

# La generación de lenguaje natural: análisis del estado actual

Marta Vicente, Cristina Barros, Fernando S. Peregrino, Francisco Agulló, Elena Lloret

Universidad de Alicante, Departamento de Lenguajes y Sistemas Informáticos, Alicante,  
España

{mvicente, cbarros, fsperegrino, fagullo, elloret}@dlsi.ua.es

**Resumen.** El ser humano se comunica y expresa a través del lenguaje. Para conseguirlo, ha de desarrollar una serie de habilidades de alto nivel cognitivo cuya complejidad se pone de manifiesto en la tarea de automatizar el proceso, tanto cuando se trata de producir lenguaje como de interpretarlo. Cuando la acción comunicativa ocurre entre una persona y un ordenador y éste último es el destinatario de la acción, se emplean lenguajes computacionales que, como norma general, están sujetos a un conjunto de reglas fuertemente tipadas, acotadas y sin ambigüedad. Sin embargo, cuando el sentido de la comunicación es el contrario y la máquina ha de transmitir información a la persona, si el mensaje se quiere transmitir en lenguaje natural, el procedimiento para generarla debe lidiar con la flexibilidad y la ambigüedad que lo caracterizan, dando lugar a una tarea de alto nivel de complejidad. Para que las máquinas sean capaces de manejar el lenguaje humano se hacen necesarias técnicas de Lingüística Computacional. Dentro de esta disciplina, el campo que se encarga de crear textos en lenguaje natural se denomina Generación de Lenguaje Natural (*GLN*). En este artículo se va a hacer un recorrido exhaustivo de este campo. Se describen las fases en las que se suelen descomponer los sistemas de *GLN* junto a las técnicas que se aplican y se analiza con detalle la situación actual de esta área de investigación y su problemática, así como los recursos más relevantes y las técnicas que se están empleando para evaluar la calidad de los sistemas.

**Palabras clave.** Lingüística computacional, generación de lenguaje natural, *GLN*, fases, técnicas, evaluación.

## Natural Language Generation: Revision of the State of the Art

**Abstract.** Language is one of the highest cognitive skills developed by human beings and, therefore, one of the most complex tasks to be faced from the computational perspective. Human-computer communication processes imply two different degrees of difficulty depending on

the nature of that communication. If the language used is oriented towards the domain of the machine, there is no place for ambiguity since it is restricted by rules. However, when the communication is in terms of natural language, its flexibility and ambiguity becomes unavoidable. Computational Linguistic techniques are mandatory for machines when it comes to process human language. Among them, the area of Natural Language Generation aims to automatical development of techniques to produce human utterances, text and speech. This paper presents a deep survey of this research area taking into account different points of view about the theories, methodologies, architectures, techniques and evaluation approaches, thus providing a review of the current situation and possible future research in the field.

**Keywords.** Computational linguistics, natural language generation, NLG, stages, techniques, evaluation.

## 1. Introducción

La Lingüística Computacional (*LC*) es un campo en el que convergen diversas disciplinas: la lingüística aplicada, la informática y la inteligencia artificial. Se ocupa de las interacciones entre los ordenadores y los lenguajes humanos, por lo que en ocasiones se refiere también como Procesamiento del Lenguaje Natural (*PLN*), abarcando la comprensión del lenguaje y su generación.

Siendo un área de investigación en continuo desarrollo, la *LC* se ocupa en la actualidad de un conjunto de tareas cuyo propósito es comprender y/o producir lenguaje natural. Lo que busca, entonces, es conseguir sistemas, capaces tanto de entender el significado del lenguaje humano (lenguaje natural) como de generarla. Entre las citadas tareas cabe destacar la traducción automática [44], los sistemas de recuperación de información [90], la elaboración automática de resúmenes [63], o la

generación de lenguaje natural [66]. El presente artículo se va a centrar en esta última, la generación del lenguaje natural.

