \tag{10}$$

From the calculated truth value by one quantification method, we generate a measure called *fulfil_score*.

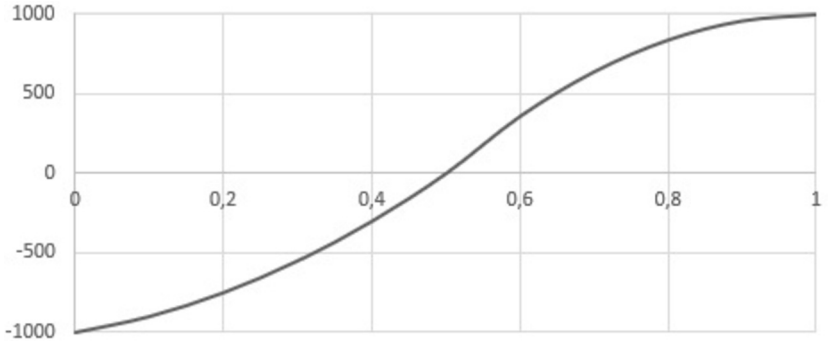


Fig. 10. *fulfil_score* representation.

In the preliminary experiments we observed that in some cases descriptions with low fulfillment degree achieved high rankings. In order to avoid this undesirable effect, we designed the *fulfil_score* variable, which essentially consists of a mapping of the fulfillment degree, aiming to boost descriptions with high fulfillment degree (from 0.8 on, in our case) and penalize descriptions with low fulfillment degree (0.5 in our case).

The mapping of the *fulfil_score* is the weight function represented in Fig. 10 of the paper which expands the [0, 1] range to a broader one [−1,000, 1,000]. This expanded range was experimentally selected, analyzing the resulting rankings, starting from [-100, 100] until reaching a range where unsuitable descriptions did not occupy the top positions of the ranking.

3.1.2. Coverage

The coverage refers to the extent of land covered by the quantifier, i.e. the number of locations that satisfy the sentence. A quantified statement is more suitable the larger the proportion of the land extent it covers as it is more representative of the general meteorological situation. Therefore, the need arises to define a heuristic that boosts the most representative descriptions and, at the same time, penalizes those descriptions that are not (e.g., "In almost all locations the sky is covered" vs "In a few locations the sky is covered").

To this aim we defined a heuristic mapping that associates a numerical weight to each quantifier, according to the proportion of terrain it covers, so that the higher the proportion covered by a quantifier, the greater the associated weight.

We design the coverage which essentially consists of a mapping of the concept of coverage, defined as a value in [0, 1] which represents the percentage of locations covered by the quantifier. Also in this case, our purpose generating this score is to extend the range of the values in order to accentuating the differences between low coverage quantifiers and high coverage quantifiers, boosting descriptions involving quantifiers that describe a larger proportion of the territory while penalizing those involving quantifiers that cover less land area.

To design this measure we used as reference the designed fuzzy set based on the percentage of locations in the map covered by each quantifier (Fig. 5). We opted for a generic definition for these weights based on the quantifiers definition. Since all quantifiers are defined as trapezoidal fuzzy sets by four points *a, b, c, d*, the numerical weights were initially defined as the average of the points *a* and *d* which define its support. From these values, we performed an empirical study adjusting them until achieve a suitable ranking of quantified statements for a specific situation. In Table 3 the assigned coverage to each quantifier is presented.

With these coverage values, we calculate the *cov_score* measure. Also, in this case, we set negative values for low coverage values under 0.5 and, from the quantifier "Some" on-wards, the assigned coverage score is higher than 0.

So, *cov_score* expands the coverage values from the range [0, 1] to [−1,000, 1,260] as represented in Fig. 11 of the paper. As in the previous case, the selection of the range was the result of an empirically study, starting from [−100, 100] until reaching a range where unsuitable descriptions did not occupy the top positions of the ranking.

3.1.3. Sentence length

The last evaluated description feature is the sentence length, preferring longer sentences in terms of the number of meteorological variables.

Table 3
Coverage values for each defined quantifier.

Quantifier	Coverage
None	0
A few	0.17
Some	0.34
About half	0.5
Many	0.67
Almost all	0.84
All	1

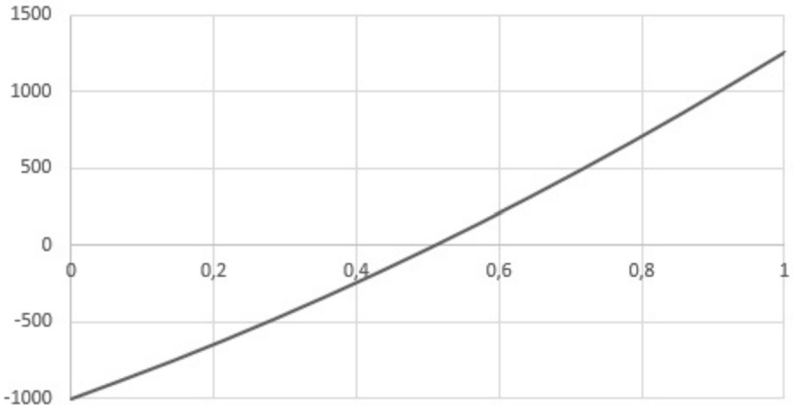


Fig. 11. *cov_score* representation.

We normalize this value in the range [0, 1] called *length_score*, which is calculated by dividing the number of variables by the maximum number of meteorological variables a description can contain (3, since we are considering sky state, wind and temperature). Therefore, a full length score (1) will be obtained when a sentence provides a description that includes all variables.

3.1.4. Score

Having those three values; coverage, sentence length and truth value; and the calculated scored from them, we calculate a score based on them as described in what follows:

$$score = fulfil\_score + cov\_score + length\_score \tag{11}$$

In the related linguistic summaries literature [17,46,47], it is shown that both the fulfillment degree and the coverage are more relevant when it comes to ordering descriptions according to how informative they are. Therefore, in our approach we have decided to assign a greater weight to both variables. However, for those cases where two descriptions have equal or similar values for fulfillment degree and coverage, the sentence length will break the tie between the scores however negligible it may seem beforehand.

3.2. Description generation approaches

In this paper, we propose two generation strategies, based on the elements that compose a description and the assessment metrics we are interested in evaluating.

3.2.1. Simulated annealing (SA)

Simulated annealing (SA) is a meta-heuristic search and optimization technique inspired by annealing in metallurgy. To avoid staying in local optima, it accepts movements to worse solutions, decreasing the probability of accepting these movements as the search advances.

This algorithm uses a variable called temperature, whose value determines to what extent candidate solutions worse than the current one can be accepted. This variable is initialized with a high value and is reduced in each iteration by a cooling mechanism which decreases progressively reducing the probability of acceptance.

In each iteration a number of candidate descriptions are generated, applying the acceptance criteria for each one to check if it replaces the current one. If the candidate description is better, it is automatically accepted, while if it is worse, there is still a probability that it will replace the current one. This probability depends on the cost difference between the current and the candidate description and on the temperature, so that the lower the cost difference and the higher the temperature, the higher the probability of acceptance.

Once an iteration is finished, the temperature is cooled down and the next one is passed on until the stop criterion is reached.

We use the classic definition of the algorithm [37]. To apply this algorithm to our solution, we design an experiment to define its design parameters with different values with the aim of finding a proper configuration.

- S_0: the initial solution is the same in both use cases. First, we aim to generate a solution based on knowledge: after selecting the most repeated value for each linguistic meteorological variable and, all the descriptions involving all quantifiers and geographical descriptors are sorted by coverage and truth value. Among them, we select the best description with a defined truth value threshold. If none of the descriptions satisfied these requirements, we generate a random initial solution.
- T_0: this parameter has the same value in both cases. To set its value high enough, we initialize it with a value inversely proportional to the maximum number of different type-1 and type-2 fuzzy quantified statements (the value of the variable $MAX\_DESCRIPTIONS$ is 63,203 for type-1 descriptions and 568,827 for type-2 sentences). The other two design attributes are the probability ϕ that a new solution $\mu\%$, worse than the initial one S_0 is accepted. These values allow initializing T_0 as indicated in Equation (12).

