XIX Congreso Español sobre Tecnologías y Lógica Fuzzy (XIX ESTYLF)

ESTYLF 12: Sesión Especial Soft Computing y Generación del Lenguaje Natural II

Organizadores:

Alberto Bugarín, Nicolás Marín,
Alejandro Ramos, Daniel Sánchez

Descripciones lingüísticas de datos de observación meteorológica usando temple simulado

Andrea Cascallar Fuentes, Alejandro Ramos Soto, Alberto J. Bugarín Diz
{andrea.cascallar.fuentes, alejandro.ramos, alberto.bugarin.diz}@usc.es
Centro Singular de Investigación en Tecnoloxías da Información (CiTIUS), Universidade de Santiago de Compostela

***Resumen*—En este trabajo presentamos una aproximación para la generación de descripciones lingüísticas en tiempo real sobre datos de observación meteorológica proporcionados por la Agencia Meteorológica gallega (MeteoGalicia). Las descripciones son sentencias cuantificadas borrosas, que incluyen referencias geográficas imprecisas, y resultan válidas para la etapa de determinación de contenidos de un sistema de Generación de Lenguaje Natural. La generación de las descripciones va guiada por la metaheurística Temple Simulado, que permite seleccionar las descripciones lingüísticas más adecuadas, de acuerdo con un conjunto de criterios objetivos.**

***Index Terms*—Descripciones lingüísticas de datos, Generación de lenguaje natural, Computación con palabras.**

I. Introducción

Ante la ingente cantidad de datos presente actualmente en todas las facetas de la vida, la Inteligencia Artificial provee herramientas que permiten analizar conjuntos de datos con la finalidad de extraer información útil y comprensible para usuarios y expertos, basada en el potencial del lenguaje humano. Por ejemplo, la generación de lenguaje natural (NLG, en inglés) se ocupa de la generación de texto a partir de diversas fuentes de datos [1]. Dentro de este campo son de interés los sistemas *data-to-text* (D2T) [2], que generan textos a partir de conjuntos o series de datos numéricos o simbólicos.

De forma complementaria a los sistemas D2T, las descripciones lingüísticas de datos (LDD, en inglés) proporcionan mecanismos para obtener resúmenes sintéticos de conjuntos de datos numéricos, generalmente basados en el uso de sentencias cuantificadas borrosas [3]. Este tipo de aproximaciones, surgidas a partir del concepto de protoforma definido por Zadeh [4], permiten modelar la imprecisión de los términos lingüísticos inherente al lenguaje humano, aunque su uso en sistemas reales es muy limitada por el momento [5]. Según sus componentes, existen dos tipos de protoformas: tipo I ("Q Y son S") donde Q es un cuantificador, Y es un conjunto de elementos y S es un resumen; y tipo II ("Q KY son S") donde además de los componentes presentes en las de tipo I se añade un calificador K.

Es precisamente en la faceta aplicada de LDD donde se centra el objetivo de este trabajo: la generación de descripciones lingüísticas mediante temple simulado [6] sobre datos meteorológicos en tiempo real proporcionados por MeteoGalicia [7] para el conjunto de los 314 municipios de Galicia, introduciendo además el uso de referencias geográficas.

II. Trabajos relacionados

Actualmente, existen numerosos sistemas NLG, de los cuales una gran mayoría lo componen sistemas D2T [1], [8], [9]. Por ejemplo, uno de los sistemas de mayor impacto en el ámbito de la salud es el sistema BT45, dentro del proyecto BabyTalk [10], que genera informes a partir de datos recogidos durante 45 minutos sobre bebés que se encuentran en la UCI.

En el campo de LDD, la mayor parte de aproximaciones propuestas generan sentencias cuantificadas como "La mayoría de las personas son altas" (tipo I) o "La mayoría de las personas altas son rubias" (tipo II) [3], [9], [11]. Este tipo de propuestas se han aplicado a una gran variedad de casos de uso, principalmente sobre series de datos temporales, como datos de consumo energético [12], fondos de inversión [13], actividad física [14], [15] o el flujo de pacientes en hospitales [16], entre otros. En ámbitos de aplicación real, GALiWeather [5] es un sistema D2T meteorológico que emplea LDD para ciertas tareas de extracción de información, lo que ejemplifica la complementariedad que existe entre ambas disciplinas.

Otro aspecto importante en este campo es la utilización de estrategias de búsqueda, tanto heurísticas [5], [16] como basadas en algoritmos genéticos [17]–[19], para la obtención de sentencias cuantificadas con un nivel de calidad descriptiva suficiente, en problemas donde su número es demasiado grande como para obtenerlas exhaustivamente en su totalidad.

III. Motivación

Nuestro caso de uso consiste en datos de observación meteorológica proporcionados en tiempo cuasi real por MeteoGalicia [7] para las variables estado del cielo, viento y temperatura. Dada la alta frecuencia de actualización de dichos datos, aproximadamente a cada hora, se justifica la necesidad de generar descripciones en el menor tiempo posible. Además, dado que dichos datos se encuentran caracterizados geográficamente, las descripciones en nuestra propuesta contemplan también la inclusión de referencias geográficas vagas como "norte" o "este".

Concretamente, nuestra solución genera descripciones lingüísticas basadas en proposiciones cuantificadas de tipo I, donde Q es un cuantificador, X es una variable lingüística definida a partir de las variables meteorológicas y A es uno de sus valores ("En algunos ayuntamientos el cielo está despejado"); y II donde se añade un descriptor geográfico ("En algunos ayuntamientos en el Norte el cielo está despejado"), que pueden incluir una o más variables meteorológicas.

En nuestro caso, la cantidad de datos disponibles y la necesidad de disponer de una solución computacionalmente poco costosa para la obtención de las descripciones, debido a las restricciones temporales que se deben cumplir, aconsejan la utilización de una estrategia de búsqueda más simple que las de tipo evolutivo comentadas anteriormente. Por ello, proponemos utilizar la metaheurística temple simulado, de muy reducido coste computacional y que ha sido utilizada para abordar diversos problemas [20] [21], obteniendo buenas soluciones en comparación con otras metaheurísticas.

IV. Generación de descripciones meteorológicas

La solución propuesta utiliza datos numéricos para generar descripciones de observación meteorológica en tiempo real de la comunidad autónoma de Galicia.

MeteoGalicia ofrece un servicio web que muestra datos de observación sobre el estado meteorológico actual de los ayuntamientos gallegos [7], actualizados aproximadamente cada hora.

IV-A. Conocimiento del dominio

En base a las variables meteorológicas, se definen las siguientes variables lingüísticas:

- Estado del cielo: sus valores son códigos numéricos que categorizan el la cobertura nubosa y el nivel de precipitación. A partir de esta variable se crea una variable lingüística con el mismo nombre definida del mismo modo.
- Viento: los posibles valores son códigos numéricos que codifican intensidad y dirección del viento. A partir de esta variable se crea una variable lingüística con el mismo nombre que respeta la definición original.
- Temperatura: se crean dos variables lingüísticas, una para las temperaturas máximas y otra para las mínimas. La temperatura actual de un ayuntamiento se compara con las máximas y las mínimas del mes actual del registro de datos históricos. Para la temperatura actual de cada ayuntamiento t_i, las etiquetas que toman las variables lingüísticas se calculan utilizando la media $\bar{x}$ y la desviación típica σ del mes actual de los datos históricos.
  Las protoformas tipo I, siguen la plantilla "En Q ayuntamientos A", donde Q es un cuantificador y A puede ser una o varias de las estructuras definidas en la Tabla I ("En pocos ayuntamientos el estado del cielo es soleado y la temperatura es baja con respecto a las máximas y normal con respecto a las mínimas"). Se han definido siete cuantificadores borrosos ("ninguno", "pocos", "algunos", "aproximadamente la mitad", "bastantes", "casi todos", "todos"), sobre el porcentaje de ayuntamientos (PA) que verifican la sentencia.
  Las protoformas tipo II siguen la plantilla "En Q ayuntamientos en G A", donde Q y A son como en el caso anterior y G es un descriptor geográfico ("En algunos ayuntamientos del Sur el estado del cielo es soleado").

  Los descriptores geográficos, definidos de forma borrosa, son {Norte, Sur, Este, Oeste, Centro} definidos de la siguiente forma:
  - Sur: definido mediante el conjunto borroso trapezoidal de soporte [41.75, 42.57] y núcleo [41.75, 42.16]
  - CentroLatitud: definido mediante el conjunto borroso trapezoidal de soporte [42.16, 43.39] y núcleo [42.57, 42.98]
  - Norte: definido mediante el conjunto borroso trapezoidal de soporte [42.98, 43.8] y núcleo [43.39, 43.8]
  - Oeste: definido mediante el conjunto borroso trapezoidal de soporte [-9.31, -8.266] y núcleo [-9.31, -8.788]
  - CentroLongitud: definido mediante el conjunto borroso trapezoidal de soporte [-8.788, -7.222] y núcleo [-8.266, -7.744]
  - Este: definido mediante el conjunto borroso trapezoidal de soporte [-7.744, -6.7] y núcleo [-7.222, -6.7]

  Para considerar la parte geográfica de la descripción tenemos datos meteorológicos geolocalizados, es decir, se dispone de la localización de cada ayuntamiento. Para cada localización se toma la etiqueta que mejor la representa calculando el grado de cumplimiento aplicando el modelo de cuantificación de Zadeh.

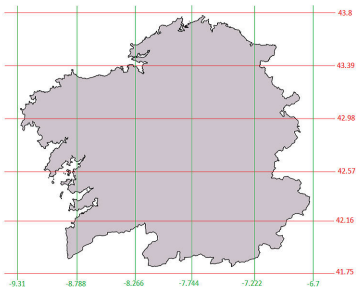


Figura 1: Coordenadas de referencia para la definición de los descriptores geográficos.

Tabla I: Plantillas de texto

Variable lingüística	**Plantilla**
Estado del cielo	el estado del cielo es <valor>
Viento	el viento tiene dirección <valor_dirección> e intensidad <valor_intensidad>
Temperatura	la temperatura es <valor_máx> con respecto a las máximas y <valor_min> con respecto a las mínimas

IV-B. Generación de descripciones lingüísticas

Debido a que el objetivo de nuestra solución es generar sentencias que describan la situación meteorológica en tiempo real, y que la gran mayoría de estaciones meteorológicas sirven datos diezminutales, hemos fijado como restricción que el tiempo de ejecución para la obtención de las descripciones no supere los 5 minutos[1]. Esto se debe a que nuestra solución

[1]Pruebas ejecutadas en un Intel Core i7-6700HQ @2.60GHz 2.59GHz con 16 GB de RAM

se centra en la fase de determinación de contenido de un sistema NLG, por lo tanto, a la hora de abordar las demás fases para desarrollar un sistema completo, el tiempo de ejecución necesario será mayor. Por esta razón y teniendo en cuenta que se busca describir el estado actual en tiempo real, a partir de la versión base hemos introducido una serie de optimizaciones.

En primer lugar, para cada ayuntamiento se calcula el grado de cumplimiento de su temperatura actual con respecto a las etiquetas que pueden tomar las dos variables lingüísticas. Para realizar este cálculo, utilizando el valor de la media y la desviación típica, para cada ayuntamiento se guarda el grado de cumplimiento para cada una de las etiquetas definidas en la Sección III. Además, para cada posible valor de la variable "temperatura" L, se calcula el grado de pertenencia del conjunto de ayuntamientos como se muestra en la expresión 1 siendo n el número de ayuntamientos, μ_L la función que evalúa el grado de cumplimiento y t_i la temperatura actual de cada ayuntamiento.

$$\mu(L) = \frac{\sum_{i=1}^{|n|} \mu_L(t_i)}{n} \tag{1}$$

IV-B1. Generación de sentencias tipo I: Esta parte del sistema se centra en generar las sentencias de tipo I a partir de los datos obtenidos.

Versión inicial. Genera todas las sentencias posibles y, para cada una de ellas, se calcula su grado de cumplimiento. Este cálculo se realiza de forma diferente si la sentencia contiene una única variable lingüística o si está compuesta por más de una.

Para las descripciones D donde solo se describe una variable se aplica el modelo de cuantificación de Zadeh. En el caso de las variables "crisp" el grado de cumplimiento es 1 si el valor actual v_i coincide con la etiqueta S y 0 en caso contrario.

$$\mu(Q\ X\ son\ A) = Q(\frac{\sum_{i=1}^{|n|} \mu_S(v_i)}{n}) \tag{2}$$

Para las descripciones D donde se describe más de una variable, debemos aplicar, siguiendo nuevamente el modelo de cuantificación de Zadeh, la conjunción de todas las variables, utilizando la t-norma mínimo:

$$\mu(Q\ X\ son\ A) = Q(\frac{\sum_{i=1}^{|n|} \mu_{S1}(v_{S1i}) \cap ... \cap \mu_{Sm}(v_{Smi})}{n}) \tag{3}$$

Al ejecutar esta versión se obtiene un total de 63.875 descripciones y la ejecución tarda en completarse aproximadamente 35 minutos.

Optimización 1. La versión anterior no cumple la restricción, por lo tanto es necesario optimizarla.

Por muy diversas que sean las condiciones meteorológicas, no se van a dar todos los posibles valores para las variables contempladas. Esto es debido, en parte, al tamaño reducido de esta región.

Para reducir el coste temporal que supone generar todas las posibles combinaciones, se propone implementar una optimización que descarte los valores no presentes en el panorama actual. De este modo se reduce notoriamente el tiempo necesario para completar una ejecución. El tiempo necesario para obtener las descripciones varía entre 50 y 60 segundos y se obtienen alrededor de 2000. Además, eliminando situaciones meteorológicas que no están presentes se obtienen descripciones más representativas del mapa.

Mejores soluciones. Aplicando esta optimización se consigue una mejora notoria, sin embargo, los resultados obtenidos no son suficientemente representativos. Para solucionar este problema se proponen las siguientes mejoras:

- Eliminar descripciones "Ninguno": este cuantificador es útil en cuanto que aísla las descripciones que no se cumplen, sin embargo, éstas no son, en general, útiles para el usuario, por lo tanto, se descartan.
- Umbral en el grado de cumplimiento: algunas sentencias describen casos poco relevantes, esto es, tienen un grado de cumplimiento muy bajo. Para evitar esta situación se define un umbral $u = 0{,}5$ eliminando aquellas sentencias que tengan un grado de cumplimiento inferior.
- Ordenación: para evaluar las descripciones se propone utilizar dos criterios además del grado de cumplimiento: cobertura del cuantificador, prefiriendo sentencias que abarquen mayor extensión y así evitar describir una misma situación mediante varias sentencias referidas a extensiones más reducidas; y tamaño de la sentencia, prefiriendo sentencias que comprendan un número mayor de variables, ya que de esta forma se reduce el número de sentencias y estas son más específicas. Las sentencias se ordenan según los criterios descritos anteriormente, priorizados por el siguiente orden: grado de cumplimiento, cobertura del cuantificador y longitud de la sentencia. Una excepción en dicho orden son las descripciones del cuantificador "Pocos", que se sitúan después de las de cuantificadores con mayor cobertura de modo que, aunque según la ordenación indicada podrían ser seleccionadas en detrimento de algunas sentencias con mayor cobertura, se colocan después de modo que el usuario obtenga información más general y pueda acceder a éstas si necesita más grado de detalle.

Una vez que se han aplicado estas mejoras, se define un número máximo de descripciones que se van a mostrar ya que un número elevado de descripciones puede provocar que el texto generado no sea útil para el usuario.

En la Tabla II se muestra una comparativa entre las diferentes versiones.

Tabla II: Resumen versiones tipo I

Versión	Tamaño solución	Tiempo de ejecución
Inicial	63875	∼35 minutos
Optimización 1	∼2000	50-60 segundos
Mejores soluciones	un máximo de 100	50-60 segundos

IV-B2. Generación de sentencias tipo II: Se parte de una versión inicial, donde se generan todas las combinaciones posibles, y a partir de ahí se proponen optimizaciones.