La Generación de Lenguaje Natural (*GLN*) es una tarea multidisciplinar. Su investigación y desarrollo incorpora conocimiento procedente de áreas diversas como la lingüística, la psicología, la ingeniería y la informática. El objetivo principal de la disciplina es investigar cómo se pueden crear aplicaciones informáticas capaces de producir por sí mismas textos de alta calidad en lenguaje natural. Para ello parte o bien de representaciones de datos estructurados y procesables (ficheros binarios, datos numéricos, bases de datos, etc.) o bien de textos escritos en lenguaje natural. La transformación de estos datos no puede ser realizada de manera directa, se han de tomar muchas decisiones relativas a diferentes aspectos como la determinación del contenido del mensaje y su estructura, las relaciones retóricas en varios niveles (texto, párrafo, frase), la elección de las palabras adecuadas, la disposición final del texto (título, cabeceras, pies de página, etc.) o los patrones acústicos en el caso de que la salida final del mensaje sea oral. Uno de los mayores desafíos de la *GLN* es la construcción de arquitecturas en las que se puedan tomar todas estas decisiones de manera que sea temporalmente abordable la producción de textos, ya sea en formato de texto o audio [4].

Respecto a las aplicaciones de la *GLN*, éstas son muy amplias y variadas. Existen en el mercado sistemas que se encargan, por ejemplo, de la generación de partes meteorológicas. También los hay que generan manuales de instrucciones, informes de evaluaciones académicas o que crean resúmenes. Éstas son solo algunas aplicaciones. Se revisarán con más detalle en la sección 2.2.

El objetivo de este artículo es realizar un análisis amplio y completo sobre el estado actual de la *GLN*. Para ello se describirán las distintas fases que los sistemas de *GLN* han de superar para llevar a cabo dicha tarea, analizando y detallando las técnicas y sistemas de *GLN* más destacados. También se comentarán los recursos disponibles más importantes para la creación de un sistema *GLN* así como los aspectos relevantes en cuanto a su evaluación. Finalmente, se aportará una

visión general tanto de las tendencias en investigación como de los retos que quedan todavía por abordar.

Para ello, este artículo está estructurado de la siguiente forma: en la sección 2 se presenta una clasificación de los distintos sistemas de *GLN* obedeciendo tanto al tipo de entrada del sistema como a los objetivos del mismo. A continuación, en la sección 3, se detallan los distintos enfoques bajo los que se llevan a cabo las tareas que integran un sistema de *GLN*. Posteriormente, en la sección 4, se describen las fases habituales cuando se crea un sistema de *GLN*, y en la sección 5 se hace una exposición de las técnicas más destacadas, asociándolas a aquellas fases. Seguidamente, en la sección 6, se describen algunas herramientas y los corpus que se utilizan más frecuentemente en generación, así como los recursos disponibles y dónde se pueden encontrar. Tanto la problemática inherente a la evaluación de los sistemas de *GLN* como los métodos para llevarla a cabo se cuentan en la sección 7. Para finalizar, en la última sección se exponen las conclusiones de este artículo junto a un análisis del futuro inmediato de la investigación en el campo de la *GLN*.

## 2. Clasificación de los sistemas de *GLN*

Un sistema de *GLN* se puede clasificar considerando diferentes criterios. El estudio realizado en el presente artículo pone de manifiesto dos factores imprescindibles que se han de tener en cuenta para abordar la tarea de generación. Dada su importancia en el contexto, tales factores se han seleccionado como criterios de clasificación: por un lado se considerará la entrada al sistema y por otro los objetivos que persiga (véase la tabla 1).

### 2.1. Según la entrada del sistema

Según el tipo de entrada que se introduce en el sistema se consideran dos posibles enfoques en la *GLN*: datos-a-texto (*D2T*: *data-to-text*) y texto-a-texto (*T2T*: *text-to-text*). Mientras que en la perspectiva *D2T* la entrada al sistema es un conjunto de datos que no conforman un texto (p. ej., datos numéricos representando información