$$T_0 = -\mu / (\ln(\phi) * MAX\_DESCRIPTIONS) \tag{12}$$

- New candidate solution: a new candidate is generated applying some changes to the current one. Any of the elements in the sentence (quantifier, geographical descriptor, weather variables) can be changed randomly. However, since the algorithm could stay in a local maximum due to generating too many repeated candidates, we limit the number of repeated candidates in one iteration. In observation, we limit it to 20 whereas in prediction to 10 because in the second case SA tends to generate solutions that were previously generated, thus consuming too much time in this step. If this threshold is reached, a totally random candidate is generated.
- Cooling speed and strategy: we define the limit of maximum candidates generated and the maximum number of accepted candidates so when one of these values is reached, the temperature is decreased (following, in our case, a Cauchy cooling strategy). We set both limits to 350 and 35 respectively because, after experimenting with different values, we achieve an adequate cooling speed in the way worse solutions are not accepted when the search process is advanced.
- Stop condition: we design a double stop condition composed by:
  - A maximum number of iterations (1,000 in the type-1 generation and 3,000 in the type-2 generation for both observation and prediction). The execution ends when that threshold is reached.
  - A maximum number of generated solutions which were already generated and evaluated in previous iterations (1,000 in the type-1 generation and 3,000 in the type-2 generation for both observation and prediction). All generated solutions are stored in a list of candidate solutions so, when we generate a new solution we first check if it was already generated. The number of previously generated candidate solutions is stored in a counter that increases each time a new solution has been previously generated and the execution ends if this threshold is achieved. This condition is designed in order to prevent the execution from getting stuck generating already evaluated descriptions.
- Acceptance criterion: this condition is the same in both cases. SA has a default acceptance mechanism which allows the acceptance of a solution worse than the current one with a certain probability based on the cost difference between solutions (δ). Apart from this usual mechanism, in our design, solutions with a truth value higher than 0 are also accepted in case they also meet one of the following conditions:
  - Having an equal or higher truth value than the current solution
  - Having a higher coverage than the current solution

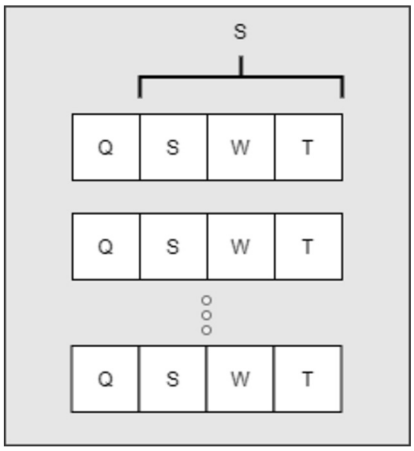


Fig. 12. Structure of a type-1 chromosome in the genetic algorithm.

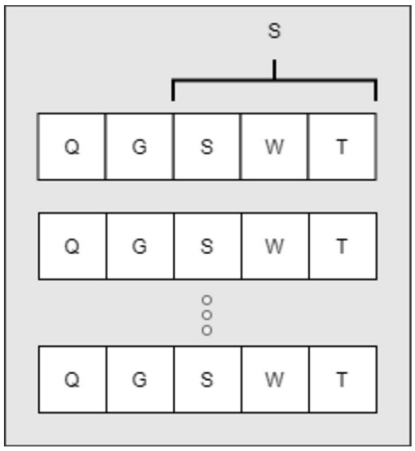


Fig. 13. Structure of a type-2 chromosome in the genetic algorithm.

3.2.2. Genetic algorithm

A genetic algorithm is a population based evolutionary meta-heuristic based on natural selection. It starts with a random generated population that undergoes modifications along the execution. At each iteration, the genetic algorithm selects the best solutions, whose features will be transmitted to their offspring in the next iteration by relying on bio-inspired operators such as crossover, selection, mutation and replacement.

To design the genetic algorithm is necessary to define the representation of the population. The resulting ranking of descriptions is represented by a chromosome composed by a set genes that encode a quantified statement.

As we mentioned in Section 1, type-1 descriptions have the structure "*Q Y* are *S*" where *Q* are the quantifiers defined in Section 2.4, *Y* is one or a combination of the linguistic variables defined in Sections 2.1, 2.2 and 2.3; and *S* a linguistic value of *Y*. Fig. 12 shows the representation of the chromosome in the type-1 quantified statements generation (SS for state sky, W for wind and T for temperature). The information within each gene is the quantifier and the values for each of the meteorological variables.

On the other side, type-2 descriptions have the structure "*Q KY* are *S*" where *K* are the geographical descriptors defined in Section 2.5. Fig. 13 shows the representation of the type-2 chromosome where each gene contain the values of the quantified, the geographical descriptor and the meteorological variables.

To avoid repetition of descriptions, each of the components of the gene have a fixed place in the gene, as represented in Figs. 12 and 13.

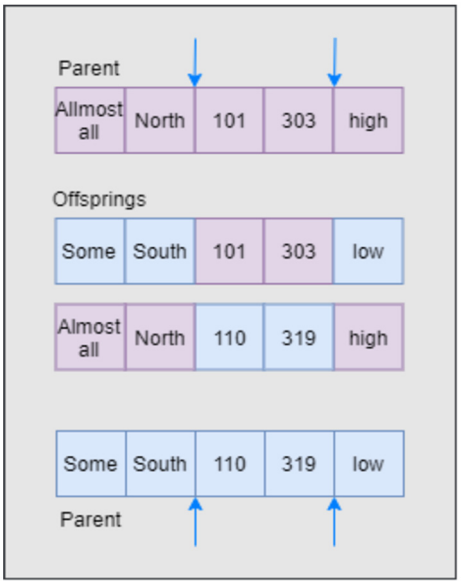


Fig. 14. Example of a crossover between two description parents to generate two new type-2 quantified statements.

For our design we follow a classical definition of a genetic algorithm, testing several options to set the most appropriate parameters values and methods:

- Initial population: we decided to generate a fixed set of random descriptions as initial population. To define the size of the initial population, we generate all possible descriptions from several meteorological situations analyzing the number of descriptions that we consider of interest in each case in the ranking. We observed in the cases analyzed that the maximum number of representative descriptions was around 200, so we defined this value as the size of the initial solution.
- Tournament phase: regarding the tournament phase, the binary option was selected since is a widely used type and tests with this option reported satisfactory results.
- Crossover: to select the crossover type, we tested with one-point and two-point crossover with percentages between 80% and 98% in pairs analyzing the quality of the descriptions set. We select the two-point crossover 90% (Fig. 14) because we obtained satisfactory results in terms of the representativeness of the descriptions.
- Mutation: we experimented with percentages between 1% and 20% also in pairs, choosing a mutation of all elements with a probability of 16%.
- Replacement: for the replacement phase we tested with percentages between 5% and 50% in 5% increments, selecting a replacement of the 30%.
- Stop condition: we define the stop condition as a fixed number of iterations. To select the number of iterations to finalize the execution, we performed tests with iterations from 500 to 4,000 in five hundred until achieving a balance between the quality of the solutions and the execution time. We set a maximum of 3,000 iterations with observation data and 500 with prediction data.

3.3. Evaluation

3.3.1. Metrics

Once we have generated the ranking of type-1 and type-2 quantified sentences for a data set, we have to evaluate its adequacy. To test these results, we set as baseline a greedy approach which generates all possible different descriptions from the combination of the elements that make them up (Q, G, S, W, T) after a full search throughout the search space, obtaining a total of 632,030 quantified statements (63,203 type-1 descriptions and 568,827 type-2 sentences).

This resulting set of all possible quantified statements, *All-Statements* from now on, is used to design the quality evaluation measures.

We analyzed and compared our two approaches in terms of quality and execution time,[1] comparing the execution time of the three approaches.

For the quality measures, for each data set, we selected the best solutions setting a threshold in a truth value of 0.8. To analyze how descriptive the set of descriptions retrieved by each proposal are, we used some evaluation metrics. Having two sets, *All-Statements* (which represents the greedy approach) and *MH* (which represents Simulated Annealing or the genetic algorithm), with cardinality $|All - Statements|$ and $|MH|$, respectively, we consider the following metrics:

- Accuracy: calculates the percentage of sentences generated in the approach that match with the set formed by all possible descriptions above the threshold.

$$accuracy = \frac{|All - Statements \cap MH|}{|All - Statements|} \tag{13}$$

- Jaccard index [48]: measures the degree of similarity between the sets (sets intersection) with regard to all different sentences generated joining them (sets union).

$$J(All - Statements, MH) = \frac{|All - Statements \cap MH|}{|All - Statements \cup MH|} \tag{14}$$

- Quality of the solutions: since meta-heuristic algorithms usually do not generate all the descriptions in *All-Statements*, the number of descriptions that have not been generated and its position in the ranking provides a quality measure for the meta-heuristics, as defined in Expression (15).

$$quality = 1 - \left(\frac{\sum_{i=0}^{n-1} score_i}{|All - Statements|} \right) \tag{15}$$

being n the size of the set $All - Statements \setminus MH$ and $score_i$ the score of each description of that set. This quality measure consist of calculating the relative cardinality (using Zadeh's cardinality operator [43]) of the number of descriptions generated by the meta-heuristics that are not included in the full set of descriptions ($All - Statements \setminus MH$). This cardinality is applied to *score*, which is the normalized score described in Equation (11). The cardinality value is subtracted from 1 in order to define a quality measure (higher values mean better behavior).