Para esta descripciones hay que calcular, para cada ayuntamiento, los grados de pertenencia con respecto a los descriptores geográficos. Este cálculo se realiza utilizando la coordenada correspondiente y calculado su grado de cumplimiento. Los cálculos para "Centro" difieren del resto ya que está formado por dos componentes. El proceso es el siguiente: para cada ayuntamiento se calcula el grado de cumplimiento para cada coordenada, se aplica la t-norma mínimo entre estos dos valores y al valor resultante se le aplica el descriptor geográfico, calculando el grado de cumplimiento.

$$\mu_{Centro(longitud_i,latitud_i)} = \mu_{CentroLongitud(longitud_i)} \cap \mu_{CentroLatitud(latitud_i)} \quad (4)$$

Versión inicial. Genera todas las sentencias posibles y, para cada una de ellas, se calcula su grado de cumplimiento con la finalidad de saber cuáles son las descripciones más representativas.

$$\mu(Q\ X\ son\ A\ en\ G) = Q(\frac{\sum_{i=1}^{|n|} \mu_G(x_i) \cap \mu_{S1}(x_i) \cap ... \cap \mu_{Sm}(x_i)}{\sum_{i=1}^{|n|}(\mu_G(x_i))}) \quad (5)$$

Para obtener todas las descripciones, el sistema necesita aproximadamente 3 días y se generan 1094170 descripciones, incumpliendo la restricción temporal.

Optimización 1. La primera optimización es eliminar los valores no presentes, como en las de tipo I, reduciendo el coste temporal a aproximadamente 8 horas y obteniendo alrededor de 13000 descripciones.

V. Algoritmo de búsqueda meta-heurístico

Debido a la elevada cantidad de descripciones que se generan de tipo II y al coste temporal que esto supone, se propone el uso de un algoritmo de búsqueda metaheurística para la extracción de información relevante consumiendo menos recursos.

En general, los algoritmos basados en poblaciones no parecen la mejor opción para este caso. Los algoritmos metaheurísticos basados en métodos constructivos tampoco son una buena opción ya que, en este caso, la solución inicial debe tener un mínimo de componentes. Las soluciones basadas en trayectorias pueden aplicarse en este caso, ya que utilizan una heurística de búsqueda local que explora posibles soluciones siguiendo una trayectoria en el espacio de búsqueda. Realizando un análisis de los algoritmos más utilizados se concluye que el Temple Simulado [6] es una buena alternativa. Es un algoritmo de búsqueda por entornos con un criterio probabilístico de aceptación de soluciones inspirado en la Termodinámica. En cada iteración se genera un determinado número de vecinos, con cierta probabilidad de aceptar soluciones peores para evitar que el algoritmo no se estanca en un óptimo local.

Este algoritmo cuenta con una serie de parámetros configurables de modo que, realizando un estudio empírico, se puedan establecer valores que obtengan resultados de calidad. A continuación se muestran los valores establecidos después de realizar el estudio.

- Valor inicial del parámetro de control T_0: esta variable se debe inicializar a un valor suficientemente alto ya que, si es muy bajo converge demasiado rápido y si es muy alto tarda en converger. Después de experimentar con diversos valores se inicializa con un valor proporcional al número máximo de descripciones posible, tomando como parámetros $\mu = 0{,}01$ y $\phi = 0{,}999$.

$$T_0 = (\mu / -\ln(\phi))/MAX\_SOLUCIONES \quad (6)$$

- Solución inicial S_0: después de probar diversas alternativas, S_0 se inicializa con la mejor sentencia formada por una única variable lingüística obtenida aplicando los criterios de evaluación descritos anteriormente.
- Nueva solución: la forma de generar una nueva solución es modificar [1, 4] componentes aleatoriamente. Con este método en ocasiones el algoritmo se queda estancado, para solucionar esto, se define un parámetro $maxRepeated$, que define el número máximo de soluciones repetidas en una iteración. Si se alcanza este valor, se genera una nueva solución completamente aleatoria, evitando el estancamiento.
- Velocidad y mecanismo de enfriamiento: $MAX\_CANDIDATAS = 300$ y $MAX\_ACEPTADAS = 30$ se consigue un buen balance en cuanto a soluciones peores aceptadas y la velocidad de enfriamiento. En cuanto al mecanismo de enfriamiento, se utiliza el Esquema de Cauchy.
- Condición de parada: después de experimentar con otras alternativas que no se adecúan a este problema, se experimenta con una condición de parada que está formada por dos condiciones: número máximo de iteraciones y máximo de intentos fallidos de generar nuevas soluciones para cada iteración. Para esto se definen dos parámetros: $MAX\_ITER = 2300$, que establece el número máximo de iteraciones que el algoritmo puede hacer y $MAX\_NEIGHBOUR\_ATTEMPTS = 2000$, que define el número máximo de intentos de generar una nueva solución candidata en una iteración.
- Condición de aceptación: realizamos una experimentación con diferentes opciones concluyendo que la que mejor resultados ofrece es la expresión 7, donde se da preferencia al grado de cumplimiento, aceptando las nuevas soluciones que mejoren el de la actual, y a la cobertura, aceptando soluciones con mayor cobertura aunque el grado de cumplimiento sea inferior.

$$\begin{aligned}(cobertura(S_{cand}) <= cobertura(S_{act})\ \&\&\\ \mu(S_{cand}) >= \mu(S_{act})\ \&\&\ \mu(S_{cand} > 0))\ ||\\ (cobertura(S_{cand}) > cobertura(S_{act})\ \&\&\\ \mu(S_{cand} > 0))\ ||\ (\mu(S_{cand} > 0))\ \&\&\\ aleatorio < e^{-\delta/T_k})\end{aligned} \quad (7)$$

A esta configuración se le aplican las mismas optimizaciones que en las descripciones de tipo I descritas en la Sección IV-B1:

Aplicando este algoritmo, el tiempo de ejecución es de aproximadamente 2 minutos y el conjunto de descripciones que se muestra tiene un tamaño de aproximadamente 400 sentencias ofreciendo una descripción representativa del estado meteorológico.

Tabla III: Resumen versiones tipo II

Versión	Tamaño solución	Tiempo de ejecución
Inicial	1094170	3 días
Optimización 1	∼13000	∼8 horas
SA	∼400	∼2 minutos

VI. Evaluación

Para evaluar el grado de adecuación de las descripciones a diferentes casos hemos generado cuatro mapas, en cada uno de los cuales se representa el estado de una de las variables (estado del cielo, viento, temperatura máxima y temperatura mínima). Se representa un meteoro por cada uno de los 314 ayuntamientos de Galicia.

Para las variables "estado del cielo" y "viento" los mapas se construyen utilizando los iconos que ofrece MeteoGalicia. Para los componentes de la variable "temperatura" se utilizan los siguientes códigos que representan las posibles etiquetas de la variable lingüística "temperatura": "MB" para "muy baja", "B" para "baja", "N" para "normal", "A" para "alta" y "MA" para "muy alta". Cada uno de estos códigos tiene un color desde azul oscuro para "MB" hasta rojo para "MA". Debido al elevado número de iconos, interpretar correctamente la información visual es una tarea difícil.

En las tablas IV y V se muestran las mejores descripciones con respecto a los mapas que se muestran en la Figura 2. Estas descripciones son las 10 mejores obtenidas en cada caso después de ordenarlas siguiendo los tres criterios, descritos en IV-B1. Aunque las descripciones de tipo I proporcionan una idea general sobre el estado meteorológico, por sí solas no son suficientemente informativas. En este sentido, las descripciones de tipo II son de mayor utilidad ya que, al centrarse en regiones más pequeñas tienen en cuenta situaciones que las de tipo I no consideran precisamente por su carácter general.

Para comprobar la calidad de las descripciones se han realizado 20 ejecuciones en momentos del día diferentes durante dos semanas. Las descripciones de tipo I siempre obtuvieron buenos resultados en cuanto a que describen la situación meteorológica informando sobre los valores de las variables meteorológicas que más se repiten. Las descripciones de tipo II, en general también obtuvieron resultados representativos salvo en casos en los que un fenómeno ocurre en un área muy pequeña debido a la definición demasiado amplia del descriptor geográfico (por ejemplo cielos soleados en toda la región salvo en el extremo Norte de Galicia donde está "muy nublado") de modo que no ocupa una posición relevante en el conjunto de soluciones resultante.

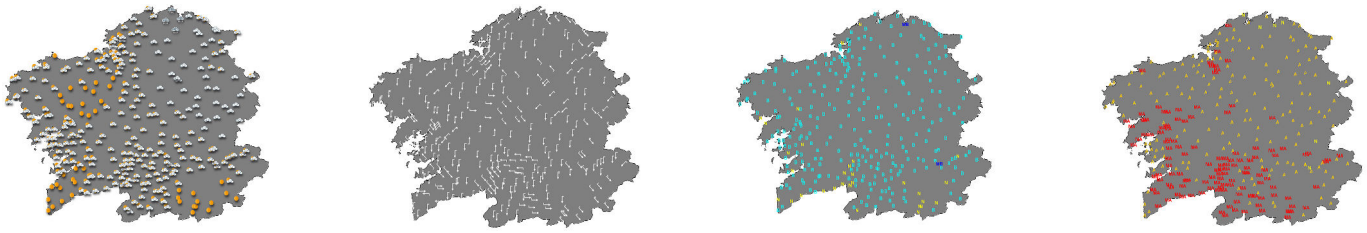
Tabla IV: Mejores 10 descripciones tipo I para 15 de septiembre a las 17:00 (Figura 2).

Descripción
En aproximadamente la mitad de ayuntamientos el estado del cielo es muy nublado
En aproximadamente la mitad de ayuntamientos el viento tiene dirección Norte e intensidad baja
En algunos ayuntamientos el cielo está muy nublado y la temperatura es baja con respecto a las máximas y alta con respecto a las mínimas
En algunos ayuntamientos el viento tiene dirección Norte e intensidad baja y la temperatura es baja con respecto a las máximas y alta con respecto a las mínimas
En algunos ayuntamientos la temperatura es baja con respecto a las máximas y muy alta con respecto a las mínimas
En bastantes ayuntamientos la temperatura es baja con respecto a las máximas y alta con respecto a las mínimas
En algunos ayuntamientos el estado del cielo es nubes y claros
En algunos ayuntamientos el estado del cielo es muy nublado y el viento tiene dirección Norte e intensidad baja
En pocos ayuntamientos el estado del cielo es despejado y el viento tiene dirección Norte e intensidad baja

Tabla V: Mejores 10 descripciones tipo II para 15 de septiembre a las 17:00 (Figura 2).

Descipción
En aproximadamente la mitad de ayuntamientos del Este la temperatura es baja con respecto a las máximas y alta con respecto a las mínimas
En algunos ayuntamientos del Oeste el viento tiene dirección Norte e intensidad baja
En algunos ayuntamientos del Oeste el estado del cielo es nubes y claros
En algunos ayuntamientos del Oeste el estado del cielo es muy nublado
En algunos ayuntamientos del Sur el viento tiene dirección Norte e intensidad baja
En algunos ayuntamientos del Sur la temperatura es baja con respecto a las máximas y alta con respecto a las mínimas
En aproximadamente la mitad de ayuntamientos del Norte la temperatura es baja con respecto a las máximas y alta con respecto a las mínimas
En algunos ayuntamientos del Este el viento tiene dirección Norte e intensidad baja
En algunos ayuntamientos del Este el estado del cielo es cubierto y la temperatura es baja con respecto a las máximas y alta con respecto a las mínimas
En algunos ayuntamientos del Sur el estado del cielo es muy nublado y la temperatura es baja con respecto a las máximas y muy alta con respecto a las mínimas

Hemos analizado cuantitativamente la calidad de las descripciones generadas con la metaheurística frente a la totalidad de descripciones posibles (en casos de uso donde esto último resultaba factible). Los resultados indican que la metaheurística genera: i) el 25 % de las descripciones de mayor calidad (utilizando como métrica de calidad la suma de las tres condiciones de evaluación normalizadas) si descartamos las que incluyen el cuantificador de menor cobertura espacial (el menos específico, "pocos ayuntamientos"), ii) el 40 % de las descripciones que incluyen los tres cuantificadores de mayor cobertura y iii) el 75 % de las descripciones que incluyen los dos cuantificadores de mayor cobertura ("casi todos" y "bastantes"). Por lo tanto, se puede concluir que la metaheurística es efectiva para los cuantificadores que suponen mayor cobertura espacial (los más específicos).



(a) Estado del cielo (b) Viento (c) Temperaturas máximas (d) Temperaturas mínimas

Figura 2: Mapas que definen el estado meteorológico del 15 de septiembre de 2017 a las 17:00 descrito en IV y V.

Por último, hemos evaluado la eficacia de la metaheurística, comparando la calidad de la solución final frente a la solución inicial. Aquí se observa que la media de mejora de las soluciones es del 25 % para cinco ejecuciones realizadas.

VII. Conclusiones

En este trabajo hemos presentado una aproximación basada en fuerza bruta para la generación de descripciones lingüísticas compuestas de sentencias cuantificadas de tipo I y temple simulado para las de tipo II. Dichas descripciones se generan a partir de datos meteorológicos de observación, sobre un conjunto de variables lingüísticas tanto *crisp* como borrosas, entre las que destacan la inclusión de referencias geográficas. Hemos comparado las sentencias obtenidas con mapas para comprobar si eran representativas. Además, para el caso de las sentencias tipo II, se ha evaluado que la metaheurística genera entre el 25 % y el 75 % de las descripciones de mayor calidad de entre todas las posibles. La metaheurística también resulta efectiva, ya que la calidad de la solución final frente a la inicial mejora en un 25 % en promedio.

Como trabajo futuro, ampliaremos el modelo para nuevas metaheurísticas que puedan compararse en cuanto a rendimiento y calidad de las descripciones generadas.

Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Economía y Competitividad (proyectos TIN2014-56633-C3-1-R y TIN2017-84796-C2-1-R) y la Consellería de Educación de la Xunta de Galicia (proyectos GRC2014/030 y Acreditación 2016-2019, ED431G/08"). Todos los proyectos fueron cofinanciados por el programa FEDER. A. Ramos-Soto agradece la financiación de la Consellería de Cultura, Educación e Ordenación Universitaria" (Beca Postdoctoral ED481B 2017/030).

Referencias

[1] A. Gatt and E. Krahmer, "Survey of the state of the art in natural language generation: Core tasks, applications and evaluation," *Journal of Artificial Intelligence Research*, vol. 61, pp. 65–170, 2018.

[2] E. Reiter, "An architecture for data-to-text systems," in *Proceedings of the Eleventh European Workshop on Natural Language Generation*. Association for Computational Linguistics, 2007, pp. 97–104.

[3] N. Marín and D. Sánchez, "On generating linguistic descriptions of time series," *Fuzzy Sets and Systems*, vol. 285, pp. 6–30, 2016.

[4] 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.

[5] A. Ramos-Soto, A. J. Bugarin, S. Barro, and J. Taboada, "Linguistic descriptions for automatic generation of textual short-term weather forecasts on real prediction data," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 1, pp. 44–57, 2015.

[6] P. J. Van Laarhoven and E. H. Aarts, "Simulated annealing," in *Simulated annealing: Theory and applications*. Springer, 1987, pp. 7–15.

[7] MeteoGalicia: servicio de datos de observación en tiempo real. (2017). [Online]. Available: http://servizos.meteogalicia.gal/rss/observacion/observacionConcellos.action

[8] J. Bateman and M.Zock. NLG systems wiki. (2017). [Online]. Available: http://nlg-wiki.org/systems/

[9] A. Ramos-Soto, A. Bugarín, and S. Barro, "On the role of linguistic descriptions of data in the building of natural language generation systems," *Fuzzy Sets and Systems*, vol. 285, pp. 31–51, 2016.

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

[11] J. Kacprzyk and S. Zadrozny, "Computing with words is an implementable paradigm: fuzzy queries, linguistic data summaries, and natural-language generation," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 3, pp. 461–472, 2010.

[12] A. van der Heide and G. Triviño, "Automatically generated linguistic summaries of energy consumption data," in *Intelligent Systems Design and Applications, 2009. ISDA'09. Ninth International Conference on*. IEEE, 2009, pp. 553–559.