**Tabla 1.** Clasificación de los sistemas *GLN*

Clasificación	Tipos	Definición
Entrada al sistema	<i>T2T</i> <i>D2T</i>	Reciben como entrada texto u oraciones Reciben como entrada datos generalmente numéricos, bases de datos, bases de conocimiento o corpus etiquetados
Objetivo del sistema: tipo de texto producido	Textos informativos	Generan informes a partir de un conjunto de datos
	Resúmenes	Generan texto que incluye las ideas fundamentales de uno o varios documentos
	Borradores	Generan versiones preliminares de textos técnicos
	Textos simplificados	Simplifican el pasaje de entrada para facilitar su comprensión
	Textos persuasivos	Generan relatos con el objetivo de convencer o motivar a los usuarios sobre un tema
	Sistemas de diálogos	Generan producciones interactivas manteniendo una comunicación entre el usuario y el sistema
	Explicaciones de razonamiento	Exponen en un texto los pasos seguidos en la resolución de un problema
	Recomendaciones	Generan sugerencias y valoraciones relativas a lugares, productos, servicios, etc

meteorológica), en el enfoque *T2T* el sistema sí parte de un texto del que se extrae la información relevante para construir la salida.

### 2.1.1. Datos-a-texto

El formato de la entrada de un sistema *D2T* puede ser muy variado. Tomar como punto de partida datos numéricos es habitual pero no deja de ser una posibilidad entre muchas otras para este enfoque. Es muy común, por tanto, encontrar sistemas que parten de ese tipo de información (de tal naturaleza es la información que proveen los sensores, estaciones meteorológicas, equipos médicos,...). Pero también lo es considerar otros orígenes de datos estructurados como corpus etiquetados, bases de datos, bases de conocimiento, archivos de *logs*, etc. Algunos autores emplean el término *concepto* para referirse a este tipo de datos no lingüísticos, por lo que también se menciona este enfoque como concepto-a-texto [3, 52].

Un ejemplo de sistema de este tipo es *Proteus* [23] que generaba un resumen de una partida del juego *tres en raya* a partir de una lista de movimientos. Lo que sigue es la lista de movimientos que constituyen una partida entre *Proteus* y el autor, Davey:

P:1 D:3 P:4 D:7 P:5 D:6 P:9

Le corresponde esta salida, construida desde el punto de vista del sistema:

*The game started with my taking a corner, and you took an adjacent one. I threatened you by taking the middle of the edge opposite that and adjacent to the one which I had just taken but you blocked it and threatened me. I blocked your diagonal and forked you. If you had blocked mine, you would have forked me, but you took the middle of the edge opposite of the corner which I took first and the one which you had just taken and so I won by completing my diagonal.*

### 2.1.2. Texto-a-Texto

Este tipo de enfoque puede tomar como datos de entrada tanto textos como oraciones aisladas. Existen muchas aplicaciones en la tarea de *GLN* que utilizan los sistemas *T2T* como la generación de resúmenes, la fusión de salidas en sistemas de búsqueda de respuestas o la simplificación de texto.

Un ejemplo de este tipo de sistemas se encuentra en [88] donde, tomando como entrada un conjunto de documentos procedentes de internet,

se generan artículos de *Wikipedia* cuya estructura es determinada por el dominio al que pertenece la producción (p. ej. los artículos sobre enfermedades incluyen cuatro apartados: *diagnóstico, causas, síntomas y tratamiento*).

## 2.2. Segundo los objetivos del sistema

Tal y como se dijo anteriormente, los sistemas de *GLN* también pueden ser clasificados atendiendo al propósito para el que han sido creados, y tal propósito se revela en el tipo de producción que genera el sistema. Se han recogido en este estudio los más relevantes:

- *Generación de textos informativos.* La finalidad del sistema en este caso es generar informes a partir de datos factuales (información objetiva). *FoG* [36] y *SumTime* [83], por ejemplo, son dos sistemas de este tipo. Toman como entrada información numérica procedente de sistemas de simulación que representan magnitudes como la temperatura, el nivel de las precipitaciones o la velocidad del viento en diferentes lugares y tiempos. Existen aplicaciones en otros contextos, como *SkillSum* [102], herramienta que genera informes sobre evaluaciones académicas, cuyo objetivo inicial era ayudar a personas con escasos conocimientos de aritmética y lengua.
- *Generación de resúmenes.* Este tipo de generación tiene como objetivo producir una versión abreviada de una o más fuentes de información. Estos resúmenes pueden estar asociados a diversos campos: resúmenes médicos [77], de ingeniería [103], financieros [55], deportivos [86] o de patentes [71], entre otros.
- *Generación de textos simplificados.* Estos sistemas se han diseñado como herramientas de ayuda a personas con alguna discapacidad o con problemas de comprensión lectora, ya provengan de su desconocimiento del idioma o de alguna dificultad cognoscitiva. Existen sistemas de este tipo que producen texto dirigido a personas afásicas [84] o texto que permite a personas invidentes examinar gráficos [29]. *LexSiS* es otro ejemplo de sistema simplificador, éste en español y más centrado en el léxico. Y *SkillSum*, que se ha mencionado anteriormente como ejemplo de generación de informes, incorpora técnicas para producir textos que personas casi analfabetas puedan leer.
- *Generación de textos persuasivos.* En [24] se emplea el término *afectivo* para caracterizar a aquellos sistemas de *GLN* que pretenden influenciar o toman en cuenta el estado emocional del oyente. Al margen de la función comunicacional de lenguaje, y en ese sentido, éste se puede emplear para lograr múltiples propósitos (motivar, persuadir, reducir el estrés,...). Entre los sistemas diseñados bajo estas premisas, como ejemplo, encontramos un generador de cartas que pretenden disuadir a los usuarios de fumar (*STOP* [82]), un sistema que busca disminuir la ansiedad de pacientes con cáncer proporcionándoles información [16] o algunos sistemas que procuran entretenimiento, ya sea mediante acertijos [10] o cuentos [56].
- *Generación de sistemas de diálogo.* Los sistemas de diálogo están orientados a la comunicación entre las personas y las máquinas. Se caracterizan porque el usuario interactúa con el sistema, que va generando oraciones en lenguaje natural condicionadas por el contexto inmediatamente anterior. Se han diseñado sistemas de este tipo para múltiples propósitos, como aquellos que ayudan a mejorar habilidades (*G-Asks* [62], escritura de ensayos) o que tutorizan para incrementar el conocimiento de ciertas materias a través del diálogo (*Beetle II* [26], electricidad y electrónica), pero también están los que permiten elaborar itinerarios de viaje (*CMU Communicator* [87]) o los que se utilizan en entornos virtuales de juegos, por ejemplo (*GIVE software* [49]).
- *Generación de explicaciones de razonamiento.* La salida de este tipo de sistemas es la explicación de una secuencia de pasos que el sistema ha seguido en la ejecución de un algoritmo, el procesamiento de una transacción,

la resolución de un problema matemático, etc. Un ejemplo: *P.Rex* [30], que es una herramienta de explicación de demostraciones de teoremas.

- *Generación de recomendaciones.* Parece consecuencia inevitable de la posibilidad de disponer de información relativa a los gustos y opiniones de los usuarios, que se desarrollen herramientas para procesar y traducir esos datos a recomendaciones y tendencias. En ese contexto tiene sentido la aparición de sistemas que, en referencia a cualquier campo o materia, ya se trate de restaurantes, cines, destinos turísticos o productos tecnológicos, proporcionen los resultados del procesamiento en lenguaje natural. Un sistema que realiza esta tarea, por ejemplo, es *Shed* [60], que en base al perfil proporcionado por el usuario y obteniendo la información de la Web 2.0 (reseñas, tweets, etc.), recomienda dietas nutricionales personalizadas.

### 3. Enfoques genéricos para abordar la GLN

En esta sección se realizará un análisis de los enfoques más relevantes empleados para abordar la tarea de la *GLN*, que en este caso son los enfoques basados en conocimiento y los enfoques estadísticos.

Por un lado, hablamos de sistemas basados en conocimiento cuando las técnicas que lo implementan se nutren de fuentes con un marcado carácter lingüístico, como diccionarios, tesauros, bases de conocimiento léxicas, reglas o plantillas. De esos recursos se extrae información morfológica, léxica, sintáctica, semántica, etc. Por otro lado, cuando un sistema se desarrolla bajo un enfoque estadístico, la información que necesita para transformar la entrada a un texto en lenguaje natural procede principalmente de un corpus y de las probabilidades extraídas de los textos que lo componen, pudiendo estos estar etiquetados o no. Estos sistemas están menos restringidos a un dominio o a un idioma que los basados en conocimiento, dado que, si el corpus empleado es adecuado tanto en tamaño como en tipo de contenido, no

tiene tantas restricciones como las que resultan de generar unas reglas que han de ceñirse a las características del contexto para el que se desarrollan o a las peculiaridades de una lengua concreta, como suele ser el caso en los sistemas basados en conocimiento.