- Execution time: we collected the results in seconds. For type-2 descriptions, we designed a sequential algorithm and a parallel version in order to analyze if improvements could be achieved. We found out that most of the computational resources were consumed by loops in the sequential version, so those parts were optimized in the parallel version in order to improve its performance.
- Statistical significance: we performed statistical significance tests to check if there exist significant differences between our two approaches when generating type-1 and type-2 descriptions or not.
  To compare them, we tested both approaches with 20 different observation data sets and 12 different prediction data sets. Since in our case the data sets we used include weather data in the same geographical area, we selected cases separated temporally by more than 15 days, which we consider independent.
  The basis for this temporal separation is the average number of days that forecast services cover. For instance, MeteoGalicia only provides prediction data for 8 days. Other weather agencies show similar temporal windows, e.g., AccuWeather [49] shows five days, Windguru [50] eight days, and Meteosat [51] nine days. Thus we assumed 15 days as a safe condition for ensuring data independence, so we have 20 independent observations in the observation use case whereas we have 12 independent observations in the prediction scenario.
  We performed this statistical study comparing the accuracy obtained by each algorithm (Table 4 shows an example).
  Thus, for each case, observation and prediction, we performed statistical tests twice: for type-1 descriptions and for type-2 descriptions comparing if they achieve or not the same accuracy. We do not have previous information about the behavior of these algorithms in this case, but since both of them were used in similar use cases, we

[1] Tests executed on an Intel Core i7-6700HQ @2.60 GHz 2.59 GHz with 16 GB of RAM.

Table 4
Example of input data where the numerical data means the percentage of accuracy of our approaches when processing the available data sets.

Data set	Algorithm	
	SA	Genetic
A	40	100
B	57	100
C	100	100
D	100	100
E	75	75

assumed there does not exist differences between their performances: H_0: *There are not significant differences between the two compared algorithms used for generating type-1 and type-2 descriptions from observation and prediction data sets*.

To determine which is the most adequate statistical test for this purpose, we need first to analyze the features of our data, performing the three following tests: *i)* independence of the variables, *ii)* normal distribution and *iii)* heterocedasticity.

First, regarding the variables, we have two variables: SA accuracy and genetic accuracy. Since the execution of one algorithm does not depend on the execution of the other, it is proved that these two variables are independent.

To analyze the data distribution and test if they follow a normal distribution, in Table 5 the results of skewness, kurtosis and the Jarque-Bera test are presented. To analyze these values, we previously defined the following thresholds:

- Since a total skewed data has an skewness of 0, we consider the data has a normal skewness if this value is in the range $[-1.5, 1.5]$. In particular, if the skewness is in the range $[-0.5, 0.5]$ we consider the data is slightly skewed, and if the value is in the ranges $[-1.5, -0.5]$ and $[0.5, 1.5]$ we consider the data is moderately skewed.
- We accept as normal kurtosis a value in the range $[2, 4]$ since a normal distributed data as a kurtosis of 3.
- To interpret the results of the Jarque-Bera test, we set an $\alpha = 0.05$. Since this test follow a chi-square distribution with 2 degrees of freedom, we consider the data satisfies the null hypothesis (H_0: *the data is normally distributed*) if the statistic has a value lower than 5.991 and the p-value is higher than α.

Table 5 contains the results of the normality test. Only one data set, GA generating type-1 descriptions with observation data with the ZS quantification method, do not follow a normal distribution. The results with this data set do not satisfies the mentioned conditions for none of the metrics. All other data sets have a skewness value in the predefined range. Regarding the kurtosis measures, some data sets have values lower than 2, however, they have their values of skweness and Jarque-Bera test satisfies the conditions so we consider they follow a normal distribution. Except the mentioned data set above, all cases have an $\alpha > 0.05$ for the Jarque-Bera test.

Finally, we checked if this data satisfies the homocedasticity condition. We applied the Breusch-Pagan test which allows to detect heteroscedasticity in the data. The null hypothesis of this test is H_0*: homocedasticity exists*.

To interpret the results of this test, we set an $\alpha = 0.05$. This test follows a chi-squared distribution with 1 degree of freedom so we consider the data satisfies the H_0 if the statistic is lower than 3.841.

In Table 6 we present the results of this test applied to the different analyzed cases. In this case, we can affirm our data satisfies the homocedasticity condition since the statistic value is lower than 3.841 and the p-value is greater than 0.05 in all cases.

To decide the most suitable test for our data, we have to consider the previously described features:

- Independent observations
- Two groups are compared
- Independent variables
- Distribution is normal in all data sets except one
- Data satisfies the homocedasticity condition

Considering the conditions described above and their results, as suggested by the reviewer, the most adequate test for the data sets which satisfy all previous conditions is a unpaired two-sample t-test setting an $\alpha = 0.05$ [52]. For

Table 5
Results of skewness, kurtosis and Jarque-Bera test about the compared data.

				Skewness	Kurtosis	Jarque-Bera test		Normal
						statistic	p-value	
Zadeh	Observation	Type-1	SA	−0.74	2.32	2.20	0.33	yes
			GA	−1.33	3.07	5.90	0.052	yes
		Type-2	SA	−0.13	2.18	0.62	0.73	yes
			GA	−1.10	2.69	4.14	0.13	yes
	Prediction	Type-1	SA	−1.01	3.34	0.81	0.67	yes
			GA	−1.07	3.17	2.09	0.35	yes
		Type-2	SA	−0.03	1.41	1.26	0.53	yes
			GA	−0.82	2.27	1.60	0.45	yes
GD	Observation	Type-1	SA	−0.26	1.56	1.86	0.40	yes
			GA	−1.36	3.24	5.92	0.0519	yes
		Type-2	SA	−0.98	2.51	2.74	0.25	yes
			GA	1.30	−0.55	3.24	0.20	yes
	Prediction	Type-1	SA	−0.55	2.49	0.73	0.69	yes
			GA	−1.17	2.60	2.82	0.24	yes
		Type-2	SA	−0.06	1.64	1.01	0.60	yes
			GA	−1.19	3.47	3.17	0.21	yes
ZS	Observation	Type-1	SA	−0.05	1.29	2.33	0.31	yes
			GA	−1.70	4.34	10.57	0.0005	no
		Type-2	SA	−0.53	1.81	2.01	0.37	yes
			GA	−0.06	2.17	0.55	0.76	yes
	Prediction	Type-1	SA	−0.28	1.96	0.69	0.71	yes
			GA	−0.56	2.14	1.01	0.60	yes
		Type-2	SA	−0.30	1.87	0.82	0.66	yes
			GA	−0.16	2.10	0.46	0.80	yes

Table 6
Results of the Breusch-Pagan test for heterocedasticity.

			Breusch-Pagan test		Homocedasticity
			statistic	p-value	
Zadeh	Observation	Type-1	2.08	0.15	yes
		Type-2	0.57	0.45	yes
	Prediction	Type-1	0.14	0.71	yes
		Type-2	1.42	0.23	yes
GD	Observation	Type-1	3.76	0.052	yes
		Type-2	1.67	0.20	yes
	Prediction	Type-1	0.11	0.74	yes
		Type-2	1.01	0.31	yes
ZS	Observation	Type-1	0.05	0.84	yes
		Type-2	0.01	0.92	yes
	Prediction	Type-1	0.06	0.81	yes
		Type-2	0.21	0.64	yes

that data set which does not follow a normal distribution, the most appropriate test is the Mann-Whitney-Wilcoxon Test [53,54] also with an $\alpha = 0.05$.

3.3.2. Baseline approach

In order to have a baseline to compare the proposed meta-heuristics with, we designed and implemented a simple strategy (random search). Confronting the meta-heuristics with the baseline will provide an experimental basis for showing that meta-heuristic algorithms are more suitable for this problem than simpler approaches.

Table 7
Summary of the results of the type-1 random baseline.

#attempts	Observation				Prediction			
	Accuracy (%)		Time (s)		Accuracy (%)		Time (s)	
	Average	SD	Average	SD	Average	SD	Average	SD
30,000	36.64	8.59	28.64	5.99	32.68	7.60	282.08	76.31
50,000	38.02	4.76	45.34	1.75	41.10	4.52	490.58	132.45
70,000	45.40	4.65	72.39	15.86	49.44	5.05	663.14	104.68
90,000	50.90	4.13	95.44	17.81	55.30	3.93	876.95	283.85

Table 8
Summary of the results of the type-2 random baseline.