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

[14] D. Sanchez-Valdes, L. Eciolaza, and G. Trivino, "Linguistic description of human activity based on mobile phone's accelerometers." in *IWAAL*. Springer, 2012, pp. 346–353.

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

[16] R. M. Castillo-Ortega, N. Marín, and D. Sánchez, "A Fuzzy Approach to the Linguistic Summarization of Time Series." *Journal of Multiple-Valued Logic & Soft Computing*, vol. 17, 2011.

[17] R. Castillo-Ortega, N. Marín, D. Sánchez, and A. G. Tettamanzi, "Linguistic summarization of time series data using genetic algorithms," in *EUSFLAT*, vol. 1, no. 1. Atlantis Press, 2011, pp. 416–423.

[18] C. A. Donis-Díaz, R. Bello, and 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.

[19] T. Altintop, R. R. Yager, D. Akay, F. E. Boran, and M. Ünal, "Fuzzy linguistic summarization with genetic algorithm: An application with operational and financial healthcare data," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 25, no. 04, pp. 599–620, 2017.

[20] R. Tavakkoli-Moghaddam, M.-B. Aryanezhad, N. Safaei, and A. Azaron, "Solving a dynamic cell formation problem using metaheuristics," *Applied Mathematics and Computation*, vol. 170, no. 2, pp. 761–780, 2005.

[21] S.-W. Lin, J. Gupta, K.-C. Ying, and Z.-J. Lee, "Using simulated annealing to schedule a flowshop manufacturing cell with sequence-dependent family setup times," *International Journal of Production Research*, vol. 47, pp. 3205–3217, june 2009.

Modelado lingüístico y síntesis de series temporales heterogéneas de consumo energético

S. Martínez-Municio*, L. Rodríguez-Benítez*, E. Castillo-Herrera*, J. Giralt-Muiña*, L. Jiménez-Linares*
*Escuela Superior de Informática, Universidad de Castilla-La Mancha, Paseo de la Universidad s/n. 13071. Ciudad Real, España
{sergio.martinez, luis.rodriguez, ester.castillo, juan.giralt, luis.jimenez}@uclm.es

***Resumen*—En la actualidad, gracias a la presencia de sensores o al auge de tecnologías propias del *internet de las cosas*, podemos monitorizar y registrar los consumos energéticos obtenidos en los edificios a lo largo del tiempo. Mediante un análisis efectivo de estos datos que capturen los patrones de consumo, se pueden conseguir reducciones significativas de los mismos, contribuyendo a su sostenibilidad. En este trabajo proponemos un marco de trabajo a partir del cual definir modelos que capturen esta casuística, haciendo acopio de un conjunto de series temporales de consumo eléctrico. El objetivo de estos modelos es obtener un resumen lingüístico que describa en lenguaje natural cual es la situación consumista de un edificio o conjunto de edificios en un periodo temporal concreto. La definición de estas descripciones se ha resuelto mediante resúmenes lingüísticos difusos, y como novedad en este campo, proponemos una extensión de los mismos que capturen situaciones donde la pertenencia a los conjuntos difusos no resulte muy marcada (i.e. que no supere un cierto valor umbral, como por ejemplo, 0.8), obteniendo una semántica enriquecida. Para la experimentación, se hará uso de datos de consumo energético de una organización educativa sobre los que se definirán los modelos propuestos y se obtendrán los resúmenes lingüísticos asociados a los mismos, para demostrar la capacidad que otorgan estas técnicas a la hora de obtener conclusiones sobre los diferentes escenarios de consumo en términos lingüísticos.**

***Palabras Clave*—consumo energético, modelo difuso, resúmenes lingüísticos, series temporales**

I. Introducción

La energía constituye uno de los pilares fundamentales para el desempeño adecuado de las actividades imperantes en la sociedad actual. De acuerdo con la Agencia de la Energía [1], los edificios representan el ámbito de mayor consumo energético, lo que conlleva a una responsabilidad directa en un tercio de las emisiones mundiales de dióxido de carbono. Del mismo modo, se ha reportado que los edificios operan de manera ineficiente [2] y por lo tanto, poder entender los patrones de consumo de los mismos resulta de especial interés para optimizar los recursos de las organizaciones, tanto económicos como de uso de la energía, y ser más sostenibles desde el punto de vista medioambiental [3]. En la actualidad, gracias a las tecnologías existentes así como al auge de otras nuevas, como los sensores inteligentes o redes complejas que configuran el *internet de las cosas*, junto con el interés cada vez más acuciante por las técnicas de aprendizaje máquina [4], se dispone de una cantidad de datos sin precedentes que permiten obtener un conocimiento detallado del uso que se está haciendo de la energía en un instante temporal dado en términos cuantitativos; sin embargo, resulta fundamental incorporar modelos que caractericen estos patrones de consumo energético en términos cualitativos, de modo que permitan aumentar su expresividad mediante descripciones breves en lenguaje natural con el propósito de planificar políticas que contribuyan a una mayor sostenibilidad de los edificios en un marco de revisión y ejecución continua. Dichas descripciones vendrán dadas por medio de resúmenes lingüísticos, ya que han demostrado su utilidad en diferentes ámbitos, como a la hora de describir el tráfico [5] o evaluar la calidad de la forma de andar de las personas [6]. Estos resúmenes están basados en la propuesta de Yager [7], quien proporciona una definición formal basada en el concepto de protoformas introducido por Zadeh [8]:

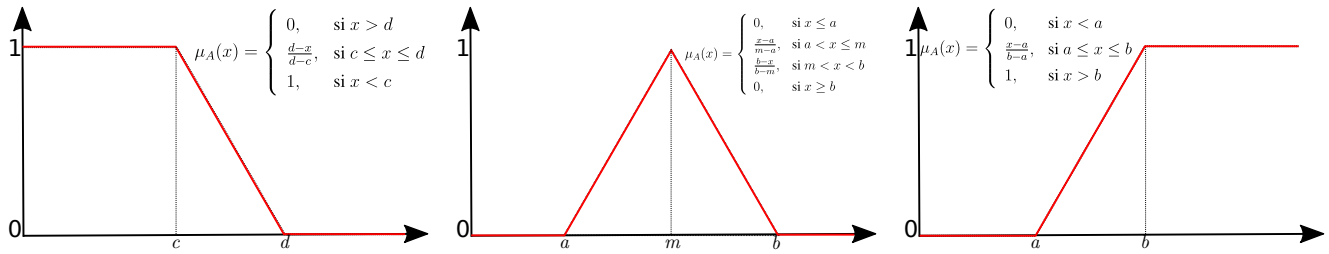
$$Q[R]y \; are \; P \tag{1}$$

donde y es un objeto caracterizado por un atributo con un dominio finito; Q y R (opcional) son cuantificadores, que llevan asociado un valor lingüístico definido sobre el dominio del atributo de y, y P un resumen que lleva asociado otro valor lingüístico. Además, se han realizado propuestas de generación automática de descripciones lingüísticas alineadas con la temática de este trabajo, donde la principal diferencia radica en que generan dichas descripciones partiendo de las series temporales de un día, sin definir un modelo previo que segmente dichos datos temporales en patrones de consumo [9].

La estructura del trabajo es la siguiente: en la Sección II se introducen los conceptos de modelo de edificio y organizacional, como principales abstracciones de caracterización de consumos, así como la definición de resumen lingüístico. En la Sección III se estudia la validez de los modelos definidos en términos lingüísticos. Una extensión de los resúmenes clásicos que aumenta las capacidades semánticas de los mismos es presentada en la Sección IV, mientras que en la Sección V se lleva a cabo la experimentación mediante la aplicación de los modelos definidos sobre datos de consumo real. Finalmente, la Sección VI presenta las conclusiones obtenidas y establece futuras líneas de trabajo.

II. Definición de Modelos

Se propone la definición de modelos para caracterizar el consumo energético obtenido en grandes instituciones con el fin de obtener en términos lingüísticos las conclusiones



(a) Trapezoidal R(a,b,c) (b) Triangular T(a,b,c) (c) Trapezoidal L(a,b,c)

Figura 1: Funciones de pertenencia difusas empleadas

derivadas de los mismos. Para ello, en primer lugar, se define un modelo base (modelo de edificio) que permita categorizar a nivel de edificio cómo ha sido su consumo por medio de una segmentación de los datos que es resuelta con el algoritmo *k*-means, para posteriormente, definir un modelo general (modelo de organización) que permita establecer un marco comparativo entre los modelos básicos para saber cómo se han comportado con respecto a los demás y obtener esta conclusión en términos lingüísticos. Como caso de estudio, se emplearán datos de consumo provenientes de una institución educativa española geográficamente distribuida: la Universidad de Castilla-La Mancha (UCLM), que cuenta con un total de 97 edificios repartidos por toda la región. En concreto, se utilizarán datos de consumo por día del año 2016 para la construcción de los modelos, y del año 2017 para la validación de los mismos.

II-A. Modelo de Edificio

Un modelo M de un edificio i se define como un conjunto de k grupos que tienen una semántica asociada:

$$M^i = \{G_0, G_1, ..., G_{k-1}\} \tag{2}$$

donde cada G_j viene definido por un prototipo, c_j:

$$G_j = \{c_j\} \tag{3}$$

Cada uno de los modelos de edificio M^i estará compuesto por dos grupos que caracterizarán los siguientes patrones de consumo: periodos con unos niveles bajos o de no actividad, que se corresponderá con G_0, y periodos donde existe una actividad relevante, que se corresponderá con G_1. Por tanto, los términos {no actividad, actividad} representan la semántica asociada a los grupos del modelo de edificio M^i. De este modo, en la Tabla I aparecen recogidos un subconjunto de los diferentes modelos de edificio que han sido generados, donde cada columna, representa la caracterización del consumo como no actividad (G_0) y actividad (G_1) respectivamente, y cada fila el edificio en cuestión. Así, por ejemplo, tenemos que el modelo para el edificio 95 quedaría formalizado como: $M^{95} = \{276.31, 1994.66\}$, de manera que, si se dispone del consumo real x para un día concreto, podemos saber cómo ha sido dicho consumo de acuerdo a su modelo, es decir, obtener su categoría semántica (caracterización).

Tabla I: Modelos de Edificio de la UCLM - Año 2016

		Modelos definidos	
		G_0	G_1
Edificios	0	500.44	1215.19
	1	570.23	1213.66
		...	
	95	276.31	1994.66
	96	211.07	406.82

II-B. Modelo de Organización

Un modelo organizacional es un modelo de modelos compuesto por cada uno de los sub-modelos de edificio M^i que queda definido como:

$$O = \begin{Bmatrix} O_{G_0} = \{M^0_{G_0}, M^1_{G_0}, ..., M^{|M|}_{G_0}\} \\ O_{G_1} = \{M^0_{G_1}, M^1_{G_1}, ..., M^{|M|}_{G_1}\} \\ \vdots \\ O_{G_k} = \{M^0_{G_k}, M^1_{G_k}, ..., M^{|M|}_{G_k}\} \end{Bmatrix} \tag{4}$$

donde cada O_{G_j} se refiere al conjunto de modelos de edificio $M^i_{G_j}$ que caracteriza cada patrón de consumo conforme a lo expuesto en el apartado anterior. Así pues, haciendo uso de (4), definimos el modelo organizacional para la UCLM como:

$$O_{\text{UCLM}} = \begin{Bmatrix} O_{G_0} = \{M^0_{G_0}, M^1_{G_0}, ..., M^{96}_{G_0}\} \\ O_{G_1} = \{M^0_{G_1}, M^1_{G_1}, ..., M^{96}_{G_1}\} \end{Bmatrix} \tag{5}$$

donde O_{G_0} vendrá dado por el dominio formado por todos los prototipos de los modelos de edificio M^i correspondientes a la semántica de no actividad:

$$Dom(O_{G_0}) = \{500.44, 570.23, ..., 211.07\} \tag{6}$$

y O_{G_1} por los de la semántica de actividad:

$$Dom(O_{G_1}) = \{1215.29, 1213.66, ..., 406.82\} \tag{7}$$

Partiendo de este modelo de organización O_{UCLM}, en el siguiente apartado se discutirá la obtención de los resúmenes lingüísticos basados en técnicas difusas que describirán el comportamiento consumista de la organización.

II-C. Caracterización lingüística

La caracterización lingüística será realizada en base a (1), donde y será un sintagma significativo, R vendrá dado por la semántica asociada al modelo, y P será definido en términos difusos. Un caso especial ocurre con Q, pues será tratado de manera conjunta con y y no como un modificador independiente con el que evaluar la sentencia completa (i.e. obtener el grado de verdad de la frase cuantificada lingüísticamente). De esta manera, (1) puede ser simplificada a la siguiente forma:

$$[R]y\ are\ P \tag{8}$$

La formalización del resumen P se lleva a cabo haciendo uso de una variable lingüística, $\mathcal{L}_v$, que está definida mediante n conjuntos difusos ($S^{'}$) que configuran una partición difusa del dominio cuyo universo de discurso viene dado por el conjunto de todos los prototipos que conforman cada dominio del modelo O:

$$Dom(\mathcal{L}_v) = Dom(O_{G_j}) \tag{9}$$

A su vez, cada conjunto difuso de $\mathcal{L}_v$ queda definido mediante una función de pertenencia compuesta por tres parámetros P_r, que representan el valor del percentil r, pues permiten conocer el posicionamiento de cada edificio i en lo que a consumo se refiere, con respecto al total de la organización. En la Fig. 1 se muestran las funciones de pertenencia consideradas para la definición de los valores de $\mathcal{L}_v$.

$$\mathcal{L}_v = \{S_0^{'}, S_1^{'}, ..., S_m^{'}\} \tag{10}$$

Dado que el modelo O está compuesto de tantos dominios como grupos tengan los modelos de edificios que lo conforman, es necesario identificar el dominio de O al que pertenece un consumo dado x de un edificio específico i para poder aplicar la variable lingüística $\mathcal{L}_v$ en las magnitudes adecuadas. Para ello, partiendo del modelo de edificio M^i que categoriza el consumo de un edificio i, obtenemos el grupo semántico al que pertenece dicho consumo x a través de la siguiente fórmula:

$$\arg\min dist(x, G_j) \tag{11}$$

donde $dist(x, G_j)$ es una medida de distancia que cuantifica la similaridad entre el consumo x y el prototipo del grupo G_j, típicamente la euclídea. Sabiendo el grupo semántico al que pertenece x, podemos obtener el dominio de O adecuado y aplicar la variable lingüística $\mathcal{L}_v$ para obtener la etiqueta lingüística que definirá el resumen P, aplicando la operación de la `t-conorma` del máximo. Por lo tanto, si aplicamos estos conceptos al modelo de estudio, $O_{\texttt{UCLM}}$, la caracterización lingüística vendrá dada de acuerdo con (8), donde y será *consumo*, R será *actividad* o *no actividad* y P quedará definido mediante dos variables lingüísticas $\mathcal{L}_v$, $\mathcal{L}_v^0$ y $\mathcal{L}_v^1$, una para cada conjunto de modelos de edificio M^i que caracterizan los patrones de consumo de actividad y no actividad en $O_{\texttt{UCLM}}$, es decir, para O_{G_0} y O_{G_1} respectivamente. Dichas $\mathcal{L}_v$ son definidas mediante cinco conjuntos difusos haciendo uso de (10), tal y como se aprecia en la Tabla II, donde la única diferencia entre $\mathcal{L}_v^0$ y $\mathcal{L}_v^1$ radica en el dominio empleado.

Tabla II: Definición de $\mathcal{L}_v$ para $O_{\texttt{UCLM}}$

Etiqueta	Función de Pertenencia (μ)
Insignificante	R$\{P_0,P_{10},P_{25}\}$
Leve	T$\{P_{10},P_{25},P_{50}\}$
Normal	T$\{P_{25},P_{50},P_{75}\}$
Grande	T$\{P_{50},P_{75},P_{90}\}$
Enorme	L$\{P_{75},P_{90},P_{100}\}$

La caracterización lingüística vista hasta ahora es aplicable a la hora de obtener conclusiones acerca de cómo ha sido el consumo obtenido durante el periodo de vigencia del modelo, pero no nos proporciona la información necesaria sobre el error cometido en la estimación del modelo de edificio M^i para poder concluir sobre la validez o bondad del mismo. Por ello, en la siguiente sección se propondrá un mecanismo que permita la validación de los modelos en términos lingüísticos.