Se ha de considerar que estos enfoques no son excluyentes y que, aunque en muchas ocasiones todas las técnicas que se aplican en un sistema se asocian a una perspectiva concreta, existen aproximaciones híbridas que combinan los enfoques estadísticos con técnicas basadas en conocimiento. Tanto las técnicas como las tareas para las que se desarrollan se verán en secciones posteriores, baste decir aquí que, siendo que en un sistema de *GLN* se suceden diferentes etapas de tratamiento de los datos, es posible que cada una de ellas se aborde desde distintos planteamientos.

Se presenta a continuación una descripción general de ambas perspectivas. Ya más adelante, en la sección 5, se profundizará en las técnicas asociadas a cada enfoque.

#### 3.1. Enfoques basados en conocimiento

El factor común de los sistemas basados en conocimiento es su capacidad para representar explícitamente el conocimiento. Con tal propósito, estos sistemas hacen uso de herramientas como ontologías, conjuntos de reglas o tesauros.

Se considera que tales sistemas están constituidos a su vez por dos subsistemas: una base de conocimiento y un motor de inferencia. La base de conocimiento es un tipo de base de datos para la gestión del conocimiento que proporciona los medios necesarios para la recolección, organización y recuperación del mismo. El motor de inferencias es la parte del sistema que razona utilizando el contenido de la base de conocimiento en una secuencia determinada. Este motor examina las reglas de la base de conocimiento una por una, y cuando se cumple la condición de una de las reglas, se realiza la acción especificada para la misma.

Esta sistematización del conocimiento se sustenta sobre teorías lingüísticas que fundamentan el diseño y aplicación de las técnicas adecuadas. A continuación se presentan las teorías más relevantes, en tanto que son las más empleadas en el

desarrollo de sistemas basados en conocimiento o híbridos.

- La teoría de la estructuración retórica (*RST: Rhetorical Structure Theory*) [65].

Es una de las principales teorías empleadas en la *GLN* y está relacionada tanto con la cohesión del discurso como con la estructura de los mensajes y los párrafos. La idea que subyace a la *RST* es la posibilidad de descomponer recursivamente cualquier texto en un conjunto de elementos entre los que se establecen una serie de relaciones retóricas o discursivas, denominadas esquemas. Ejemplos de tales relaciones se pueden observar en la figura 1. El análisis de las relaciones retóricas considera, además, las intenciones de quien origina la comunicación así como los efectos buscados en el que la recibe. Algunos elementos del conjunto son más relevantes y se constituyen como núcleos, mientras que los elementos que dependen de ellos se refieren como satélites. Dos oraciones pueden estar también relacionadas, y bajo el mismo esquema lo estarían sus núcleos. En la figura 1 obtenida de [64] se puede ver la estructuración de una oración basada en esta teoría.

- La gramática sistémico funcional (*SFG*) [39].

Para la lingüística sistémico-funcional el lenguaje es un recurso que permite construir significado, y se estratifica en tres niveles de abstracción: semántico, lexicogramático y fonológico/grafológico. La *SFG* describe cómo se pueden expresar las funciones comunicativas e incide en la dimensión social del lenguaje. Esta teoría considera tres dimensiones del significado: la proposicional, la interpersonal (relaciones entre emisor y receptor y cómo influyen en el uso de la lengua) y la textual (cómo se estructura y empaqueta la información en un texto). Generalmente estas dos últimas metafunciones de la lengua no se tratan en otras teorías lingüísticas [7].

- Gramática de adjunción de árboles (*Tree-adjoining grammars - TAG*) [47].