#attempts	Observation				Prediction			
	Accuracy (%)		Time (s)		Accuracy (%)		Time (s)	
	Average	SD	Average	SD	Average	SD	Average	SD
100,000	39.08	12.49	134.89	51.84	44.51	8.56	1,124.97	385.14
300,000	65.88	10.01	627.89	42.39	79.94	12.17	2,705.69	212.22
500,000	84.73	9.26	897.53	38.05	82.89	8.38	5,480.28	1,363
700,000	91.07	9.11	1,056.75	27.11	84.34	10.31	8,255.75	1,183.52

Taking into account that in this case we can generate 63,203 different type-1 statements, we executed four random baselines with 30,000, 50,000, 70,000 and 90,000 type-1 generated descriptions. In the case of type-2 statements, we can generate 568,203 different statements. Then, we also performed four baselines with respectively 100,000, 300,000, 500,000, and 700,000 generated descriptions.

Table 7 shows the summary of accuracy and execution time of the baseline for type-1, both observation and prediction. This baseline obtained a low accuracy in both scenarios (lower than a 60% on average regardless of the number of generated descriptions).

On the other side, Table 8 shows the summary of accuracy and execution time of the baseline for type-2, also for the observation and prediction scenarios. For type-2, this baseline obtained a high average accuracy generating 700,000 attempts, with an accuracy of 91.07% for observation and 84.34% for prediction, on average. Nevertheless, its execution time is very large, with more than 17 minutes in observation and more than 2 hours in prediction generating 700,000 attempts.

4. Experimentation results

In this section, we show the experimentation we performed in order to evaluate the obtained results with our approaches.

For the three quantitative measures of accuracy, Jaccard index and time, we calculate their average and their standard deviation (SD).

4.1. Observation

4.1.1. Type-1 descriptions

In Table 9 the summary of the measures is presented for the type-1 test with observation data where we compare both approaches regarding four measures: the percentage of sentences generated by the approach with match with *All-Statements*, the percentage of matches between the descriptions generated by the approach and *All-Statements* with regard to those sets union, the quality of the solution identifying the position in the *All-Statements* set of the non-generated descriptions by the approach; and the time in seconds.

Regarding accuracy, the GA achieved better results than its SA counterpart regardless of the quantification method used in the evaluation. The standard deviation was higher in GA with Zadeh's method, whereas in all other cases SA

Table 9
Metrics results from type-1 observation test.

		Accuracy (%)		Jaccard index (%)		Solution quality	Time (s)	
		Average	SD	Average	SD		Average	SD
Zadeh	SA	84.56	2.85	80.56	2.83	0.55	11.43	2.40
	GA	96.11	7.11	96.11	7.11	0.65	1.00	0.00
GD	SA	63.38	30.73	63.38	30.73	0.45	12.23	2.10
	GA	91.05	16.18	91.05	16.18	0.64	1.11	0.00
ZS	SA	67.12	32.00	67.12	32.03	0.47	11.55	2.00
	GA	89.41	22.29	73.43	17.43	0.52	1.00	0.10

Table 10
Degree of fulfillment obtained by the three quantification methods.

	Zadeh	GD	ZS
In almost all locations the sky is covered	1.00	1.00	1.00
In about half locations the sky is covered and the temperature is high	1.00	0.91	0.97
In a few locations the sky is rainy and the temperature is normal	1.00	1.00	1.00
In all locations in the Southeast the sky is covered	1.00	0.99	1.00
In almost all locations in the West the sky is covered and the temperature is high	1.00	0.90	1.00
In all locations in the Northeast the sky is covered	1.00	1.00	1.00
In many locations in the Center the sky is covered and the temperature is high	0.94	0.58	0.58
In almost all locations in the East the temperature is high	0.67	0.44	0.47
In almost all locations in the East the sky is covered	1.00	0.99	1.00
In some locations in the West the wind has North direction and moderate speed	0.94	0.67	0.87

has higher values. It is also noticeable the standard deviation increased considerably with the GD and ZS methods in both approaches.

Regarding the Jaccard index, GA got better percentages than SA with all quantification methods. For instance, with the Zadeh's quantification method, SA got an 80.56%, which means about a 20% of generated sentences it generated were different from the expected whereas GA got a 96.11%. Both approaches had a low standard deviation, which means most of the results are concentrated on average.

Besides, the GA also achieved better results analyzing the unmatched positions, which means it avoids generating descriptions in lower positions in the ranking than the SA.

The differences in the metrics between the three quantification models with each algorithm cannot be justified solely by the selection of the method but also to the non-deterministic nature of the algorithms. In Table 10 a list of descriptions with the truth value calculated by each quantification method proves they have a similar behavior evaluating the sentences. Besides, in [45] we proved there is no significant differences between this set of quantification models when evaluating type-1 descriptions with data from 15 different data sets.

With respect to execution duration, the performance of both approaches was not conditioned by the quantification model in the evaluation since the results are very close. For example, with the Zadeh's method SA consumed 11.43 seconds on average while the genetic algorithm 1 second, so the genetic approach achieved the best result not only in terms of accuracy and Jaccard index but also in duration.

Results of the statistical tests are summarized in Table 12. As mentioned above, to interpret the test results, we set an $\alpha = 0.05$. In this case we tested the data with a t-test for all cases except for type-1 with ZS because it did not follow a normal distribution so a Wilcoxon-Mann-Whitney test was performed. In all cases, the p-value is lower than our α so we can conclude there exist differences between our two approaches when generating type-1 descriptions.

4.1.2. Type-2 descriptions

The results of the type-2 tests are presented in Table 11. In this case SA obtained an accuracy lower than the GA with the three quantification models. It is remarkable the accuracy is very similar in both approaches when the ZS method is used in the evaluation. However, as described above, this percentages are not only influenced by the quantification method but also by the algorithms features.

Table 11
Metrics results from type-2 observation test.

			Accuracy (%)		Jaccard index (%)		Solution quality	Time (s)	
			Average	SD	Average	SD		Average	SD
Zadeh	SA	sequential	72.81	3.29	72.81	3.29	0.54	39.80	3.05
		parallel						26.63	2.32
	GA	sequential	95.02	3.71	85.21	3.76	0.77	4.75	0.89
		parallel						4.71	0.72
GD	SA	sequential	82.37	15.18	82.37	15.18	0.62	39.00	2.73
		parallel						27.03	2.23
	GA	sequential	96.32	4.96	87.81	8.52	0.89	5.82	1.20
		parallel						4.83	0.68
ZS	SA	sequential	80.55	9.92	70.88	5.65	0.59	37.80	5.56
		parallel						28.20	3.14
	GA	sequential	80.63	7.37	80.55	9.92	0.60	4.95	1.73
		parallel						4.81	1.22

Table 12
Summary of the results of the statistical tests applied to the observation case.

		test	statistic	df	p-value
Zadeh	Type-1	t-test	−2.5106	38	0.0160
	Type-2	t-test	−11.9910	38	$1.742e^{-14}$
GD	Type-1	t-test	−3.3659	36	0.0018
	Type-2	t-test	−3.8068	36	0.0005
ZS	Type-1	Wilcoxon-Mann-Whitney	86	-	0.002964
	Type-2	t-test	−0.027581	33	0.9781

Analyzing the Jaccard index, it is noticeable that even when the performance seems very similar, the GA generated better solutions whereas SA generated a higher percentage of sentences different from the expected. For instance, with the Zadeh's method SA got a 72.81% whereas genetic got a 85.21%. On the basis that this measure calculates the proportion of the matches, the union of the SA set and the best set contains about a 30% of not matched sentences whereas in the genetic algorithm case is about a half of SA. Both approaches had a low standard deviation, so the results distribution is very close to the average.

If we refer to the score obtained in the solution quality metric, on top of getting worse accuracy and Jaccard index scores, SA also achieved a lower score in this metric. Therefore, its quality can be considered worse in general.

Both SA and GA obtained worse results than when generating type-1 sentences. This can be caused by the larger amount of type-2 sentences that are generated. In type-2 it is easier to remain within a local maximum, generating many undesirable sentences and increasing the execution time as a result.

Comparing the execution duration, it is no influenced by the selection of the quantification model. GA had the best results, consuming, on average, less than 30 seconds than SA.

However, in this case, the execution duration was not excessive, as the observation data set is smaller. The parallel version reduced the execution time considerably in all combinations and SA but even so, the genetic algorithm achieved the best result.

In this case, all type-2 data satisfied the conditions to apply a t-test. As in the type-1 case, the results of the t-test with 38 degrees of freedom are presented in Table 12 where also significant differences between the two compared algorithms are detected, since the p-value is lower than α.

4.2. Prediction

4.2.1. Type-1 descriptions

Both algorithms achieved similar results with the Zadeh's quantification method, with an accuracy average of 73.26% with SA and a 74.58% with GA. With GD, SA experienced a drop in accuracy of up to 59.25% whereas GA

Table 13
Descriptions for Fig. 1 situation.