III. Validación de Modelos

En esta sección se propondrá un mecanismo para conocer la validez de cada uno de los modelos de edificio M^i definidos, así como del modelo organizacional O, y de este modo saber si los modelos definidos arrojan las conclusiones adecuadas.

III-A. Validación del Modelo de edificio

En primer lugar, se debe asociar el consumo real x obtenido en un instante de tiempo en cada edificio i, con el grupo semántico G_j del modelo que mejor lo define para obtener el error cometido en la estimación del modelo por día de consumo real. A este grupo semántico lo denominaremos $\widetilde{G}$, y será el prototipo que mejor describe el consumo real x. Formalmente, $\widetilde{G}$ viene definido en términos generales mediante la siguiente función f, que combina los diferentes grados de pertenencia a los distintos grupos que conforman el modelo de edificio M^i:

$$\widetilde{G} = f\big(\mu_{G_0}(x), \mu_{G_1}(x), ..., \mu_{G_{k-1}}(x)\big) \tag{12}$$

donde $\mu_{G_j} = dist(x, G_j)$ y el grupo semántico vendrá dado por $j = \arg\max\big\{\mu_{G_0}(x), \mu_{G_1}(x), ..., \mu_{G_{k-1}}(x)\big\}$. De este modo, el error obtenido a la hora de catalogar el consumo real x en uno de los grupos semánticos vendrá dado por:

$$\epsilon = \widetilde{G} - x \tag{13}$$

lo que nos proporciona un valor que es independiente de la escala escogida, permitiéndonos comparar el error obtenido para cada grupo semántico $\widetilde{G}$. Así, para catalogar si dicho error cometido con respecto a la estimación de su modelo de edificio M^i es significativo, con un nivel de confianza del 95 %, consideramos el intervalo definido por $\pm 2\sigma$, con σ siendo la desviación típica propia del grupo semántico G_j al que pertenece el consumo real x, de modo que:

- Si $\epsilon > 2\sigma$, entonces se ha predicho un mayor consumo que el real, y por lo tanto está siendo **sobreestimado**.
- Si $-2\sigma \leq \epsilon \leq 2\sigma$, entonces se ha predicho un consumo que se muestra acorde con el real, y por lo tanto está siendo **adecuado**.

- Si $\epsilon < -2\sigma$, entonces se ha predicho un menor consumo que el real, y por lo tanto está siendo **subestimado**.

Con el error ϵ cometido por cada consumo real x, el interés ahora se centra en categorizar la significatividad de dicho error a nivel del modelo de edificio M^i, es decir, si éste se sitúa en los márgenes establecidos como adecuados o si por el contrario, se sitúa en el 5 % de las observaciones restantes, con el fin de obtener una descripción lingüística que resuma la validez del modelo estimado. Para ello, emplearemos una variable lingüística $\mathcal{L}_m$ (Tabla III) compuesta por cinco conjunto difusos, cuyo dominio vendrá dado por la cantidad de observaciones (errores) que encajan en cada categoría descrita anteriormente de acuerdo con la siguiente fórmula:

$$L_j = \frac{|\epsilon_j|}{|\epsilon|} \times 100 \tag{14}$$

donde $j \in \{sobreestimado, adecuado, subestimado\}$, $|\epsilon_j|$ será la cantidad de errores que encajan en una de las categorías definidas, y $|\epsilon|$ será la cantidad total de error cometido. Dado

Tabla III: Definición de $\mathcal{L}_m$ para validar el modelo

Etiqueta	Función de pertenencia (μ)
Insignificante	R{0,10,25}
Leve	T{10,25,50}
Normal	T{25,50,75}
Grande	T{50,75,90}
Enorme	L{75,90,100}

que se tienen tres categorías: *sobreestimado*, *adecuado* y *subestimado*, la descripción lingüística que resuma de manera global la validez del modelo de edificio M^i, vendrá dada por la categoría cuyo L_j sea máximo. Así pues, aplicando (8), el resumen lingüístico vendrá dado por: y, que será la categoría donde L_j es máximo, y P, que será la etiqueta lingüística de $\mathcal{L}_m$ asociada a la categoría j.

III-B. Validación del Modelo de organización

Los conceptos vistos en los modelos de edificio pueden ser extendidos para determinar la bondad o validez del modelo organizacional O. Para ello, hay que realizar una agregación de cada una de las categorías: *subestimado*, *adecuado* y *sobreestimado* para cada modelo de edificio M^i que componen el metamodelo O y aplicar la misma variable lingüística $\mathcal{L}_m$ que en los modelos de edificio M^i. Dicha agregación, $L_j^{'}$, viene definida por la siguiente ecuación:

$$L_j^{'} = \frac{\sum L_j}{|L_j|} \tag{15}$$

y al igual que en el caso anterior, la descripción lingüística que resuma de manera global la validez del modelo organizacional O, vendrá dada por la categoría cuyo $L_j^{'}$ sea máximo. De este modo, aplicando nuevamente (8), el resumen lingüístico vendrá dado por: y, que será la categoría donde $L_j^{'}$ es máximo, R que será el sintagma *organizacional* para distinguirlo del caso anterior, y P, que será la etiqueta lingüística de $\mathcal{L}_m$ asociada a la categoría j.

A lo largo de esta sección, se ha propuesto un método que permite evaluar cada modelo propuesto gracias al uso de resúmenes lingüísticos difusos. El empleo de conjuntos difusos para su definición, brinda la posibilidad de trabajar con límites poco definidos, de modo que la conclusión derivada de los mismos puede hacer uso con *cierto nivel de pertenencia* de varios de estos conjuntos. Cuando se produce esta casuística, los resúmenes lingüísticos basados en (8) no resultan lo suficientemente expresivos como para poner de manifiesto dicha situación. Por ello, en la siguiente sección introducimos un nuevo concepto de resumen extendido que permite expresar en términos lingüísticos conclusiones cuyos niveles de pertenencia difusa no resulten muy marcados.

IV. Resúmenes Extendidos

Los resúmenes lingüísticos deben poseer la suficiente capacidad expresiva como para no enmascarar u ocultar información que induzcan a conclusiones inexactas. Para tratar de capturar esta casuística, proponemos una modificación sobre la definición dada en (8), permitiendo la adición de un cuantificador absoluto (*cercano a, próximo a...*) [10], W, al resumen P, dando lugar a un resumen extendido $P^{'}$:

$$P^{'} = WP \tag{16}$$

de modo que es capaz de modelar la descripción lingüística en términos de dos etiquetas lingüísticas con un *nexo* que le proporciona la semántica adecuada. El criterio para discernir si resulta necesario la adición del cuantificador W al resumen P para obtener una descripción que capture este tipo de casuísticas en términos lingüísticos vendrá dado por un umbral de pertenencia δ asociado a cada conjunto difuso, de modo que: si el valor de pertenencia μ es inferior a dicho umbral, por ejemplo $\delta = 67\,\%$, entonces es necesario emplear un resumen extendido $P^{'}$; en caso contrario, un resumen P clásico resulta suficiente. De este modo, la definición dada en (8) quedaría como sigue:

$$[R]y \; are \; P^{'} \tag{17}$$

V. Resultados Experimentales

En esta sección se prueba la capacidad expresiva que proporcionan los resúmenes lingüísticos a la hora de resumir el estado relativo al consumo energético en una organización, que como se mencionó en la Sección II, será la Universidad de Castilla-La Mancha. Para ello, se emplearán datos de consumo diario del año 2017 para el edificio 62, cuya área de desempeño es docente e investigadora, en el cual destacamos un primer periodo de no actividad, correspondiente al periodo vacacional de Semana Santa, y otro segundo de actividad, correspondiente a un periodo lectivo, seleccionando un día cualesquiera de cada periodo con los que formalizar los resúmenes lingüísticos: $x_{\text{na}} = 1203.42$ y $x_{\text{a}} = 2036.24$.

V-A. Caracterización del consumo respecto a M^{62}

De acuerdo a (2), el modelo de edificio viene dado por:

$$M^{62} = \{1070.84, 1563.12\}$$

mientras que el modelo de la organización $O_{\texttt{UCLM}}$ es el mismo que (4). Para obtener la caracterización lingüística asociada a los consumos de no actividad (x_{na}) y actividad (x_{a}), en primer lugar definimos la variable lingüística $\mathcal{L}_v$ con la que formalizar el resumen lingüístico, cuya definición aparece en la Tabla IV. Acto seguido, se identifica el dominio de $O_{\texttt{UCLM}}$

Tabla IV: Definición de $\mathcal{L}_v$ para $O_{\texttt{UCLM}}$

Etiqueta	Función de Pertenencia (μ)
Insignificante	R{P_0,P_{10},P_{25}}
Leve	T{P_{10},P_{25},P_{50}}
Normal	T{P_{25},P_{50},P_{75}}
Grande	T{P_{50},P_{75},P_{90}}
Enorme	L{P_{75},P_{90},P_{100}}

sobre el que aplicar la variable lingüística $\mathcal{L}_v$ para cada uno de los consumos haciendo uso de (11):

$$\arg\min\{dist(1203.42, 1070.84), dist(1203.42, 1563.12)\} = 0$$

$$\arg\min\{dist(2036.24, 1070.84), dist(2036.24, 1563.12)\} = 1$$

lo que nos indica que para x_{na}, su grupo semántico se corresponde con G_0, y que para x_{a} es G_1, algo lógico si atendemos a su modelo de edificio. Sabiendo los grupos semánticos de x_{na} y x_{a}, se concluye que el dominio de $O_{\texttt{UCLM}}$ sobre el que aplicar $\mathcal{L}_v$ para cada caso son los definidos en (6) y (7) respectivamente. De este modo, obtenemos la pertenencia de x_{na} y x_{a} a cada conjunto difuso definido en $\mathcal{L}_v$, y mediante la `t-conorma` del máximo, se obtiene el conjunto al que pertenece x_{na}:

$$\begin{aligned}\text{máx}\{&\mu_{insignificante}(x) = 0, \mu_{leve}(x) = 0,\\ &\mu_{normal}(x) = 0, \mu_{grande}(x) = 0,\\ &\mu_{enorme}(x) = 1\} = \mu_{\textbf{enorme}}(x)\end{aligned}$$

y x_{a}:

$$\begin{aligned}\text{máx}\{&\mu_{insignificante}(x) = 0, \mu_{leve}(x) = 0,\\ &\mu_{normal}(x) = 0, \mu_{grande}(x) = 0,\\ &\mu_{enorme}(x) = 1\} = \mu_{\textbf{enorme}}(x)\end{aligned}$$

Una vez se dispone del conjunto difuso al que pertenece el consumo, ya se puede formalizar el resumen lingüístico de acuerdo a (8). Para el caso del consumo de no actividad, x_{na}, se tiene que:

«el consumo está siendo *enorme*»

lo cual es debido a que el edificio seleccionado posee un centro de datos que está funcionando continuamente, y para el caso del consumo de actividad, x_{a}, se tiene que:

«el consumo está siendo *enorme*»

lo cual es debido a un pico de consumo, ya sea por ser un día veraniego, con el consiguiente consumo en los sistemas de aire acondicionado, o acumulación de horas lectivas en ese periodo.

V-B. *Comparación de M^{62} respecto a O_{UCLM}*

El experimento anterior proporciona información sobre el consumo obtenido por el edificio 62 comparado con la estimación resuelta por su modelo de edificio M^{62}. Sin embargo, también es posible obtener cómo se sitúa dicho modelo de edificio M^{62} con respecto al modelo de la organización $O_{\texttt{UCLM}}$. Para ello, debemos seleccionar el prototipo del modelo de edificio que se quiere comparar respecto al modelo organizacional. Si se selecciona $G_0 = 1070.84$, empleando la misma variable lingüística $\mathcal{L}_v$ usando como dominio el definido en (6), tenemos que el conjunto difuso al que pertenece G_0, aplicando la `t-conorma` del máximo es:

$$\begin{aligned}\text{máx}\{&\mu_{insignificante}(x) = 0, \mu_{leve}(x) = 0,\\ &\mu_{normal}(x) = 0, \mu_{grande}(x) = 0,\\ &\mu_{enorme}(x) = 1\} = \mu_{\textbf{enorme}}(x)\end{aligned}$$

mientras que si se selecciona $G_1 = 1563.12$, usando $\mathcal{L}_v$ sobre el dominio definido en (7), tenemos que el conjunto difuso al que pertenece G_1, aplicando la `t-conorma` del máximo es:

$$\begin{aligned}\text{máx}\{&\mu_{insignificante}(x) = 0, \mu_{leve}(x) = 0,\\ &\mu_{normal}(x) = 0, \mu_{grande}(x) = 0.49,\\ &\mu_{enorme}(x) = 0.51\} = \mu_{\textbf{enorme}}(x)\end{aligned}$$

de modo que los resúmenes asociados son, para el caso de G_0:

«el consumo de *no actividad* está siendo enorme»

y para el caso de G_1:

«el consumo de *actividad* está siendo enorme»

lo que nos sugiere que el edificio seleccionado es de los que más consumen de toda la Universidad, tanto en periodos donde no existe gran actividad docente o investigadora, como en periodos donde sí la hay. Destacar que para el caso de G_1, hubiese sido más adecuado emplear un resumen extendido de acuerdo a (17), dándonos como resultado:

«el consumo de *actividad* está siendo *grande con tendencia a enorme*»

V-C. *Validación de Modelos*

En este apartado llevaremos a cabo la validación del modelo de edificio M^{62} y el de la organización $O_{\texttt{UCLM}}$. Para el primer caso es necesario calcular el error ϵ cometido a la hora de estimar el modelo M^{62}. Por cada día de consumo del año 2017, se debe obtener el grupo semántico $\widetilde{G}$ de su modelo al que pertenece, M^{62}, para poder calcular dicho error de acuerdo a (12) y (13) respectivamente. Por ejemplo, utilizando los datos de x_{na} y x_{a} del experimento anterior, se tiene que, $\widetilde{G}_{\text{na}} = G_0 = 1070.84$ y $\widetilde{G}_{\text{a}} = G_1 = 1563.12$, por lo que los errores obtenidos en esos días concretos son: $\epsilon_{\text{na}} = 1070.84 - 1203.42 = -132.58$ y $\epsilon_{\text{a}} = 1563.12 - 2036.24 = -473.12$. Una vez se tienen todos los errores

ϵ cometidos en la estimación, el siguiente paso consiste en categorizar su significatividad a nivel de M^{62} aplicando la variable lingüística $\mathcal{L}_m$ definida en la Tabla III, donde el L_j para cada categoría junto con la pertenencia asociada a cada conjunto difuso de $\mathcal{L}_m$ identificada viene expresada en la Tabla V.

Tabla V: Definición de $\mathcal{L}_m$ para validar el modelo

Sobreestimado		***Adecuado***		***Subestimado***		
L_j	μ	L_j	μ	L_j	μ	
	1		0		**0.9**	**Insignificante**
	0		0		0.1	**Leve**
0 %	0	89 %	0	11 %	0	**Normal**
	0		0		0	**Grande**
	0		**1**		0	**Enorme**

Luego como el valor de L_{adecuado} resulta ser máximo, con una pertenencia total a la etiqueta enorme, podemos concluir que:

«el modelo resulta adecuado de manera *enorme*»

o lo que es lo mismo, que el modelo M^{62} captura correctamente los patrones de consumo subyacentes y nos arroja estimaciones que son correctas.