Una *TAG* es una gramática lexicalizada compuesta por un conjunto finito de árboles básicos que incorporan contenido semántico. Partiendo de tales árboles, mediante operaciones de sustitución o unión (*adjoining*), es posible construir un nuevo árbol etiquetado que represente la derivación correspondiente a una oración. Una de las ventajas que proporciona el uso de una *TAG* es que resuelve en una misma acción la planificación de los mensajes y su realización como oración [50], aunque esto comporte cierta pérdida de flexibilidad. En la figura 2 vemos un ejemplo de uso.

- Teoría sentido-texto de Mel'čuk (*Meaning-Text Theory (MTT)*) [73].

Esta teoría utiliza un modelo de representación que diferencia los niveles semántico, sintáctico, morfológico y fonético. Estos tres últimos se desdoblan en representaciones profundas y superficiales. Según este modelo, el proceso de *GLN* consistiría en la transformación progresiva de las representaciones a través de los niveles mencionados. Se utilizan reglas de equivalencia para realizar la conversión de un nivel a otro.

- Teoría del centrado [37, 38].

La coherencia del discurso y el modo en que se relacionan las entidades que lo componen se tratan en los llamados modelos de cohesión discursiva. Entre ellos, la teoría del centrado ha sido muy utilizada en *LC* en relación al problema de la anáfora, fenómeno lingüístico que se da cuando elementos de la oración hacen referencia a entidades que ya han aparecido en el discurso. Según esta teoría, un elemento de una parte del discurso a nivel local se constituye como el foco de atención o centro de ese contexto, como la entidad más relevante a la que se refiere el resto de preferencias. En lo que al proceso de generación respecta, afecta, por ejemplo, a la selección y uso de pronombres y descripciones.