Description	Score
In almost all locations the sky's state is covered	1,286.58
In a few locations in the South-east the sky's state is rain	1,572.26
In many locations the temperature is normal	1,572.23
In about half locations in the North-east the temperature is low	1,572.23
In some locations the wind has North direction and low speed	1,572.23
In about half locations in the South-east the wind has North-west direction and low speed	1,318.05
In about half locations in the North-east the wind has North-west direction and low speed	1,220.39
In about half locations in the South-east the wind has North direction and low speed	1,000.64
In about half locations in the South the wind has North direction and low speed	1,000.61
In some locations in the West the wind has North direction and low speed	984.49

Table 14
Metrics results from type-1 prediction test.

		Accuracy (%)		Jaccard index (%)		Solution quality	Time (s)	
		Average	SD	Average	SD		Average	SD
Zadeh	SA	73.26	22.97	73.26	22.97	0.60	126.71	2.43
	GA	74.58	29.15	74.58	29.15	0.66	36.34	3.35
GD	SA	59.25	27.89	59.25	27.89	0.56	182.32	5.57
	GA	78.41	35.94	78.41	35.94	0.67	42.36	3.97
ZS	SA	72.08	18.55	70.92	17.61	0.66	194.10	4.55
	GA	85.50	14.68	82.58	15.34	0.68	50.07	4.06

had an accuracy of 78.41% on average, increasing the difference between the two approaches in terms of accuracy. With ZS, SA achieved a 72.08% whereas GA A 85.5%.

In this case both approaches obtained the same percentage for the Jaccard index and for the accuracy except for ZS. This can be possible if the set of results from each approach is contained in the *All-Statements* set, but at the same time it does not contain all the most relevant sentences in the highest tier of the ranking.

In terms of their standard deviations, both approaches obtained an average of 22.97% for SA and 29.15% for GA using Zadeh's method whereas it decreased with ZS method where SA obtained a 18.55% and GA 14.68%. Unlike in the observation case, data are not as concentrated on average.

In the solution quality metric, our approaches also achieved similar results, but the genetic algorithm result is slightly higher, so its quality solution is better than the SA . (See Table 14.)

Regarding the results of the significance statistical test, with Zadeh's quantification method a t-test with 22 degrees of freedom was performed for type-1 descriptions where no significant differences were detected by the statistical test between the two meta-heuristic algorithms.

With GD, the statistical comparison between our two approaches with type-1 descriptions was performed with a t-test with 22 degrees of freedom that did not find significant differences.

Also in the ZS case, no differences between our two approaches were detected by the statistical test.

The results are presented in Table 16.

Regarding the execution duration, in this case our meta-heuristic approaches need more time to converge than in the observation case due to the amount of data they have to deal with. SA needs about 2 minutes, whereas the genetic algorithm needs about 30 seconds with the Zadeh's quantification model. However, with the other methods the necessary time to complete the execution increased by more than 1 minute for SA whereas only a few seconds for GA.

This means the prediction data type not only means a challenge in terms of time to our approaches but also in terms of accuracy. This is due to type-1 descriptions show a general idea of the status, but the covered area is larger than in the observation case. Therefore, low coverage quantifiers have a low score because of our penalization and high coverage values have a low truth value because of the diversity of the area. This means that the best descriptions subset is too small, so generating it is a difficult task for our algorithms.

Table 15
Type-2 prediction statistic tests results.

			Accuracy (%)		Jaccard index (%)		Solution quality	Time (s)	
			Average	SD	Average	SD		Average	SD
Zadeh	SA	sequential	72.35	13.04	72.35	13.04	0.52	130.61	4.32
		parallel						43.34	3.56
	GA	sequential	75.02	8.71	75.21	8.76	0.56	45.33	3.24
		parallel						13.0	2.42
GD	SA	sequential	70.81	12.59	70.70	12.60	0.56	719.63	2.32
		parallel						212.88	2.04
	GA	sequential	83.35	14.46	85.51	22.04	0.67	362.43	2.17
		parallel						171.96	1.88
ZS	SA	sequential	77.73	6.78	71.90	12.55	0.42	790.23	7.78
		parallel						322.33	5.88
	GA	sequential	87.81	5.92	85.97	6.98	0.49	481.44	5.64
		parallel						227.60	3.57

Table 16
Summary of the results of the t-test applied to the prediction data.

		test	statistic	df	p-value
Zadeh	Type-1	t-test	−0.3311	22	0.7437
	Type-2	t-test	−1.0244	22	0.3168
GD	Type-1	t-test	−1.4596	22	0.1585
	Type-2	t-test	−2.3585	24	0.0268
ZS	Type-1	t-test	−1.9647	22	0.06221
	Type-2	t-test	−3.8794	22	0.0008

4.2.2. Type-2 descriptions

These results, presented in Table 15, show that GA achieved better results than SA.

Regarding the quality of the solutions, both approaches achieved similar results although the genetic algorithm score is also in this case, better.

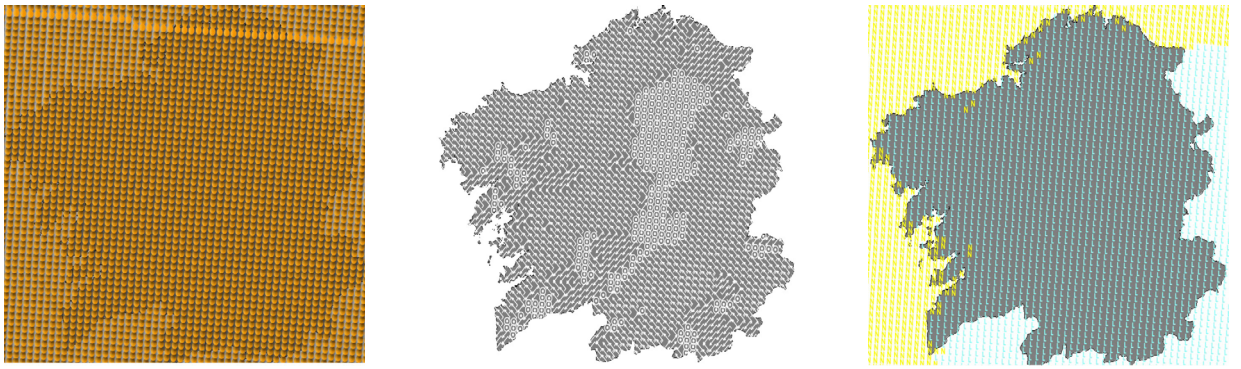
With regard to the t-test (Table 16), only with the Zadeh's quantification method no differences were detected between our two approaches. Nevertheless, both with GD and ZS the t-test found significant differences between the two approaches generating type-2 descriptions.

Regarding the execution time, SA needed more time than the genetic algorithm both in their sequential and in their parallel version for the three quantification methods. Besides, the performance of the approaches seems to be conditioned by the quantification model since the execution time increased with GD and ZS.

After performing all statistical tests, significant differences were detected in the observation case, both for type-1 and type-2 descriptions except for type-2 statements with the ZS quantification method. On the other hand, in the prediction scenario no differences were detected between our approaches with type-1 sentences, regardless the quantification method whereas with type-2 descriptions no differences were detected only with the Zadeh's quantification method but significant differences were detected with GD and ZS.

Analyzing the behavior of the algorithms, we can conclude the genetic algorithm is more stable in terms of performance than the SA regardless of the number of described locations. SA tends to generate descriptions composed by quantifiers with low coverage (specially with the "few" quantifier) which usually have high degrees of fulfillment, since when describing a small region is more probable the locations satisfy the situation. However, as we described above, we are interested in those sentences which cover more geographical extension. Therefore, although SA generates descriptions which are representative of small extents of land, those sentences should not occupy high positions in the ranking.

In those cases where the algorithms have an inaccurate behavior, analyzing the set of generated descriptions we realized the set of generated descriptions does not provide an adequate description of the situation since the truth



(a) Sky state (b) Wind (c) Temperatures

Fig. 15. Prediction state.

Table 17
Descriptions for Fig. 15 situation.

Description	Score
In all locations the sky's state is clear	2,328.45
In many locations the temperature is normal	2,100.99
In many locations the temperature is low	2,000.56
In about half locations the wind has South-east direction and low speed	1,937.44
In a few locations in the North-east the wind is calm	1,937.44
In a few locations in the Center the wind is calm	1,710.28
In a few locations in the South the wind has South-west direction and low speed	1,599.31
In a few locations South the wind is calm	1,308.37
In a few locations in the North-west the wind is calm	1,308.37

degree of the descriptions is lower than 0.2 being 0 in most cases which means a sentence does not adequately represent the situation described by the data. Besides, the descriptions are composed by the "All" quantifier and the three linguistic variables: sky, wind and temperature. So, to achieve a faithful description, 100% of the locations must meet a complex condition, which in turn is made up of these three clauses.

This behavior seems to be more noticeable where the number of locations to describe is low since we clearly detected this trend analyzing the SA output in the observation case. In the prediction case we also detected this phenomenon but in the intermediate ranking positions which did not have such an impact on the algorithm performance.