Por otro lado, para validar el modelo O_{UCLM}, se agrega cada una de las categorías de acuerdo a (15), cuyo resultado aparece reflejado en la Tabla VI. Aplicando la variable lingüística $\mathcal{L}_m$

Tabla VI: Pertenencias de L_j sobre $\mathcal{L}_m$

counter	L_{sobre}	$L_{adecuado}$	L_{sub}
0	9 %	91 %	0 %
1	4 %	96 %	0 %
...			
95	13 %	87 %	1 %
96	23 %	77 %	0 %
	9.35 %	**87.14 %**	**3.51 %**

sobre cada categoría $L_j^{'}$ agregada, obtenemos las pertenencias a cada conjunto difuso de $\mathcal{L}_m$, cuyos valores se pueden apreciar en la Tabla VII. Así pues, como el valor $L_{adecuado}^{'}$

Tabla VII: Pertenencias de $L_j^{'}$ sobre $\mathcal{L}_m$

Sobreestimado		***Adecuado***		***Subestimado***		
$L_j^{'}$	μ	$L_j^{'}$	μ	$L_j^{'}$	μ	
	1		0		**1**	**Insignificante**
	0		0		0	**Leve**
9 %	0	87 %	0	4 %	0	**Normal**
	0		0.20		0	**Grande**
	0		**0.80**		0	**Enorme**

resulta ser máximo, con una pertenencia de 0.8 a la etiqueta enorme, podemos concluir que:

«el modelo *organizacional* resulta adecuado de manera *enorme*»

o lo que es lo mismo, que el modelo O_{UCLM} captura correctamente los patrones de consumo encapsulados por cada modelo de edificio que lo conforma, y que por tanto, arroja estimaciones adecuadas.

VI. Conclusiones y Trabajo Futuro

Este trabajo propone un nuevo enfoque a la hora de analizar y extraer conclusiones a partir de un conjunto de series temporales de datos de consumo energético heterogéneas (i.e. provenientes de múltiples edificios enmarcados en una misma organización), mediante la definición de modelos que resuman en términos lingüísticos difusos la situación consumista de la organización, con el propósito de servir de apoyo a la toma de decisiones de la alta dirección a la hora de acometer políticas energéticas que contribuyan a la configuración de edificios sostenibles. Además, se ha propuesto una extensión de los resúmenes lingüísticos clásicos que permite tratar la casuística donde la conclusión está mejor definida en término de dos etiquetas si el umbral de pertenencia a la mismas no resulta muy marcado. Finalmente, futuras líneas de trabajo deberían estar encaminadas a obtener un aumento del rendimiento del modelo propuesto, ya sea realizando una segmentación más fina de los grupos, o empleando modelos basados en otro tipo de técnicas, como *deep learning*; definir una arquitectura *big data* basada en microservicios que dé soporte a la definición y manipulación de los modelos o incorporar al modelo un sistema de alertas basado en resúmenes lingüísticos.

Agradecimientos

Los autores quieren agradecer al Ministerio de Economía, Industria y Competitividad de España por el apoyo ofrecido mediante el proyecto TIN2015-64776-C3-3-R, cofinanciado por el Fondo Europeo de Desarrollo Regional (FEDER).

Referencias

[1] O. for Economic Co-operation and Development, *Transition to Sustainable Buildings: Strategies and Opportunities to 2050*, ser. Energy technology perspectives. OECD, 2013.

[2] M. A. Piette, S. K. Kinney, and P. Haves, "Analysis of an information monitoring and diagnostic system to improve building operations," *Energy and Buildings*, vol. 33, no. 8, pp. 783–791, 2001.

[3] L. Hernández, C. Baladrón, J. M. Aguiar, B. Carro, and A. Sánchez-Esguevillas, "Classification and clustering of electricity demand patterns in industrial parks," *Energies*, vol. 5, no. 12, pp. 5215–5228, 2012.

[4] J. Patterson and A. Gibson, *Deep Learning: A Practitioner's Approach*. .ºReilly Media, Inc.", 2017.

[5] A. Alvarez-Alvarez, D. Sanchez-Valdes, G. Trivino, Á. Sánchez, and P. D. Suárez, "Automatic linguistic report of traffic evolution in roads," *Expert Systems with Applications*, vol. 39, no. 12, pp. 11 293–11 302, 2012.

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

[7] R. R. Yager, "A new approach to the summarization of data," *Information Sciences*, vol. 28, no. 1, pp. 69–86, 1982.

[8] L. A. Zadeh, "A computational approach to fuzzy quantifiers in natural languages," *Computers & Mathematics with applications*, vol. 9, no. 1, pp. 149–184, 1983.

[9] A. van der Heide and G. Triviño, "Automatically generated linguistic summaries of energy consumption data," in *Intelligent Systems Design and Applications, 2009. ISDA'09. Ninth International Conference on*. IEEE, 2009, pp. 553–559.

[10] L. A. Zadeh, "Fuzzy logic= computing with words," *IEEE transactions on fuzzy systems*, vol. 4, no. 2, pp. 103–111, 1996.

Generación Automática de Explicaciones en Lenguaje Natural para Árboles de Decisión de Clasificación

B. López-Trigo, Jose M. Alonso, A. Bugarín
Centro Singular de Investigación en Tecnoloxías da Información (CiTIUS),
Universidade de Santiago de Compostela,
Campus Vida, E-15782, Santiago de Compostela, Spain
Email: bruno.lopez.trigo@rai.usc.es, {josemaria.alonso.moral, alberto.bugarin.diz}@usc.es

***Resumen*—En este trabajo describimos un modelo de explicaciones en lenguaje natural para árboles de decisión para clasificación. Las explicaciones incluyen aspectos globales del clasificador y aspectos locales de la clasificación de una instancia concreta. La propuesta está implementada en el servicio Web de código abierto ExpliClas [1], que en su versión actual opera sobre árboles construidos con Weka y conjuntos de datos con atributos numéricos. Ilustramos la viabilidad de la propuesta con dos casos de ejemplo, donde mostramos paso a paso cómo el modelo explica los respectivos árboles de clasificación.**

***Index Terms*—Explicabilidad, Soft Computing, Árboles de decisión para Clasificación, Generación de Lenguaje Natural**

I. Introducción

La generalización del uso de las nuevas tecnologías ha hecho que hoy trabajemos y vivamos rodeados de sistemas inteligentes [2]. Términos como ciudad inteligente, fábrica, casa, coche o teléfono inteligentes, son cada vez más populares. En realidad, existen multitud de dispositivos dotados de cierta inteligencia que nos asisten en el día a día, muchas veces sin que seamos totalmente conscientes de ello. Mención especial merece el teléfono móvil, que nos ofrece multitud de aplicaciones casi para cualquier cosa que podamos imaginar y va con nosotros a todas partes. Se puede afirmar que, si bien en el pasado vivimos una revolución industrial, ahora estamos viviendo una revolución social impulsada por la Inteligencia Artificial (IA).

Cuando un sistema inteligente toma decisiones que nos afectan (ej. filtrar llamadas, diagnóstico médico, concesión de un préstamo, etc.), surgen multitud de preguntas que deberíamos hacernos [3]: ¿quién es el responsable de las consecuencias colaterales que pudieran derivarse de las decisiones tomadas? ¿cuáles son las consecuencias éticas? ¿puede haber consecuencias legales?

Desde el punto de vista legal, el Parlamento Europeo aprobó una nueva Regulación General de Protección de Datos [4] que entró en vigor el 25 de mayo de 2018. La nueva regulación enfatiza el derecho de los ciudadanos a pedir explicaciones, independientemente de que las decisiones que les afectan sean tomadas por una persona o un programa informático. Esto significa que los ciudadanos pueden pedir a las empresas que les den explicaciones asociadas a las decisiones tomadas por los sistemas inteligentes que utilizan.

Desde un punto de vista técnico: ¿puede explicarnos la aplicación que tomó una decisión por qué tomó esa decisión y no otra? Para esto, hay básicamente dos opciones [5]: (1) el sistema inteligente está construido siguiendo un modelo interpretable (también llamado de caja blanca) que un operario experto puede analizar y entender a fin de elaborar una explicación; o (2) el sistema está construido siguiendo un modelo explicable que genera explicaciones por sí mismo. La DARPA planteó en 2016 las siguientes cuestiones técnicas [5]: ¿puede una máquina inteligente aprender de forma autónoma a explicar su comportamiento? ¿está preparada la generación actual de sistemas inteligentes para dar explicaciones de forma clara, sin ambigüedades, tanto a públicos especializados como no especializados? Y lanzó el reto de crear una nueva generación de sistemas inteligentes explicables entre 2017 y 2021. El reto fue lanzado inicialmente a universidades y centros de investigación americanos, con énfasis en la creación de equipos multidisciplinares que abordasen no sólo aspectos algorítmicos sino también de implementación y evaluación con personas. Los equipos seleccionados empezaron a trabajar en mayo de 2017 pero a día de hoy sólo hemos encontrado resultados muy preliminares (ej. [6], [7]).

Hasta donde nosotros sabemos, en la práctica, la responsabilidad de generar explicaciones recae directamente en el operario asociado al sistema inteligente, si está disponible para ello [8]. Aunque hay sistemas basados en conocimiento que son interpretables, en los últimos años son cada vez más populares las técnicas de IA para aprendizaje automático y minería de datos, supervisadas y no supervisadas (es decir, con o sin intervención humana). Estos sistemas se están demostrando ciertamente útiles y versátiles, pero la mayoría no suelen tener ninguna capacidad explicativa ni tampoco pueden ser interpretados fácilmente por personas (en cuyo caso se dice que son sistemas de caja negra).

Por tanto, el nuevo marco legal demanda que los expertos en IA desarrollen nuevos algoritmos que proporcionen explicaciones de forma automática.

En este trabajo, presentamos un modelo para la interpre-

tación de uno de los algoritmos de IA más interpretable, como son los árboles de decisión para clasificación, que introduciremos en la Sección II. El generador de explicaciones basado en dicho modelo y la combinación de técnicas de análisis inteligente de datos y generación de lenguaje natural se describe en la Sección III. La Sección IV presenta 2 casos de uso ilustrativos. Finalmente, la Sección V resume las principales conclusiones y apunta líneas de trabajo futuro.

II. Clasificación con Árboles de Decisión

Dentro del aprendizaje supervisado a partir de conjuntos de datos, los métodos basados en modelos se caracterizan por representar el conocimiento aprendido en algún formalismo de representación que explicita dicho conocimiento. Una ventaja importante de esta aproximación es que, una vez que se dispone del modelo, éste puede aplicarse directamente sobre nuevas instancias (por ejemplo, en problemas de predicción, como la clasificación) sin necesidad de seguir manteniendo los datos de entrenamiento. Los árboles de decisión utilizan como formalismo de representación un árbol donde los nodos representan condiciones sobre los valores de los atributos del conjunto de datos, que se organizan jerárquicamente, y donde las ramas de cada nodo corresponden a posibles valores del atributo. Hay diferentes métodos inductivos [9], [10] para la construcción de un árbol de decisión, pero todos ellos suelen utilizar estrategias "divide y vencerás" que construyen el árbol desde la raíz a las hojas seleccionando en cada nodo intermedio el atributo y la condición que particiona el conjunto de datos de la mejor manera posible, habitualmente en base a criterios de entropía y de maximización de la ganancia de información [11].

En el caso concreto de los árboles de clasificación, los nodos hojas contienen, idealmente, un conjunto de instancias correspondientes a la misma clase. La aplicación para la clasificación de nuevas instancias se inicia evaluando la condición del nodo raíz para los atributos de dicha instancia y continuando el recorrido por las ramas y nodos correspondientes. El proceso de clasificación finaliza cuando se alcanza un nodo hoja, que indica la clase que corresponde a la instancia. En la práctica, la condición de que un nodo hoja contenga únicamente instancias de la misma clase (nodo "puro") es demasiado restrictiva, con lo que dicha condición se debe relajar dentro de unos márgenes de pureza. Ello da lugar, por otra parte, a que los árboles clasifiquen incorrectamente algunos (idealmente muy pocos) casos, característica que se recoge en la matriz de confusión entre clases.

Nuestro modelo de explicación de árboles de clasificación se basa en estos aspectos que acabamos de comentar. Por un lado, una caracterización global del problema de clasificación y del árbol inducido. Por otro, una explicación del recorrido por el árbol en la tarea de clasificación. Veremos en la siguiente sección estos aspectos en mayor detalle.

III. Modelo para la Generación de Explicaciones

La generación de texto en Lenguaje Natural (popularmente conocida como NLG por el acrónimo de "Natural Language Generation") constituye una línea de investigación destacada en el área de la IA y la Lingüística Computacional [12].

En este trabajo, tomamos como punto de partida la arquitectura NLG más popular, inicialmente propuesta por Reiter y Dale [13], y la Teoría Computacional de Percepciones propuesta por Zadeh [14]. La generación de explicaciones en Lenguaje Natural se hace combinando plantillas y librerías de código abierto para la realización lingüística [15].

Planteamos la explicación de clasificadores mediante árboles de decisión a dos niveles (global y local), tal y como se describe a continuación. Todos los ejemplos utilizados en las siguientes secciones para ilustrar la propuesta se pueden reproducir mediante el servicio web ExpliClas [1].

III-A. Explicación global de un clasificador

El primer nivel es la explicación que denominamos global, que se orienta a describir el comportamiento general de un árbol de clasificación dado, aprendido a partir de un determinado conjunto de datos. La información que se incluye en la explicación global se refiere esencialmente a características del propio problema de clasificación y su rendimiento. Los datos de entrada para esta explicación provienen del propio conjunto de datos y de la matriz de confusión del clasificador aprendido.

La planificación de la explicación global contiene los elementos que se muestran a continuación:

- **Contextualización del problema**, que enumera las clases del mismo.

  Prototipo: `There are [N] types of beer: [Class1], [Class2], ... and [ClassN].`

  Ejemplo: `There are 8 types of beer: Blanche, Lager, Pilsner, IPA, Stout, Barleywine, Porter and Belgian Strong Ale.`

- **Fiabilidad del clasificador**, que evalúa el porcentaje global de clasificaciones correctas sobre el conjunto de datos de aprendizaje, incluyendo una valoración cualitativa del mismo de acuerdo con una definición establecida de valores lingísticos.

  Prototipo: `This classifier is [very reliable / quite confusing / very confusing] because correctly classified instances represent [percentage]%.`

  Ejemplo: `This classifier is very reliable because correctly classified instances represent 94.75%.`

- **Confusión del clasificador**, destacando qué clases se ven afectadas en mayor medida por dicha confusión. Aquí se interpreta la matriz de confusión del clasificador como una matriz de adyacencia de un grafo, cuyos ciclos se entienden como posibles caminos cerrados de confusión entre clases. Se toma el camino de mayor

longitud para ser incluido en la explicación. En caso de que el nivel de confusión sea bajo se omitirá esta parte de la explicación. A la hora de enumerar las clases se busca limitar la longitud de la explicación, tratando de forma diferente los casos en que el camino cerrado de confusión es largo (muchas clases confundidas) o corto (número reducido de clases confundidas) de modo que la longitud de la explicación sea lo más corta posible. Así, en el primer caso, se enumeran las clases para las que no hay confusión (expresándolas como excepciones) y en el segundo caso se enumeran las clases para las que hay confusión. En situaciones intermedias, como la del siguiente ejemplo, se citan los casos concretos.

Prototipo: `There may be some confusion among samples related to [a few / some / most / all] types of [object]. But among all of them [the pair / pairs [[class1]; [class2]] and [[classM-1]; [classM]] [is / are] the most confused.`

Ejemplo: `There may be some confusion among samples related to some types of beer. But among all of them the pair [IPA; Barleywine] is the most confused.`

- **Confusión elevada entre clases**, donde se destacan aquellos pares de clases que presenten un elevado nivel de confusión y no estén incluidas en los ciclos anteriores. Al igual que en los ejemplos mostrados previamente, se incluye una valoración lingüística además de la numérica.

  Prototipo: `[On the one hand / On the other hand], the following pairs are [eventually / often / usually] misleaded: Class [class1] is confused with class [class2] in [percentage]% of cases.`

  Ejemplo: `On the one hand, the following pairs are eventually misleaded: class headlamps is confused with class build wind float in 10.34% of cases.`

III-B. Explicación local de una instancia

El segundo nivel es la explicación local, que se orienta a explicar cuál es el resultado de la clasificación obtenida al aplicar el clasificador sobre una nueva instancia. La información que se incluye en la explicación local se refiere al recorrido por el árbol de clasificación desde la raíz hasta una hoja, determinado por las condiciones que se cumplen en los diferentes nodos del árbol para la instancia que se quiere clasificar. La versión actual del modelo que hemos definido para la generación de explicaciones en lenguaje natural, se aplica únicamente a atributos de tipo numérico, lo que nos permite dar una cierta flexibilidad en la explicación, para considerar posibles alternativas a la clasificación real. Para ello incluimos una cierta tolerancia en cuanto a los valores umbral de las condiciones, para de este modo contemplar que se puedan dar pequeñas variaciones en el valor de un atributo, que pudieran derivar en una clasificación diferente. Los datos de entrada para la explicación local son la instancia a clasificar, el árbol de clasificación y el valor de tolerancia permitido (por defecto, 5 % sobre el valor de cada atributo).