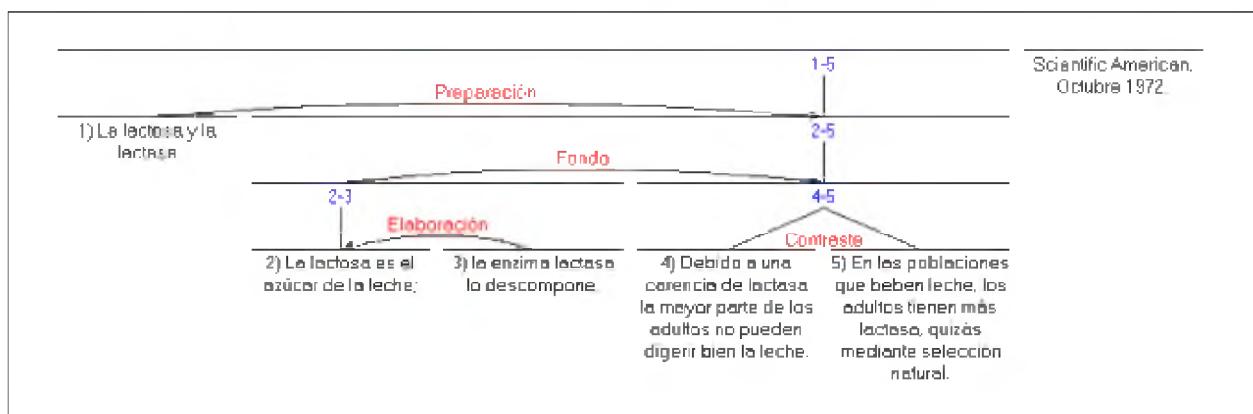


Fig. 1. Árbol RST

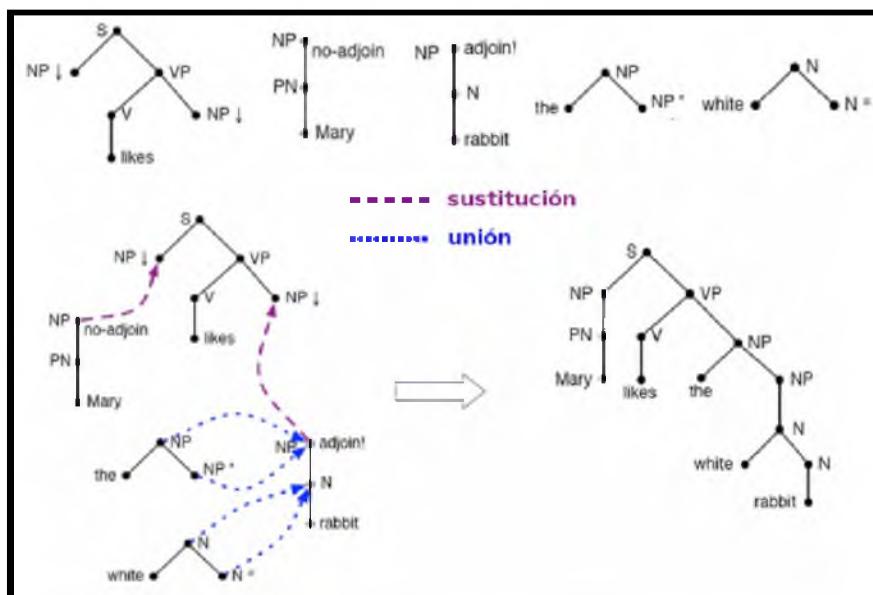


Fig. 2. Ejemplo de derivación en un TAG. En la parte superior de la imagen aparecen los árboles elementales. Aplicando sustitución y adjunción se genera el árbol de derivación para la oración "Mary likes the white rabbit"

### 3.2. Enfoques estadísticos

Como se indicó anteriormente, los enfoques estadísticos se basan en las probabilidades extraídas desde un volumen de texto base, ya sea un corpus, anotado o no, texto procedente de la Web, etc. Una de las herramientas primordiales para este tipo de enfoques son los modelos de lenguaje (*LM: Language Models*).

Un *LM* estadístico es un mecanismo que define la estructura del lenguaje, es decir, restringe adecuadamente las secuencias de unidades lingüísticas basándose en una distribución de probabilidad que expresa la frecuencia de aparición de una secuencia de  $n$  palabras  $P(w_1, w_2, \dots, w_n)$  en un conjunto de textos. Así, un buen *LM* puede determinar, a partir de la probabilidad asociada a una frase, si está construida correctamente. Se dice en este

caso que el *LM* acepta la frase. La rechaza cuando la probabilidad asociada es baja, indicando que tal secuencia no pertenece al lenguaje sobre el que se ha realizado la distribución de probabilidad. Lo interesante para la tarea de generar lenguaje natural, es que un buen *LM* puede predecir cómo se va a transformar una entrada (o parte de esa entrada) dentro del sistema o en una de sus fases. Uno de los factores que determina la calidad del *LM* es el tamaño del corpus o fuente de datos a partir del cual se entrena, dado que la cantidad de contextos de uso de una palabra o amplitud del dominio al que se pueda aplicar el *LM* será proporcional a la dimensión del cuerpo de entrenamiento. A continuación se describen tres de los *LM* más utilizados en *GLN*.

#### ■ *Modelo de n-gramas*

Un *n*-grama es una subsecuencia de *n* elementos de una secuencia dada. El modelo de *n*-gramas es un tipo de modelo probabilístico que permite hacer una predicción estadística del próximo elemento que aparecerá en una secuencia de elementos sucedida hasta el momento. Estos modelos pueden ser definidos por una cadena de Márkov<sup>1</sup> de orden *n*-1. La implementación de estos modelos es sencilla y resultan muy útiles en la construcción de algoritmos de reconocimiento y de aprendizaje automático. Sin embargo, los modelos *n*-gramas son muy generales, por lo que es necesario adaptarlo a cada aplicación. Otra de sus limitaciones es que solo son capaces de capturar relaciones a corta distancia.

#### ■ *Modelos basados en gramáticas estocásticas*

Las gramáticas estocásticas son aquellas en las cuales cada regla de la gramática tiene asociada una probabilidad, de modo que el resultado de la aplicación de las reglas proporciona una probabilidad que ha sido derivada de ellas. Los modelos basados en gramáticas estocásticas representan las restricciones del lenguaje de una manera natural. Además, permiten modelar dependencias tan largas como

<sup>1</sup>Se conoce como cadena de Márkov a un tipo especial de proceso estocástico discreto en el que la probabilidad de que ocurra un evento depende solamente del evento inmediatamente anterior.

se deseen, aunque la definición de estos modelos y sus parámetros entraña gran dificultad para tareas complejas.