In Fig. 1 some maps with specific situations from observation are presented, whereas in Fig. 15 a prediction state centered in Galicia is shown, although the covered area is larger for better visualization purposes. Even in observation, where the amount of locations is considerably lower than in prediction, it is rather difficult to analyze such maps and understand what the underlying weather situation is. This problem is further accentuated in forecasting, where the density of icons impedes distinguishing them.

In Table 13 some of the descriptions we obtained for an observation data set are shown and in Table 17 several descriptions for a forecast use case are listed. Although not actual texts, these sentences can be very useful by helping reveal situations that are difficult to identify merely visually (e.g., the cases we have referred to in Figs. 1 and 15, which are unintelligible for humans).

4.3. Baseline comparison

We performed statistical significance tests with an $\alpha = 0.05$ to compare the accuracy of this baseline with our two meta-heuristic approaches. Table 18 shows the results for the type-1 scenario, both for observation and prediction. Results show significant differences between the two approaches in terms of accuracy in all cases so these results justify the proposal of this type of algorithms for this use case. The baseline obtained an average accuracy lower than

Table 18
Summary of the results of the statistical tests between the baseline and the meta-heuristic approaches for the type-1 case.

		Observation	Prediction
		p-value	p-value
30,000	SA	$7.88e^{-13}$	$5.65e^{-05}$
	GA	$< 2.2e^{-16}$	$< 2.2e^{-16}$
50,000	SA	$1.94e^{-12}$	0.001
	GA	$< 2.2e^{-16}$	$< 2.2e^{-16}$
70,000	SA	$2.24e^{-10}$	0.03
	GA	$< 2.2e^{-16}$	$< 2.2e^{-16}$
90,000	SA	$8.68e^{-09}$	0.017
	GA	$< 2.2e^{-16}$	$< 2.2e^{-16}$

Table 19
Summary of the results of the statistical tests between the baseline and the meta-heuristic approaches for the type-2 case.

		Observation	Prediction
		p-value	p-value
100,000	SA	$1.4e^{-08}$	0.002
	GA	$< 2.2e^{-16}$	0.01
300,000	SA	0.60	0.001
	GA	$3.91e^{-08}$	0.41
500,000	SA	$1.05e^{-07}$	$9.28e^{-05}$
	GA	$1.47e^{-12}$	0.25
700,000	SA	$2.51e^{-10}$	$6.30e^{-05}$
	GA	$6.94e^{-08}$	0.19

60% both for observation and prediction whilst our meta-heuristic approaches achieved an average accuracy higher than 70%.

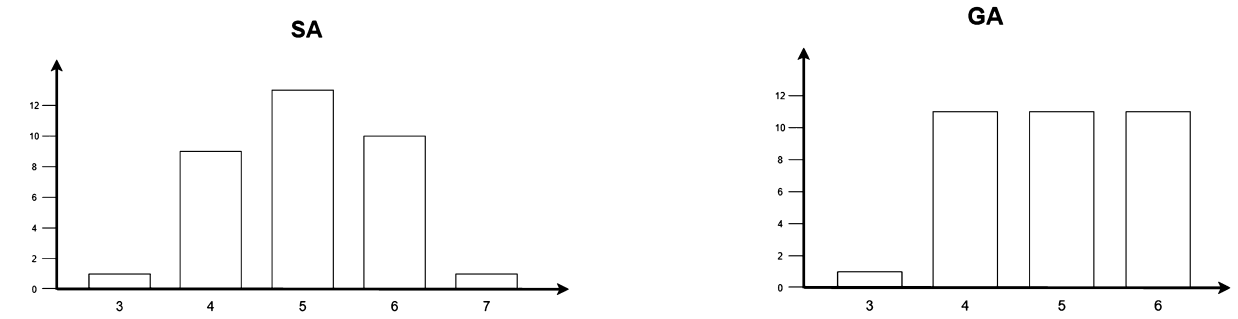
On the other side, Table 19 shows the results of the statistical significance tests for type-2 also for both observation and prediction. In this case, no significant differences are detected between GA and the baseline in several cases in terms of accuracy since the baseline. However, if we analyze the execution time, the choice of GA over the random baseline is justified since its execution time is too long with more than 17 minutes in observation and more than 2 hours in prediction generating 700,000 attempts whilst GA needs less than 5 seconds in the observation case and less than 44 seconds in the prediction scenario.

4.4. Expert validation

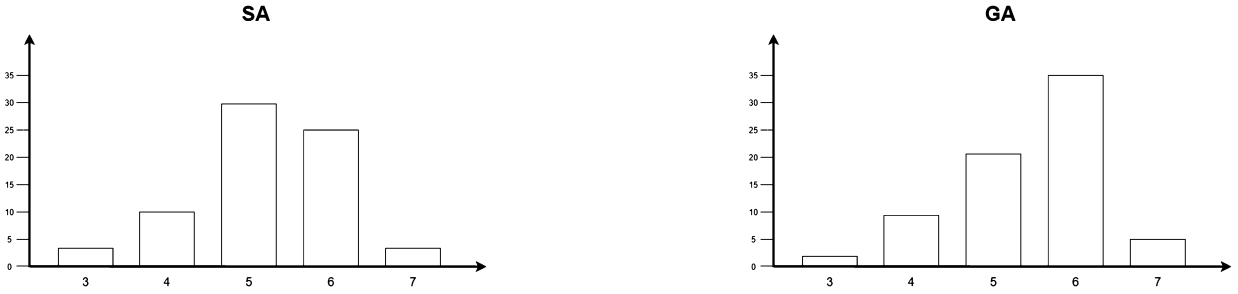
In the performed study above, we evaluated the quality of the results of the meta-heuristic algorithms in terms of the descriptions position in the ranking. It was a metric-based assessment where the adequacy of these descriptions to the domain problem was not considered.

Therefore, we also asked an expert meteorologist from the Galician Meteorological Agency (MeteoGalicia [35]) to assess the quality of the descriptions in this domain. We forwarded him a questionnaire with a variety of meteorological situations, asking him to evaluate the suitability of the type-1 and type-2 statements to describe the different situations.

The expert rated a set of 30 different meteorological cases composed by a map described by eight type-1 and type-2 quantified statements which potentially described the case. A fully blind assessment was performed, since no details were provided about what algorithm generated the descriptions and how these were generated. In fact, the expert was not aware that the descriptions were automatically generated.



(a) Type-1 scores distribution



(b) Type-2 scores distribution

Fig. 16. Distribution of the expert scores separated by algorithm and type of quantified statement.

For each case, the expert rated suitability of the descriptions using a 7-point Likert scale in the range [1, 7] where 1 means "the description is absolutely unsuitable" and 7 "the description is absolutely suitable". According to the literature, using seven values in the assessment scale maximizes reliability, validity and discriminative power [55].

The scores, whose distribution is represented in Fig. 16, show that no description was rated with the values 1 or 2 of the scale. The median of the scores of the entire set of quantified statements is 5, whereas the interquartile range is 1 (average of 5.22 and standard deviation of 0.91). Furthermore, it was also never the case that all statements of a case were rated with a score of 4 or less. Therefore, these results prove this sentence structures are suitable to describe meteorological situations according to the expert.

Regarding type-1 statements, 32.35% of the descriptions were rated with an score of 4 or lower whereas 17.61% of the type-2 statements were rated with these scores. Besides, also 32.35% of the type-1 descriptions were rated with an score of 6 or more while 47.89% of the type-2 sentences obtained these scores. It is also noticeable the expert did not rate the entire set of descriptions associated to an specific case with 4 or less.

Analyzing these results, we can conclude type-2 descriptions are more suitable than type-1 statements to describe a meteorological situation since the type-1 have a median of 5 with an interquartile range of 2 (score average of 4.99 and standard deviation of 0.89) whereas the median of type-2 is also 5 with an interquartile range of 1 (average of 5.33 and standard deviation of 0.90).

Analyzing the data by approach, the median of the SA score is 5 with an interquartile range of 2 (average of 5.03 and standard deviation of 0.9) in the type-1 scenario and it has a median of 5 and an interquartile range of 1 (average of 5.21 and standard deviation of 0.90) in the type-2 case. On the other hand, GA have a median of 5 with an interquartile range of 1 (average of 4.94 and standard deviation of 0.89) in type-1 and a median of 6 with an interquartile range of 1 in type-2 (average of 5.45 and standard deviation of 0.90). These values indicate the descriptions quality is similar for both approaches.

Type-1 score distributions are very similar for our two approaches. Only 1.47% of the statements for the two approaches were rated with 3. It is noticeable 1.47% of the SA were rated with 7 whereas none of the descriptions

generated by GA reached that score. The remaining descriptions for GA were uniformly distributed in the range [4, 6] whereas the SA are more concentrated in 5.