La planificación de la explicación local contiene los siguientes elementos:

- **Descripción de la clase**, donde se expresa cuál es el resultado de la clasificación y un resumen lingüístico de los valores de los atributos que han dado lugar a dicha clasificación. En el resumen se incluyen, para cada atributo X expresiones del tipo "X es A", donde A es un valor lingüístico predefinido.

  Prototipo: `[Object] is type [output class] because its [attribute1] is [lingTerm1Attribute1] ([or [lingTerm2Attribute1]]), its [attribute2] and [attribute3] are [lingTerm1Atributo1...], ... and its [attributeN] is [lingTerm1AttributeN].`

  Ejemplo: `Beer is type Porter because its strength is standard and its color is brown.`

- **Explicaciones alternativas**, que se construyen en base al umbral de tolerancia mencionado anteriormente. Se ha establecido un margen de tolerancia del 5 % para cada una de las condiciones nodo que justifican la clasificación, de modo que se exploran y se incluyen en la explicación las posibles clasificaciones alternativas que se obtendrían en caso de que los valores de los atributos cumpliesen las condiciones dentro del margen de tolerancia.

  Prototipo: `However, this [object] may be also [alternativeClass1] because its [alternativeAttribute1] is quite close to the split value ([thresholdValue]). For these specific values, it is [unlikely / quite likely / just as likely] to be [alternativeClass1].`

  Ejemplo: `However, this beer may be also Stout because its color is quite close to the split value (30.45). For these specific values, it is just as likely to be Stout.`

Por último también se incorporan en la explicación alternativa aquellas clases para las cuales hay un elevado nivel de confusión en general con la clase original. Para ello se tiene en cuenta la matriz de confusión en lo que respecta a las clases implicadas, adoptando por tanto una cierta perspectiva global. Así, si las clases tienen, en general, un

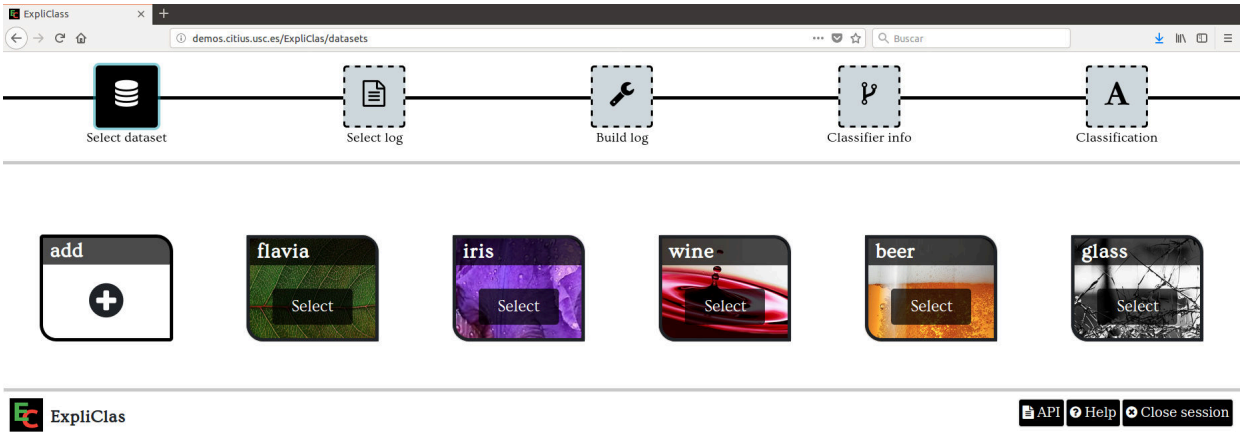


Figura 1. Página de inicio del Servicio Web ExpliClas [1].

elevado nivel de confusión, la explicación enfatiza este aspecto, mientras que si el nivel de confusión es bajo se presentará como un caso de cierta excepción. Mostramos a continuación un ejemplo de esta última situación:

Prototipo: `But [alternativeClass1] will be an exception because class [outputClass] is confused with [alternativeClass1] only in [percentage]% of cases.`

Ejemplo: `But Stout will be an exception because class Porter is confused with Stout only in 2% of cases.`

IV. Algunos Casos de Uso

Una vez descritos los elementos que componen cada explicación, veremos en esta sección dos ejemplos completos, con los que ilustraremos el funcionamiento de nuestra propuesta paso a paso. En ambos casos se aprenden clasificadores utilizando el algoritmo C4.5 [10], en la implementación disponible en Weka (J48) [16], [17]. Tanto los dos ejemplos mostrado (IRIS y FLAVIA), como otros disponibles, se pueden reproducir con el servicio Web ExpliClas [1] (Fig. 1).

IV-A. Conjunto de datos IRIS

El conjunto de datos IRIS (uno de los más conocidos del repositorio [18]) está formado por 150 instancias, 4 atributos numéricos y 3 clases. El árbol de clasificación generado por Weka (Fig. 2) está formado por 9 nodos totales, 5 de ellos nodos-hoja que deciden la clasificación y los 4 nodos restantes con las condiciones (comparaciones sobre los valores de los atributos) para decidir la clasificación. Se trata, por tanto, de un árbol simple que utilizaremos como primer ejemplo.

La explicación global generada en este caso es la siguiente:

```
There are 3 types of iris:
Setosa, Virginica and Versicolor.
This classifier is very reliable
because correctly classified
instances represent 96%.
```

```
Petal-Width <= 0.6: 1.0 (50.0)
Petal-Width > 0.6
|   Petal-Width <= 1.7
|   |   Petal-Length <= 4.9: 2.0 (48.0/1.0)
|   |   Petal-Length > 4.9
|   |   |   Petal-Width <= 1.5: 3.0 (3.0)
|   |   |   Petal-Width > 1.5: 2.0 (3.0/1.0)
|   Petal-Width > 1.7: 3.0 (46.0/1.0)
```

Figura 2. Árbol de clasificación correspondiente al conjunto de datos IRIS (captura de pantalla de Weka [17]).

La explicación local, para la instancia de la Fig. 3 (Sepal-Length: 5.6, Sepal-Width: 3, Petal-Length: 4.1, Petal-Width: 1.3) es la siguiente:

```
Iris is type Virginica because
its petal-length and petal-width
are medium.
```

En este caso, la explicación consiste en indicar los valores lingüísticos correspondientes a los valores numéricos de los atributos que han dado lugar a la clasificación, tal y como se detalla en la figura.

Sin embargo, si tomamos una instancia cuyos valores sean precisamente los de umbrales de los nodos intermedios (Sepal-Length: 5.6, Sepal-Width: 3, **Petal-Length: 4.9**, **Petal-Width: 0.6**), la explicación resulta más extensa:

```
Iris is type Setosa because its
petal-width is low.
However, this iris may be also
Virginica because its petal-width
is quite close to the split value
(0.6).
It may be also Versicolor because
its petal-width and petal-length
are quite close to the split values
(0.6 and 4.9, respectively). For
these specific values it is just
as likely to be Virginica and
```

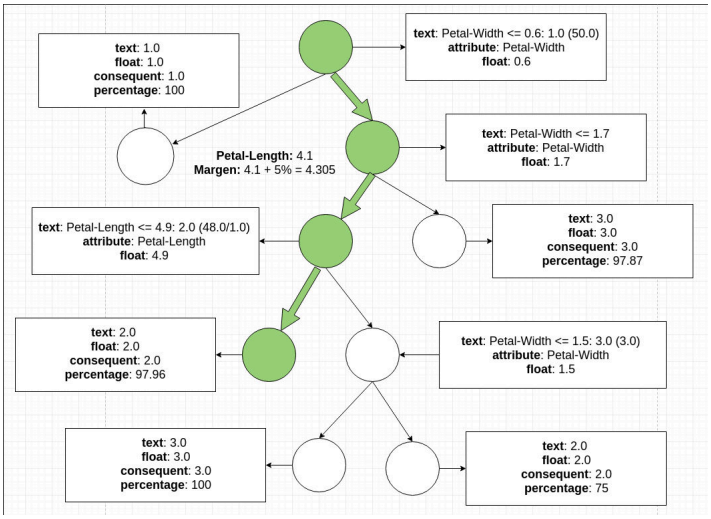


Figura 3. Clasificación de la instancia Sepal-Length: 5.6, Sepal-Width: 3, Petal-Length: 4.1, Petal-Width: 1.3.

```
Versicolor. But Virginica and
Versicolor will be an exception
because class Setosa is confused
with Virginica and Versicolor
only in 2% and 0% of cases,
respectively.
```

En este caso, la clasificación realizada es como clase Setosa. Sin embargo, al ser los valores de la instancia idénticos a los umbrales, y entrar en el rango de la tolerancia establecida del 5 %, se consideran como alternativas las dos ramas del nodo raíz y las del nodo que clasifica por longitud. Todas estas alternativas conducen a las clases Virginica y Versicolor. En ambos casos se indica el valor umbral que lo justifica y se valora la situación como que podría ser indistintamente tanto una como otra. Sin embargo, se introduce un matiz de carácter global, puesto que de acuerdo con la matriz de confusión del clasificador, la confusión de la clase Setosa con las clases Virginica y Versicolor es muy poco frecuente:

$$\begin{pmatrix} & Set. & Virg. & Vers. \\ Set. & 49 & 1 & 0 \\ Virg. & 0 & 47 & 3 \\ Vers. & 0 & 2 & 48 \end{pmatrix}$$

IV-B. Conjunto de datos FLAVIA

En esta sección discutimos un caso más realista. FLAVIA[1] es un proyecto de código abierto en el que se abordó la creación de un conjunto de datos para la clasificación automática de hojas en la región de Yangtze Delta (próxima a Shanghai) en China. El conjunto de datos está formado por 1800 muestras de hojas (15 atributos) que corresponden a 32 clases diferentes. Una red neuronal es capaz de clasificar todas las hojas con una tasa de acierto superior al 90 % [19]. Sin embargo, la clasificación se basa en un modelo de caja negra que una persona no puede entender. El árbol construido por el algoritmo J48 de Weka contiene 449 nodos (225 nodos-hoja) y una tasa de acierto de clasificación del 70.44 % (considerando *10-fold cross-validation*). Se puede apreciar cómo pasar de un modelo de caja negra a un modelo de caja blanca supone en este caso una reducción apreciable en precisión. Además, aunque el modelo generado es de caja blanca, el elevado número de clases, atributos y nodos hace que la interpretación no sea sencilla, incluso para un experto en botánica.

En [20], presentamos los resultados de una encuesta en la que demostramos la utilidad de generar explicaciones en lenguaje natural asociadas a clasificaciones hechas por un conjunto de reglas borrosas aprendidas sobre un subconjunto de los datos de FLAVIA, con 310 instancias, 3 atributos y 5 clases (Fig. 4). De los 15 atributos de partida (que caracterizan propiedades geométricas y morfológicas) seleccionamos sólo los tres (Área, Perímetro y Diámetro) que un experto en botánica consideró útiles a fin de explicar en lenguaje natural el proceso de clasificación; prestando atención únicamente a la forma de la hoja. En esta sección, consideramos el mismo conjunto de datos usado en [20]. La explicación global es la siguiente:

```
There are 5 types of flavia:
Aesculus chinensis, Berberis
anhweiensis, Cercis chinensis,
Phoebe zhennan and Lagerstroemia
indica.
This classifier is very reliable
because correctly classified
instances represent 90.97%. There
may be some confusion among samples
related to some types of flavia.
But among all of them the pair
[Cercis chinensis; Phoebe zhennan]
is the most confused.
```

La matriz de confusión correspondiente (ordenada según las 5 clases de hoja en la Fig. 4) es:

$$\begin{pmatrix} 60 & 1 & 2 & 0 & 0 \\ 0 & 55 & 2 & 1 & 0 \\ 3 & 0 & 63 & 6 & 0 \\ 1 & 0 & 5 & 52 & 2 \\ 0 & 0 & 1 & 4 & 52 \end{pmatrix}$$

El árbol construido en este caso contiene sólo 25 nodos (13 nodos-hoja) y una tasa de acierto de clasificación del 90.97 % (considerando *10-fold cross-validation*).

ExpliClas permite introducir a mano el valor numérico de los atributos cuando el objeto a clasificar no coincide con ninguna de las instancias en el conjunto de datos. Por ejemplo, la explicación local para la hoja en la Fig. 5 sería la siguiente. Nótese que una sencilla comparativa visual entre las figuras 4 y 5 permite verificar cualitativamente la explicación dada.

```
Flavia is type Cercis chinensis
because its area is not very
small and its perimeter is small.
However, this flavia may be
```

[1] http://flavia.sourceforge.net/

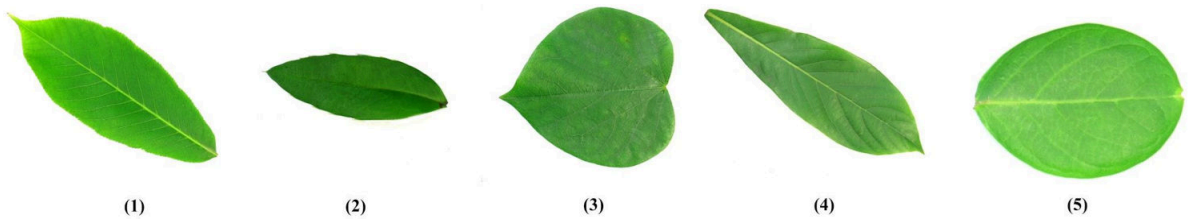


Figura 4. Las 5 clases en la versión reducicada de FLAVIA: (1) Aesculus chinensis, (2) Berberis anhweiensis, (3) Cercis chinensis, (4) Phoebe zhennan, (5) Lagerstroemia indica.

Figura 5. Ejemplo de hoja a clasificar (Área: 349.045, Perímetro: 2.964,304, Diámetro: 666,647).

```
also Phoebe zhennan because its
perimeter is quite close to the
split value (3,042.19). For these
specific values it is just as
likely to be Phoebe zhennan. But
Phoebe zhennan will be an exception
because class Cercis chinensis is
confused with Phoebe zhennan in
8.33% of cases.
```

V. Conclusiones y Trabajo Futuro

En este trabajo hemos presentado un modelo para la generación de explicaciones (globales y locales) en lenguaje natural sobre clasificaciones hechas con árboles de decisión con atributos numéricos. El modelo está implementado en el servicio web ExpliClas [1]. Como trabajo futuro, realizaremos una validación exhaustiva del modelo con usuarios reales y refinaremos las explicaciones según la realimentación recibida. Adicionalmente, extenderemos el modelo de explicación para considerar atributos categóricos y algoritmos de clasificación de caja gris, como árboles de decisión borrosos, entre otros.

Agradecimientos

Jose M. Alonso es Investigador Ramón y Cajal (RYC-2016-19802). Este trabajo está financiado por los proyectos TIN2017-90773-REDT (iGLN), TIN2017-84796-C2-1-R (BIGBISC), TIN2014-56633-C3-1-R (BAI4SOW) y TIN2014-56633-C3-3-R (ABS4SOW) (Ministerio de Economía y Competitividad) y GRC2014/030 y "Acreditación 2016-2019, ED431G/08" (Xunta de Galicia), todos con cofinanciación FEDER.

Referencias

[1] B. López-Trigo, J. M. Alonso, and A. Bugarín, "ExpliClas: Web service for the automatic explanation in natural language of classification models in data mining," 2018, http://demos.citius.usc.es/ExpliClas/.

[2] K. Panetta, "Gartner top 10 strategic technology trends for 2018," 2017, https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2018/.

[3] S. Barocas and D. Boyd, "Computing ethics. engaging the ethics of data science in practice," *Communications of the ACM*, vol. 60, no. 11, pp. 23–25, 2017.

[4] Parliament and Council of the European Union, "General data protection regulation (GDPR)," 2016, http://data.europa.eu/eli/reg/2016/679/oj.

[5] D. Gunning, "Explainable Artificial Intelligence (XAI)," Defense Advanced Research Projects Agency (DARPA), Arlington, USA, Tech. Rep., 2016, DARPA-BAA-16-53.

[6] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? explaining the predictions of any classifier," in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, San Francisco, USA, 2016, pp. 1–10.

[7] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell, "Generating visual explanations," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Amsterdam, The Netherlands, 2016, pp. 3–19.

[8] K. Darlington, "Explainable AI systems: Understanding the decisions of the machines," 2017, openMind, BBVA Group, https://www.bbvaopenmind.com/en/explainable-ai-systems-understanding-the-decisions-of-the-machines/.

[9] J. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[10] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[11] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classication and Regression Trees*, 1st ed. Wadsworth, 1984.

[12] A. Gatt and E. Krahmer, "Survey of the state of the art in natural language generation: Core tasks, applications and evaluation," *Journal of Artificial Intelligence Research*, vol. 61, pp. 65–170, 2018.

[13] E. Reiter and R. Dale, *Building natural language generation systems*. Cambridge University Press, 2000.

[14] L. A. Zadeh, "A new direction in AI: Toward a computational theory of perceptions," *Artificial Intelligent Magazine*, vol. 22, no. 1, pp. 73–84, 2001.

[15] A. Gatt and E. Reiter, "SimpleNLG: a realisation engine for practical applications," in *Proceedings of the European Workshop on Natural Language Generation (ENLG)*, Athens, Greece, 2009, pp. 90–93.

[16] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, 4th ed. Morgan Kaufmann, 2016.

[17] The University of Waikato, "Weka 3: Data Mining Software in Java," 2018, https://www.cs.waikato.ac.nz/ml/weka/.

[18] The University of California at Irvine, "UCI machine learning repository," 2018, https://archive.ics.uci.edu/ml.

[19] S. Gang Wu, F. Sheng Bao, E. You Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang, "A leaf recognition algorithm for plant classification using probabilistic neural network," in *IEEE International Symposium on Signal Processing and Information Technology*, 2007, pp. 1–6.

[20] J. M. Alonso, A. Ramos-Soto, E. Reiter, and K. van Deemter, "An exploratory study on the benefits of using natural language for explaining fuzzy rule-based systems," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Naples, Italy, 2017, pp. 1–6, http://dx.doi.org/10.1109/FUZZ-IEEE.2017.8015489.

Descripción de series de tiempo utilizando Fuzzy Piecewise Linear Segments

Juan Moreno-Garcia, Antonio Moreno-Garcia
Universidad de Castilla-La Mancha
Escuela de Ingeniería Industrial
Toledo, España
juan.moreno@uclm.es, antmorgarcia@gmail.com

Luis Jimenez-Linares, Luis Rodriguez-Benitez
Universidad de Castilla-La Mancha
Escuela Superior de Informática
Ciudad Real, España
luis.jimenez@uclm.es, luis.rodriguez@uclm.es

***Abstract*—Es muy frecuente la utilización de series temporales en gran cantidad de ámbitos, siendo necesario la obtención de información lo más detallada posible a partir de estas series. Hay diferentes posibilidades de mostrar esta información, por ejemplo, en forma de representación gráfica. Aunque cada día es más frecuente la necesidad de representar información utilizando el lenguaje natural, es decir, mediante una descripción lingüística. En este trabajo se presenta una técnica para obtener descripciones lingüísticas a partir de series temporales utilizando una representación denominada Fuzzy Piecewise Linear Segments. Se detalla la forma de obtener la información de una serie modelada utilizando esta representación y los pasos necesarios para generar la descripción utilizando plantillas. Finalmente se muestra algunos ejemplos de su uso.**

***Index Terms*—Descripción Lingüística, Series de Tiempo, Fuzzy Piecewise Linear Segments, Lógica Difusa**

I. Introduction

En un gran número de aplicaciones se necesitan las series de tiempo. Usualmente estas series se representan en forma de datos en bruto. Este formato tiene varios problemas siendo el más importante la gran cantidad de memoria necesaria para su almacenamiento. Además, esta representación presenta el problema añadido de que su procesamiento es costoso en tiempo. Por esta razón se han desarrollado otras formas alternativas para almacenar las series de tiempo, lo que permite reducir el consumo de memoria y la ejecución de operaciones más eficientemente. Una de las técnicas más utilizadas se denomina "segmentos lineales a trozos" (Piecewise Linear Segment – PLS) que consiste en la representación de las series utilizando un conjunto de segmentos, donde cada uno de ellos corresponde a un trozo de la serie. Hay diferentes métodos para obtener esta representación en la literatura [1]–[3]. Además, cada día es más frecuente la generación de informes de datos utilizando el lenguaje natural. Esto se conoce como Descripción Lingüística de Datos [4]. Es una línea de investigación que se puede considerar clásica pero que actualmente está teniendo un fuerte auge.

El principal objetivo de este trabajo consiste en diseñar un nuevo método de generación de descripciones lingüísticas de series de tiempo. Muchos de los métodos presentados en la literatura hacen uso de la lógica difusa para obtener la descripción [4]. Por todo ello, se propone el uso de una representación de series que utiliza la lógica difusa [5] para generar descripciones lingüísticas directamente desde dicha representación. Esta representación se ha denominado *Fuzzy Piecewise Linear Segments* (FPLS) y tiene la ventaja de que recoge la imprecisión creada en el proceso de generación de los segmentos. Cada segmento de PLS se convierte al dominio difuso utilizando las técnicas detalladas en [5].

El documento está estructurado de la siguiente forma. En la Sección II se presenta una breve recopilación de los trabajos más destacados en este ámbito de investigación. En la Sección III se expone la representación FPLS. En la Sección IV se detalla la forma de obtener información de un FPLS y cómo generar las descripciones en base a ésta. Finalmente, la Sección V muestra las conclusiones y trabajos futuros.

II. Estado del arte

La descripción de series de tiempo (TS) es un campo de investigación con un gran número de publicaciones en los últimos 10 años. Recientemente Marín and Sánchez [4] han publicado un trabajo que recopila las publicaciones más destacadas en la literatura. Estos autores distinguen entre Generación de Lenguaje Natural (Natural Language Generation – NLG) y la Generación de Descripciones Lingüísticas de TS (Generation of Linguistic Descriptions of Time Series – GLiDTS). Se podría afirmar que este trabajo está relacionado con GLiDTS.

En general, la lógica difusa (Fuzzy Logic – FL) es un componente esencial en este tipo de sistemas y ha sido aplicada de diferentes formas. Se presentarán algunos trabajos destacados que utilizan la FL para describir TS y otros que funcionan junto con otras tecnologías. Por ejemplo, algunas propuestas combinan sistemas OLAP [6], diseñados para su utilización en la toma de decisión y aplicados en una amplia variedad de dominios de aplicación, con la FL. En esta línea se encuentra una nueva aproximación que hace uso de particiones jerárquicas difusas del tiempo y la evaluación de sentencias cuantificadas [7], [8]. La descripción final consiste en una colección de este tipo de sentencias. Otro ejemplo es *GALiWeather* [9] que mezcla técnicas de computación de percepciones con estrategias para la descripción lingüística

Supported by the project TIN2015-64776-C3-3-R of the Science and Innovation Ministry of Spain, co-funded by the European Regional Development Fund (ERDF).

de datos junto con un sistema de NLG. El sistema ofrece información sobre el tiempo que es utilizada por la Agencia de Meteorología Gallega. Actualmente, *GALiWeather* es un servicio público ofrecido para la predicción del tiempo.

Otra posibilidad consiste en generar un modelo difuso a partir de la TS y calcular una estructura de alto nivel. Por ejemplo, en [10], [11] se presentó una estructura para modelar los eventos que ocurren en una TS. El modelo final contiene los mínimos y máximos utilizando [11]. Esta estructura también permite la búsqueda de eventos: el resultado obtenido de esta búsqueda genera la descripción lingüística final que está formada por sentencias de texto que son añadidas cuando se identifica un nuevo evento. El sistema necesita un experto en el campo de aplicación.

Otras propuestas consisten en transformar la TS en otra representación y entonces fuzzificarla. Por ejemplo, Kacprzyk et al. [12] propusieron el uso de tendencias identificando segmentos lineales de una TS. Posteriormente representan la serie mediante un conjunto de atributos que caracterizan las tendencias (la pendiente del segmento, la calidad de la aproximación y la longitud de la tendencia). El campo de aplicación que seleccionaron fue la evaluación de un fondo de inversión en un período de tiempo. En otras situaciones la información proviene de diversas fuentes y debe ser agregada apropiadamente obteniendo una nueva representación que se trata utilizando FL [13].

Otras investigaciones crean modelos específicos para generar descripciones lingüísticas. Granular Linguistic Model of Phenomena (GLMP) puede ser clasificado dentro de esta categoría. Alvarez-Alvarez y Triviño [14] introdujeron la aplicación de GLMP para generar descripciones de la calidad de la marcha humana y Sánchez-Valdés y Triviño mejoraron los resultados utilizando una máquina de estados finitos difusa [15]. GLMP ha sido utilizado en aplicaciones de tipo muy diverso.

III. Descripción de FPLS

Una FPLS es una conjunto de segmentos que han sido fuzzificados a partir de los segmentos de una PLS. Para un instante dado ofrecen un número difuso que es el valor de salida. Formalmente, una FPLS está compuesta por un conjunto de segmentos lineales difusos que serán representados como $fpls_{t_i,t_{i+1}}$. La Ecuación 1 representa formalmente a una FPLS.

$$FPLS(T) = \{fpls_{t_0,t_1}, fpls_{t_1,t_2}, \ldots, fpls_{t_{|FPLS|-1},t_{|FPLS|}}\} \quad (1)$$

donde cada segmento $fpls_{t_i,t_{i+1}} = \{m_{t_i,t_{i+1}}, c_{t_i,t_{i+1}}, p_{t_i,t_{i+1}}\}$ siendo $m_{t_i,t_{i+1}}$ y $c_{t_i,t_{i+1}}$ la pendiente y la constante de la recta que define el segmento respectivamente, y $p_{t_i,t_{i+1}}$ el promedio de la tasa de error (Ecuación 2).

$$p_{t_i,t_{i+1}} = \frac{\sum_{k=i}^{i+1} \frac{|fpls_{t_i,t_{i+1}}(t_k) - y_k|}{y_k}}{t_{i+1} - t_i + 1} \quad (2)$$

donde t_i y t_{i+1} son los instantes de comienzo y de fin del segmento, $fpls_{t_i,t_{i+1}}(t_k)$ es el valor del segmento $fpls_{t_i,t_{i+1}} \in FPLS$ en el instante t_k e y_k es el valor de la serie Y en el instante t_k.

El promedio de la tasa de error es una medida que calcula la media de la tasa del error para cada segmento utilizando la Ecuación 2.

Cada segmento lineal difuso $fpls_{t_i,t_{i+1}}$ devuelve un número difuso triangular fn_k para un instante $t_k \in \mathbb{R}$: $fn_k = fpls_{t_i,t_{i+1}}(t_k)$. Utilizando estos tres valores, el número difuso de salida fn_k se puede calcular tomando como entrada un valor t_k. La Ecuación 3 muestra la forma en que un segmento $fpls_{t_i,t_{i+1}}$ calcula el número difuso fn_k.

$$fn_k = \begin{cases} 0 & if \quad t_k < t_i \\ calcular(fpls_{t_i,t_{i+1}}, t_k) & if \quad t_i \leq t_k \leq t_{i+1} \\ 0 & if \quad t_{i+1} < t_k \end{cases} \quad (3)$$

donde $calcular(fpls_{t_i,t_{i+1}}, t_k)$ es una función que calcula fn_k^{DOWN}, fn_k^{fpls} y fn_k^{UP} que componen el número difuso triangular $fn_k = \{fn_k^{DOWN}, fn_k^{fpls}, fn_k^{UP}\}$.

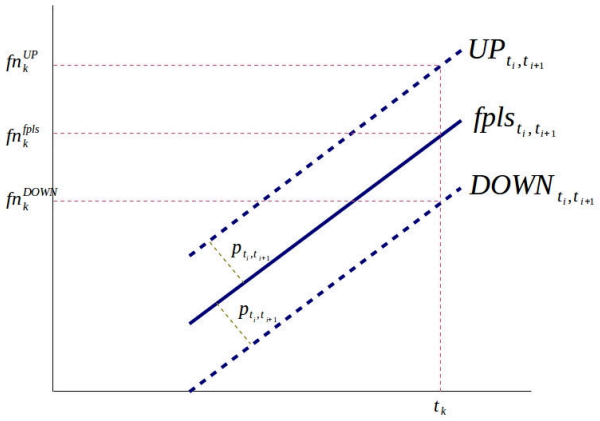


Fig. 1. Procedimiento para el cálculo de fn_k a partir de un segmento difuso.

Para los valores de t_k menores que t_i o mayores que t_{i+1}, la salida es cero. Para valores entre t_i y t_{i+1} se calcula utilizando dos segmentos paralelos con respecto a $fpls_{t_i,t_{i+1}}$ llamados $UP_{t_i,t_{i+1}}$ y $DOWN_{t_i,t_{i+1}}$ (Figura 1). $p_{t_i,t_{i+1}}$ se necesita para calcular los segmentos $UP_{t_i,t_{i+1}}$ y $DOWN_{t_i,t_{i+1}}$, que están por encima y por debajo en el eje Y respecto a $fpls_{t_i,t_{i+1}}$, es decir:

- $UP_{t_i,t_{i+1}} = (m_{t_i,t_{i+1}} * x) + c_{t_i,t_{i+1}} + p_{t_i,t_{i+1}}$
- $DOWN_{t_i,t_{i+1}} = (m_{t_i,t_{i+1}} * x) + c_{t_i,t_{i+1}} - p_{t_i,t_{i+1}}$

Los valores fn_k^{DOWN}, fn_k^{fpls} y fn_k^{UP} que definen fn_k se calculan como el valor de salida de los segmentos lineales $DOWN_{t_i,t_{i+1}}$, $fpls_{t_i,t_{i+1}}$ y $UP_{t_i,t_{i+1}}$ para un instante t_k y en orden creciente, es decir:

- $fn_k^{DOWN} = DOWN_{t_i,t_{i+1}}(t_k)$
- $fn_k^{fpls} = fpls_{t_i,t_{i+1}}(t_k)$
- $fn_k^{UP} = UP_{t_i,t_{i+1}}(t_k)$

TABLE I
UN FPLS DE EJEMPLO. FUENTE MORENO-GARCIA ET AL. [5]

$fpls_{t_i,t_{i+1}}$	$m_{t_i,t_{i+1}}$	$c_{t_i,t_{i+1}}$	$p_{t_i,t_{i+1}}$
$fpls_{0.0,\ 17.0}$	-0.0432	0.6786	0.313
$fpls_{17.0,\ 49.0}$	0.0339	-0.5487	0.5544
$fpls_{49.0,\ 91.0}$	-0.0249	2.1815	0.4561
$fpls_{91.0,\ 94.0}$	-0.0007	0.1543	0.0905
$fpls_{94.0,\ 131.0}$	0.0229	-2.1022	0.4582

Estos valores representan un número difuso triangular simétrico (Symmetric Triangular Fuzzy number – STFN) con la etiqueta lingüística "aproximadamente fn_k" y una función de pertenencia que se muestra en la Ecuación 4.

$$\mu_{fn_k}(y) = \begin{cases} 0 & if \quad |fn_k^{fpls} - y| > fn_k^{UP} - fn_k^{fpls} \\ 1 & if \quad y = fn_k^{fpls} \\ \frac{|fn_k^{fpls} - y|}{fn_k^{UP} - fn_k^{fpls}} & if \quad |fn_k^{fpls} - y| < fn_k^{UP} - fn_k^{fpls} \end{cases} \tag{4}$$

Como el soporte de fn_k se calcula en base al promedio de la tasa de error para el segmento obtenido, el uso de la media es apropiado para "medir la incertidumbre", es decir, cuanto mayor es el error, mayor es el soporte del número difuso.