Type-2 scores are also similar between the two meta-heuristic algorithms. In this case, 2.11% of the statements were rated with 3 for SA whereas 1.41% of the GA descriptions obtained this score. Besides, also 2.11% of the SA sentences were rated with 7 while 3.52% of the GA statements achieved this score. The remaining SA statements are mainly rated with 5, with a 20.42% whereas the GA descriptions were mostly rated as 6, with a 24.65%.

The results of the expert evaluation do not reveal a difference between our two approaches in terms of quality in the type-1 scenario since the distribution of the scores is very similar.

Nevertheless, in the type-2 case, although there rating distribution is also rather similar, the GA percentage of descriptions rated with 6 and 7 is higher than for SA, so we can conclude the set of GA type-2 descriptions is more suitable to describe a meteorological situation.

5. Conclusions

In this work, we assessed the quality of the linguistic descriptions of data automatically generated for fuzzy quantified type-1 and type-2 protoforms using two meta-heuristic algorithms for three fuzzy quantification models. On an experimental setting made up by real meteorological observation and prediction data corresponding to several different situations for the sky state, wind and temperature variables, we evaluated the statistical significance of the differences between the adequacy of the linguistic descriptions generated by the meta-heuristics. Both metric-based and human expert evaluation were conducted.

The meta-heuristics considered were *i)* a solution based on the Simulated Annealing, which was not previously applied to linguistic descriptions generation and could be used as a lightweight alternative; *ii)* a genetic algorithm, already used in the literature for linguistic descriptions generation in other realms. For them, we generated descriptions for meteorological situations from different sources, a real-time observation service and a forecasting one. These sources allowed us to test both meta-heuristics and models in different ways: with live observation data we tested the performance of our algorithms in terms of execution time, since in this case this was the critical dimension. With prediction data we tested their behavior with big amounts of data. The fuzzy quantification models considered were Zadeh's scalar quantification model [42], Delgado's GD quantification method [43] and Zadeh's fuzzy cardinality method [44].

Empirical validation was conducted, indicating that, overall, the Genetic Algorithm performs better than Simulated Annealing in the involved quality metrics and in terms of execution time in the performed experiments obtaining the most representative set of type-1 and type-2 fuzzy quantified descriptions.

The statistical significance of this out-performance depends on the following design factors: the type of meteorological data, the quantified statements structure and the quantification model selected.

First of all, regarding the type of meteorological data, results show that for smaller data set sizes GA performs significantly better than SA in terms of quality of the descriptions, thus being GA clearly the best choice. For larger data sizes the differences in performance are not significant is many cases. Therefore, the search algorithm selection should be guided by other metrics, such as execution time, where GA is also better than SA.

Regarding the structure of the fuzzy quantified descriptions seems, for the type-1 scenario, it does not to have an impact in the performance of the meta-heuristic algorithms. Therefore, the meta-heuristic selection should be guided by other dimensions such as data size. For the type-2 case, we found significant differences between GA and SA in many cases with GA performing significantly better than SA. These results indicate that in the generation of type-2 fuzzy quantified sentences, the fuzzy quantification method conditions the performance of meta-heuristic algorithms.

In the comparison of the quantification methods, results show that in type-1 the selection of this method does not seem to have an impact in the performance of the meta-heuristic approaches since, as can be seen in Table 20, GA performs significantly better than SA with observation data regardless of the quantification method and no significant differences between the two algorithms were detected for either algorithm in prediction data. Nevertheless, in the type-2 scenario the selection of the quantification method has an impact on the performance of the meta-heuristic algorithms. With observation data significant differences are detected between them for the Zadeh and GD methods while for prediction differences are detected with the GD and ZS methods. Thus, selection of the fuzzy quantification method should be made considering this.

Table 20
Summary of the significance difference detection between the GA and SA meta-heuristic where ✓ means "significant difference between GA and SA" and ✗ means "no significant differences detected between GA and SA". Differences are always in favor of GA.

		Observation	Prediction
Type-1	Zadeh	✓	✗
	GD	✓	✗
	ZS	✓	✗
Type-2	Zadeh	✓	✗
	GD	✓	✓
	ZS	✗	✓

Also human expert assessment on the adequacy of the descriptions was conducted, showing that both meta-heuristics behave similarly for type-1 descriptions. For type-2 linguistic descriptions, human expert evaluation indicates that the adequacy of the descriptions generated by the Genetic Algorithm are more suitable than the ones generated by Simulated Annealing.

As future work we plan to consider other protoforms as well as to further process the obtained descriptions, building the realization stages of a Data-To-Text system. The final aim is to create a fluent text which relates different descriptions and combines them based on rhetorical relations such as contrast [56], e.g., as in "In almost all locations the sky is clear except in a few locations in the North, where the sky is covered".

Declaration of competing interest

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

Acknowledgements

This research was funded by the Spanish Ministry for Science, Innovation and Universities (grants TIN2017-84796-C2-1-R, PID2020-112623GB-I00 and PDC2021-121072-C21) and the Galician Ministry of Education, University and Professional Training (grants ED431C2018/29 and ED431G2019/04). All grants were co-funded by the European Regional Development Fund (ERDF/FEDER program).

References

[1] E. Reiter, An architecture for data-to-text systems, in: Proceedings of the 11th European Workshop on Natural Language Generation, Association for Computational Linguistics, 2007, pp. 97–104.
[2] A. Gatt, E. Krahmer, Survey of the state of the art in natural language generation: core tasks, applications and evaluation, J. Artif. Intell. Res. 61 (2018) 65–170, https://doi.org/10.1613/jair.5477.
[3] F. Portet, E. Reiter, A. Gatt, J. Hunter, S. Sripada, Y. Freer, C. Sykes, Automatic generation of textual summaries from neonatal intensive care data, Artif. Intell. 173 (7–8) (2009) 789–816, https://doi.org/10.1016/j.artint.2008.12.002.
[4] E. Goldberg, N. Driedger, R.I. Kittredge, Using natural-language processing to produce weather forecasts, IEEE Expert 9 (2) (1994) 45–53, https://doi.org/10.1109/64.294135.
[5] S. Sripada, E. Reiter, I. Davy, SumTime-Mousam: configurable marine weather forecast generator, Expert Update 6 (3) (2003) 4–10.
[6] E. Reiter, S. Sripada, J. Hunter, J. Yu, I. Davy, Choosing words in computer-generated weather forecasts, Artif. Intell. 167 (1–2) (2005) 137–169, https://doi.org/10.1016/j.artint.2005.06.006.
[7] L. Wanner, B. Bohnet, N. Bouayad-Agha, F. Lareau, D. Nicklaß Marquis, Generation of user-tailored multilingual air quality bulletins, Appl. Artif. Intell. 24 (10) (2010) 914–952, https://doi.org/10.1080/08839514.2010.529258.
[8] L.S. Riza, B. Putra, Y. Wihardi, B. Paramita, Data to text for generating information of weather and air quality in the R programming language, J. Eng. Sci. Technol. 14 (1) (2019) 498–508.
[9] L.A. Zadeh, J. Kacprzyk, Fuzzy Logic for the Management of Uncertainty, John Wiley & Sons, Inc., 1992.
[10] L.A. Zadeh, Fuzzy logic = computing with words, IEEE Trans. Fuzzy Syst. 4 (2) (1996) 103–111, https://doi.org/10.1109/91.493904.
[11] L.A. Zadeh, J. Kacprzyk, Computing with Words in Information/Intelligent Systems 1: Foundations, Physica-Verlag, Heidelberg, 1999.