FPLS permite realizar diferentes operaciones. Actualmente están definidas la comparación entre dos FPLSs que representan dos TS o a una subsecuencia de una TS, que conceptual y prácticamente son lo mismo. La idea básica del funcionamiento del método consiste en realizar un conjunto de comparaciones a instantes de tiempo igualmente espaciados sobre las dos FPLS. Los números difusos obtenidos para cada FPLS como salida en cada instante pueden ser comparados mediante operaciones de la lógica difusa, y posteriormente recoger el resultado en un valor que agregue las comparaciones realizadas representando la similitud/disimilitud de ambas FPLS, y por tanto de ambas TS o subsecuencias, según el caso. Para comparar dos STFN se utilizó una medida que toma el valor del área comprendida entre ellos considerando el valor de los números difusos a comparar.

IV. Generación de las descripciones

De una FPLS se puede obtener información para generar las descripciones lingüísticas. Esta información se va a clasificar en dos niveles:

- De segmento: a este nivel se pueden generar descripciones lingüísticas que reflejan información sobre las tendencias, dado que un segmento representa una tendencia.
- De FPLS: en este caso se mostrará información de la TS completa o de una parte de ella ya que una FPLS representa una TS.

Primero se presentará la información que se puede obtener a nivel de una tendencia (segmento), más concretamente, se detallará la siguiente:

- Tipo de tendencia: se obtendrá una etiqueta lingüística por segmento que se ha denominado $type_{i+1}$ y que indica el tipo de tendencia obtenida para el segmento $i+1$. Estará en función de la pendiente del segmento que modela. La pendiente de una recta es mayor o menor que 0 si la recta es creciente o decreciente respectivamente. Se distinguirán tres tipos de segmento: descendente, plana y ascendente (Tabla II), aunque se puede redefinir este conjunto de etiquetas para incorporar más grados. $type_{i+1}$ toma de valor la etiqueta de máxima pertenencia de las de ese conjunto (Ecuación 5).

$$type_{i+1} = argmax_T \ \ \mu_T(m_{t_i,t_{i+1}}) \ \forall T \in TYPE \tag{5}$$

- Potencia de la tendencia: vendrá definida por el ángulo del segmento difuso ($arctan(m_{t_i,t_{i+1}})$) y para clasificarla se utilizará el conjunto de etiquetas lingüísticas $POWER$. La Ecuación 6 muestra la forma de obtener dicha etiqueta.

$$power_i = argmax_P \ \ \mu_P(arctan(m_{t_i,t_{i+1}})) \forall P \in POWER \tag{6}$$

$power_i$ se asigna a la etiqueta lingüística $P \in POWER$ que obtiene el máximo valor de pertenencia para el ángulo del segmento, es decir, la etiqueta que representa mejor el incremento o el decremento del segmento difuso. La Tabla III muestra un conjunto ejemplo que utiliza etiquetas difusas donde el soporte está considerado como el ángulo de la pendiente medido en grados sexagesimales.
- Duración: se definirá un conjunto de etiquetas llamado $LONG$ para representar la longitud del segmento. Cada segmento $fpls_{t_i,t_{i+1}}$ cuenta con sus instantes de inicio

TABLE II
CONJUNTO DE ETIQUETAS LINGÜÍSTICAS $TYPE$.

Label	a	b	c	d
descendente	$-\infty$	$-\infty$	-0.1	0.0
plana	-0.1	0.0	0.0	0.1
ascendente	0.0	0.1	∞	∞

TABLE III
CONJUNTO DE ETIQUETAS LINGÜÍSTICAS $POWER$.

Label	a	b	c	d
fuerte descenso	-90	-90	-45	-40
descenso	-45	-40	-2	0
llano	-2	0	0	2
ascenso	0	2	40	45
fuerte ascenso	40	45	90	90

TABLE IV
CONJUNTO DE ETIQUETAS LINGÜÍSTICAS $LONG$.

Label	a	b	c	d
muy corta	0	0	2.5	5
corta	2.5	5	15	20
un poco corta	15	20	30	35
media	30	35	45	50
larga	45	50	55	65
muy larga	55	65	∞	∞

TABLE V
Conjunto de etiquetas lingüísticas LOC.

Label	a	b	c	d
inicio	0	0	5	10
inicio pasado	5	10	35	40
centro	35	40	60	65
centro pasado	60	65	90	95
final	90	95	100	100

y fin (t_i y t_{i+1}) que permitirán calcular la duración mediante la Ecuación 7.

$$long_i = argmax_L \ \ \mu_L(t_{i+1} - t_i) \ \forall L \in LONG \quad (7)$$

Como puede verse se realiza de forma similar a los casos anteriores. Las etiquetas de $LONG$ permitirán indicar el tamaño de los segmentos. La Tabla IV muestra que será utilizado posteriormente.

- Localización: se trata de definir la localización de la tendencia en el tiempo. Para ello se utilizará el instante central de ocurrencia de la tendencia y se fuzzificará utilizando un conjunto de etiquetas lingüísticas denominado LOC con un soporte en $[0\%, 100\%]$ del total de la longitud de la TS descrita. La Ecuación 8 muestra la forma de seleccionar la etiqueta.

$$loc_i = argmax_L \ \ \mu_L\left(\frac{p_{cen}}{|TS|}\right) \ \forall L \in LOC \quad (8)$$

donde p_{cen} se calcula utilizando la Ecuación 9.

$$p_{cen} = \left(t_i + \frac{t_{i+1} - t_i}{2}\right) * 100 \quad (9)$$

La Tabla V muestra un ejemplo de este conjunto que será utilizado en los ejemplos.

El segundo nivel que se puede realizar la descripción es a nivel de la FPLS completa. Se puede detallar información general como por ejemplo el número de tendencias, la longitud media de cada una de ellas, número de mínimos y máximos locales, localización de éstos, etc. A continuación se detallará la forma de obtener la longitud media y la tendencia media de los segmentos de la TS (potencia media) y el cálculo de mínimos y máximos y su localización.

- Longitud media de los segmentos de la TS: se debe calcular el valor medio del segmento y se fuzzifica. Para calcular el valor medio se utiliza la Ecuación 10.

$$l_{med} = argmax_L \ \ \mu_L\left(\frac{|TS|}{|FPLS|}\right) \ \forall L \in LONG \quad (10)$$

donde la operación "| |" es la anchura del soporte de TS.

El valor obtenido ($\frac{|TS|}{|FPLS|}$) se fuzzificará utilizando el conjunto de etiquetas $LONG$ que define la longitud.

- Tendencia media de la TS: para obtener la potencia de la tendencia media se utilizará el Algoritmo 1. Éste defuzzifica la etiqueta que define la potencia de cada tendencia acumulando todos los valores de defuzzicación en v_{med} (Línea 3). Finalmente, se vuelve a fuzzificar el valor medio obtenido (Línea 5) utilizando el conjunto original de etiquetas u otro distinto (según se necesite) para la descripción final (Línea 6). Para realizar la defuzzificación hay diferentes propuestas en la bibliografía, algunas opciones interesantes se detallan en [16]. La opción utilizada en los ejemplos de este trabajo ha sido la media de máximos (Mean of Maximum – MoM).

Algorithm 1 Cálculo de longitud media de las tendencias

1: $v_{med} = 0.0$ {acumula la longitud}
2: **for** $i = 0$ **to** $|FPLS|$ **do**
3: $\quad v_{med} = v_{med} + defuzz(type_{i+1})$
4: **end for**
5: $v_{med} = \frac{v_{med}}{|FPLS|}$ {longitud media}
6: $type_{med} = argmax_{E_j} \ \mu_{E_j}(v_{med})$ {selecciona la etiqueta de máxima pertenencia}

- Mínimos y máximos y su localización: FPLS permite la localización de mínimos y máximos comprobando el tipo de tendencia para dos tendencias consecutivas. Si se verifica la Ecuación 11 o 12 se ha localizado un mínimo o un máximo respectivamente.

$$(type_i = descendente) \wedge (type_{i+1} = ascendente) \quad (11)$$

$$(type_i = ascendente) \wedge (type_{i+1} = descendente) \quad (12)$$

donde $descenso \in TYPE$ y $ascenso \in TYPE$.

La localización del máximo viene indicada por el instante final de $fpls_{t_i,t_{i+1}}$ o por el inicial de $fpls_{t_{i+1},t_{i+2}}$ (es el mismo instante, t_{i+1}). Dado que FPLS es una representación aproximada se puede realizar una fuzzificación de t_{i+1} en base a la localización de dicho instante dentro del soporte del tiempo (Ecuaciones 13 y 14).

$$min_k = \mu_{TIME}(t_{min}) \quad (13)$$

$$max_k = \mu_{TIME}(t_{max}) \quad (14)$$

donde t_{min} y t_{max} son los instantes donde se ha detectado el mínimo o el máximo respectivamente.

Una vez detallado la forma de extraer información de la FPLS se expondrá la forma de generar las descripciones lingüísticas a partir de la FPLS. Para ello es necesario el uso de plantillas que ayuden a la generación de las descripciones. Primeramente se definirá una plantilla que utiliza el tipo y la potencia de la tendencia, la longitud de la misma y su localización. Ésta es:

> Es una tendencia $T \in TYPE$ que muestra un $P \in POWER$ de una longitud $L \in LONG$ situada en LOC.

Utilizando el FPLS de la Tabla I y los conjuntos de etiquetas en las Tablas de la II a la V se obtienen los resultados que se muestran en la Tabla VI. Las etiquetas generadas se destacan en el texto.

TABLE VI
UN FPLS DE EJEMPLO. FUENTE MORENO-GARCIA ET AL. [5]

$fpls_{t_i,t_{i+1}}$	Descripción lingüística
$fpls_{0.0,\ 17.0}$	Es una tendencia `descendente` que muestra un `descenso` de una longitud `corta` situada al `inicio pasado`.
$fpls_{17.0,\ 49.0}$	Es una tendencia `ascendente` que muestra un `ascenso` de una longitud `un poco corta` situada al `centro`.
$fpls_{49.0,\ 91.0}$	Es una tendencia `descendente` que muestra un `descenso` de una longitud `media` situada al `centro pasado`.
$fpls_{91.0,\ 94.0}$	Es una tendencia `plana` que muestra un `llano` de una longitud `muy corta` situada al `centro pasado`.
$fpls_{94.0,\ 131.0}$	Es una tendencia `descendente` que muestra un `ascenso` de una longitud `media` situada al `centro pasado`.

A continuación se expondrá un ejemplo utilizando el mismo FPLS que en el caso anterior que genera una descripción lingüística que detalla la longitud media de los segmentos, la tendencia media de la TS y los mínimos y máximos de la TS. La plantilla utilizada es la siguiente:

> La TS tiene una tendencia media $POWER$ y sus segmentos son de una longitud $LMEDIA$. En el min_k se presenta un mínimo y se encuentra un máximo en el max_k.

El resultado obtenido es el siguiente:

> La TS tiene una tendencia media `llana` y sus segmentos son de una longitud `un poco corta`. En el instante `inicio pasado` se presenta un mínimo y se encuentra un máximo en el `inicio pasado`.

V. Conclusiones y trabajos futuros

En este trabajo se ha presentado un nuevo método de generación de descripciones lingüísticas de TS a partir de la información capturada en una FPLS. Se ha mostrado cómo obtener información a partir de FPLS y qué se puede obtener a partir de la misma. También se ha detallado la forma en la que puede ser utilizada para generar descripciones lingüísticas. Se ha demostrado que FPLS contiene suficiente información de la TS para generar descripciones completas y complejas. Además, la forma de obtener dicha información es sencilla y eficiente permitiendo así la generación de descripciones de forma rápida.

Como trabajos futuros se pretende estudiar más detalladamente la información que se puede obtener de una FPLS. También se puede trabajar en la generación de descripciones del proceso de comparación de dos FPLS, por ejemplo, describir la comparación de partes de la serie que interesa comparar (por ejemplo, fases de un movimiento), o bien de partes que tienen un mayor parecido entre ellas. Cada una de estas partes puede corresponder a los valores que toma un sistema durante una fase. Finalmente se debe investigar en la creación de un marco de trabajo que permita generar las descripciones lingüísticas de una forma totalmente automática.

References

[1] E. Keogh, S. Chu, D. Hart, and M. Pazzani, "An online algorithm for segmenting time series," Proceedings 2001 IEEE International Conference on Data Mining, pp. 289–296, 2001.

[2] X. Huang, M. Matijaš, and J. A. K. Suykens, "Hinging Hyperplanes for Time-Series Segmentation," IEEE Transactions on Neural Networks and Learning Systems, vol. 24(8), 2013.

[3] E. Fuchs and T. Gruber and J. Nitschke, and B. Sick, "Online Segmentation of Time Series Based on Polynomial Least-Squares Approximations," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32(12), pp. 2232-2245, 2010.

[4] N. Marín, and D. Sánchez, "On generating linguistic descriptions of time series," Fuzzy Sets and Systems, vol. 285, pp. 6–30, 2016.

[5] A. Moreno-Garcia, J. Moreno-Garcia, Luis Jimenez-Linares, and Luis Rodriguez-Benitez, "Time series represented by means of fuzzy piecewise lineal segments," Journal of Computational and Applied Mathematics, vol. 318, pp. 156–167, 2017.

[6] A. Laurent, "Generating fuzzy summaries from fuzzy multidimensional databases," in: F. Hoffmann, D.J. Hand, N.M. Adams, D.H. Fisher, G. Guimarães (Eds.), IDA, in: Lecture Notes Computer Sciences, Springer, vol. 2189, pp. 24–33, 2001.

[7] R. Castillo-Ortega, N. Marín, and D. Sánchez, "A fuzzy approach to the linguistic summarization of time series," Journal of multiple-valued logic and soft computing, vol. 17(2-3), pp. 157–182, 2011.

[8] R. Castillo-Ortega, N. Marín, and D. Sánchez, "Linguistic query answering on data cubes with time dimension," International Journal of Intelligent Systems, vol. 26(10), pp. 1002–021, 2011.

[9] A. Ramos-Soto, A. Bugarin, and S. Barro and J. Taboada, "Linguistic descriptions for automatic generation of textual short-term weather forecasts on real prediction data," IEEE Transactions on Fuzzy Systems, vol. 23(1), pp. 44–57, 2015.

[10] J. Moreno-Garcia, L. Rodriguez-Benitez, J. Giralt, and E. del Castillo, "The generation of qualitative descriptions of multivariate time series using fuzzy logic," Applied Soft Computing, vol. 23, pp. 546-555, 2014.

[11] J. Moreno-Garcia, J. Abián-Vicén, L. Jimenez-Linares, and L. Rodriguez-Benitez, "Description of multivariate time series by means of trends characterization in the fuzzy domain," Fuzzy Sets and Systems, vol. 285, pp. 118–139, 2016.

[12] J. Kacprzyk, and A. Wilbik, "Linguistic summarization of time series using a fuzzy quantifier driven aggregation," Fuzzy Sets and Systems, vol. 159, pp. 1485–1499, 2008.

[13] D. Anderson, R.H. Luke III, J.M. Keller, M. Skubic, M. Rantz, and M. Aud, "Linguistic summarization of video for fall detection using voxel person and fuzzy logic", Computer Vision Image Understanding, vol. 113(1), pp. 80–89, 2009.

[14] G. Triviño, and M. Sugeno, "Towards linguistic descriptions of phenomena", International Journal of Approximate Reasoning, vol. 54(1), pp. 22–34, 2013.

[15] D. Sanchez-Valdes, and G. Triviño, "Computational Perceptions of uninterpretable data. A case study on the linguistic modeling of human gait as a quasi-periodic phenomenon", Fuzzy Sets and Systems, vol. 253, pp. 101–121, 2014.

[16] W. V. Leekwijck, E. E. Kerre., "Defuzzification: criteria and classification", Fuzzy Sets and Systems, vol. 108(2), pp. 159–178, 1999.