[12] L.A. Zadeh, 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, 2000, pp. 3–40.
[13] L.A. Zadeh, A new direction in AI: toward a computational theory of perceptions, AI Mag. 22 (1) (2001) 73, https://doi.org/10.1609/aimag.v22i1.1545.
[14] G. Triviño, M. Sugeno, Towards linguistic descriptions of phenomena, Int. J. Approx. Reason. 54 (1) (2013) 22–34, https://doi.org/10.1016/j.ijar.2012.07.004.
[15] J. Kacprzyk, R.R. Yager, Linguistic summaries of data using fuzzy logic, Int. J. Gen. Syst. 30 (2) (2001) 133–154.
[16] L.A. Zadeh, A prototype-centered approach to adding deduction capability to search engines-the concept of protoform, in: Intelligent Systems, 2002. Proceedings 2002 First International IEEE Symposium, vol. 1, IEEE, 2002, pp. 2–3.
[17] A. Ramos-Soto, A. Bugarín, S. Barro, J. Taboada, Linguistic descriptions for automatic generation of textual short-term weather forecasts on real prediction data, IEEE Trans. Fuzzy Syst. 23 (1) (2015) 44–57, https://doi.org/10.1109/tfuzz.2014.2328011.
[18] P. Conde-Clemente, J.M. Alonso, G. Triviño, Toward automatic generation of linguistic advice for saving energy at home, Soft Comput. 22 (2) (2018) 345–359, https://doi.org/10.1007/s00500-016-2430-5.
[19] J. Kacprzyk, S. Zadrozny, Computing with words is an implementable paradigm: fuzzy queries, linguistic data summaries, and natural-language generation, IEEE Trans. Fuzzy Syst. 18 (3) (2010) 461–472, https://doi.org/10.1109/tfuzz.2010.2040480.
[20] N. Marín, D. Sánchez, On generating linguistic descriptions of time series, Fuzzy Sets Syst. 285 (2016) 6–30, https://doi.org/10.1016/j.fss.2015.04.014.
[21] A. Alvarez-Alvarez, G. Trivino, Linguistic description of the human gait quality, Eng. Appl. Artif. Intell. 26 (1) (2013) 13–23, https://doi.org/10.1016/j.engappai.2012.01.022.
[22] G. Smits, P. Nerzic, O. Pivert, M. Lesot, Efficient generation of reliable estimated linguistic summaries, in: 2018 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2018, Rio de Janeiro, Brazil, July 8-13, 2018, IEEE, 2018, pp. 1–8.
[23] K. Kaczmarek-Majer, O. Hryniewicz, M. Dominiak, L. Święcicki, Personalized linguistic summaries in smartphone-based monitoring of bipolar disorder patients, in: 11th Conference of the European Society for Fuzzy Logic and Technology, EUSFLAT 2019, 2019/2008, pp. 400–407.
[24] R. Castillo-Ortega, N. Marín, D. Sánchez, Linguistic local change comparison of time series, in: 2011 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2011, IEEE, 2011, pp. 2909–2915.
[25] J. Kacprzyk, S. Zadrozny, Linguistic summarization of the contents of web server logs via the ordered weighted averaging (OWA) operators, Fuzzy Sets Syst. 285 (2016) 182–198, https://doi.org/10.1016/j.fss.2015.07.020.
[26] G. Moyse, M. Lesot, Linguistic summaries of locally periodic time series, Fuzzy Sets Syst. 285 (2016) 94–117, https://doi.org/10.1016/j.fss.2015.06.016.
[27] R.J. Almeida, M. Lesot, B. Bouchon-Meunier, U. Kaymak, G. Moyse, Linguistic summaries of categorical time series for septic shock patient data, in: FUZZ-IEEE 2013, IEEE International Conference on Fuzzy Systems, Proceedings, Hyderabad, India, 7-10 July, 2013, IEEE, 2013, pp. 1–8.
[28] A. Wilbik, R.M. Dijkman, Linguistic summaries of process data, in: 2015 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2015, Istanbul, Turkey, August 2-5, 2015, IEEE, 2015, pp. 1–7.
[29] A. Wilbik, J.M. Keller, Anomaly detection from linguistic summaries, in: FUZZ-IEEE 2013, IEEE International Conference on Fuzzy Systems, Proceedings, Hyderabad, India, 7-10 July, 2013, IEEE, 2013, pp. 1–7.
[30] S. Aydogan, D. Akay, F.E. Boran, R.R. Yager, An extension of fuzzy linguistic summarization considering probabilistic uncertainty, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 26 (2) (2018) 195–216, https://doi.org/10.1142/S0218488518500101.
[31] R.M. Castillo-Ortega, N. Marín, D. Sánchez, A fuzzy approach to the linguistic summarization of time series, J. Mult.-Valued Log. Soft Comput. 17 (2011).
[32] R. Castillo-Ortega, N. Marín, D. Sánchez, A.G. Tettamanzi, Linguistic Summarization of Time Series Data Using Genetic Algorithms, EUSFLAT, vol. 1, Atlantis Press, 2011, pp. 416–423.
[33] T. Altintop, R.R. Yager, D. Akay, F.E. Boran, M. Ünal, Fuzzy linguistic summarization with genetic algorithm: an application with operational and financial healthcare data, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 25 (04) (2017) 599–620, https://doi.org/10.1142/s021848851750026x.
[34] C.A. Donis-Díaz, R. Bello, J. Kacprzyk, Using ant colony optimization and genetic algorithms for the linguistic summarization of creep data, in: Intelligent Systems' 2014, Springer, 2015, pp. 81–92.
[35] MeteoGalicia, Galician meteorological agency website, www.meteogalicia.gal, 2021. (Accessed April 2021).
[36] A.E. Eiben, J.E. Smith, et al., Introduction to Evolutionary Computing, vol. 53, Springer, 2003.
[37] P.J. Van Laarhoven, E.H. Aarts, Simulated annealing, in: Simulated Annealing: Theory and Applications, Springer, 1987, pp. 7–15.
[38] R. Tavakkoli-Moghaddam, M.-B. Aryanezhad, N. Safaei, A. Azaron, Solving a dynamic cell formation problem using metaheuristics, Appl. Math. Comput. 170 (2) (2005) 761–780, https://doi.org/10.1016/j.amc.2004.12.021.
[39] S.-W. Lin, J. Gupta, K.-C. Ying, Z.-J. Lee, Using simulated annealing to schedule a flowshop manufacturing cell with sequence-dependent family setup times, Int. J. Prod. Res. 47 (2009) 3205–3217, https://doi.org/10.1080/00207540701813210.
[40] F. Díaz-Hermida, A. Ramos-Soto, A. Bugarín, On the role of fuzzy quantified statements in linguistic summarization of data, in: 2011 11th International Conference on Intelligent Systems Design and Applications, IEEE, 2011, pp. 166–171.
[41] Google, Google Earth website, https://earth.google.com, 2021. (Accessed April 2021).
[42] L.A. Zadeh, A computational approach to fuzzy quantifiers in natural languages, in: Computational Linguistics, Elsevier, 1983, pp. 149–184.
[43] M. Delgado, D. Sánchez, M.A. Vila, Fuzzy cardinality based evaluation of quantified sentences, Int. J. Approx. Reason. 23 (1) (2000) 23–66, https://doi.org/10.1016/s0888-613x(99)00031-6.
[44] M.D. Calvo-Flores, D. Sánchez, M.A. Vila, Un método para la evaluación de sentencias con cuantificadores lingüísticos, in: Actas del VIII Congreso Español sobre Tecnologías y Lógica Fuzzy, Pamplona, 8-10 de septiembre de 1998, Departamento de Automática y Computación, 1998, pp. 193–198.

[45] A. Cascallar-Fuentes, A. Ramos-Soto, A. Bugarín-Diz, An experimental study on the use of fuzzy quantification models for linguistic descriptions of data, in: 24th European Conference on Artificial Intelligence, IOS Press, 2020, pp. 267–274.
[46] A. Bugarín, N. Marín, D. Sánchez, G. Triviño, Aspects of quality evaluation in linguistic descriptions of data, in: 2015 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2015, Istanbul, Turkey, August 2-5, 2015, IEEE, 2015, pp. 1–8.
[47] F. Díaz-Hermida, A. Bugarín, Semi-fuzzy quantifiers as a tool for building linguistic summaries of data patterns, in: Proceedings of the IEEE Symposium on Foundations of Computational Intelligence, FOCI 2011, Part of the IEEE Symposium Series on Computational Intelligence 2011, Paris, France, 11–15 April 2011, IEEE, 2011, pp. 45–52.
[48] R. Real, J.M. Vargas, The probabilistic basis of Jaccard's index of similarity, Syst. Biol. 45 (3) (1996) 380–385, https://doi.org/10.1093/sysbio/45.3.380.
[49] AccuWeather, AccuWeather website, www.accuweather.com, 2021. (Accessed April 2021).
[50] Windguru, Windguru website, www.windguru.cz, 2021. (Accessed April 2021).
[51] Meteosat, Meteosat website, www.meteosat.com, 2021. (Accessed April 2021).
[52] D.J. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, CRC Press, 2003.
[53] F. Wilcoxon, Individual comparisons by ranking methods, Biom. Bull. 1 (6) (1945) 80–83, http://www.jstor.org/stable/3001968.
[54] H.B. Mann, D.R. Whitney, On a test of whether one of two random variables is stochastically larger than the other, Ann. Math. Stat. (1947) 50–60, https://doi.org/10.1214/aoms/1177730491.
[55] C. van der Lee, A. Gatt, E. van Miltenburg, S. Wubben, E. Krahmer, Best practices for the human evaluation of automatically generated text, in: Proceedings of the 12th International Conference on Natural Language Generation, INLG 2019, Tokyo, Japan,October 29 - November 1, 2019, Association for Computational Linguistics, 2019, pp. 355–368.
[56] W.C. Mann, S.A. Thompson, Rhetorical structure theory: toward a functional theory of text organization, in: Text-Interdisciplinary, J. Study Discourse 8 (3) (1988) 243–281.