#### ■ *Modelos del lenguaje factorizados*

Los modelos del lenguaje factorizados (*FLM: Factored Language Model*), presentados por [9], son una extensión de los *LM*. En los *FLM*, una palabra se ve como un vector de *k* características o factores(*Factors*), de modo que  $w_t \equiv \{f_t^1, f_t^2, \dots, f_t^K\}$ . Estos factores pueden ser cualquier cosa, incluyendo clases morfológicas, raíces, o cualquier otra característica léxica, sintáctica o semántica. Un *FLM* proporciona un modelo probabilístico  $P(f|f_1, \dots, f_N)$  donde la predicción de una característica *f* está basada en *N* padres  $\{f_1, \dots, f_N\}$ . Por ejemplo, si *w* representa un token de palabra y *t* representa una categoría gramatical (*POS: Part-Of-Speech*), la expresión  $P(w_i|w_{i-2}, w_{i-1}, t_{i-1})$  ofrece un modelo para predecir el actual token de palabra basándose en el modelo tradicional *n*-grama así como en la *POS* de la palabra anterior.

### 3.3. Enfoques híbridos

Los enfoques híbridos son aquellos que combinan las técnicas basadas en conocimiento y las estadísticas para realizar las distintas tareas que competen a la *GLN*. La aplicación *FLIGHTS* [101] es un ejemplo de este tipo de sistemas híbridos. En él se presenta información de vuelos de forma personalizada para cada usuario (p.ej., considerando si el usuario es un estudiante o un viajero frecuente). En su desarrollo se consideran distintas bases de conocimiento (modelos de usuario, modelos de dominio e historial de diálogos) para realizar la selección del contenido que debe aparecer en la salida. Posteriormente dicho contenido se estructura a partir de plantillas y se genera el texto final utilizando el framework *OpenCCG*<sup>2</sup>. Esta herramienta emplea internamente modelos de *n*-gramas y *FLM*. Tanto la selección como la estructuración o generación del texto son diferentes fases de la *GLN*, se ven con más detalle a continuación.

<sup>2</sup><http://openccg.sourceforge.net/>

#### 4. Fases de un sistema de GLN

En la sección anterior se ha presentado un conjunto de enfoques y teorías que se pueden utilizar en un proceso de *GLN*. Desde una perspectiva general, este proceso podría describirse como la realización de un conjunto de tareas cuya finalidad es transmitir, en lenguaje natural, cierta información a una audiencia para conseguir un objetivo. Por tanto, tan importante para el sistema es caracterizar la entrada y la salida como considerar el contexto y el objetivo comunicativo. Especificar las tareas o fases para que el sistema cumpla su propósito implica que cada una de ellas debe contemplar y concretar tales aspectos. Sin embargo, es éste un campo para el que no existe un consenso claro debido a la multiplicidad de opciones que posibilita, ya sea en cuanto a los objetivos comunicativos, a los dominios de aplicación o a los tipos de texto generado.

Considérese, por ejemplo, la entrada al sistema. Tal y como se comenta en [12], para caracterizar la entrada se tendría en cuenta en primer lugar el tipo de dato, esto es, por un lado su formato (grafo semántico, base de datos, plantilla,...) y por otro el lenguaje en que se presenta (lógica de primer orden, ontologías,...). También se debería considerar su tamaño (un grafo de dimensiones reducidas, miles de webs,...) y su grado de (in)dependencia tanto respecto al dominio como respecto a la tarea para la que se emplee. Son muchas variables a tener en cuenta tanto a nivel general a la hora de diseñar el sistema como a nivel particular en el momento de especificar cada una de sus fases. Sin embargo, esto no ha impedido que se hayan llevado a cabo esfuerzos para definir arquitecturas genéricas en el ámbito de la *GLN*, como *RAGS* [69], o que se acostumbre a referir las tareas y módulos de este ámbito haciendo alusión al modelo planteado en [81]. Se describen estos planteamientos a continuación.

Se ha mencionado *RAGS* (Reference Architecture for Generation Systems). Este sistema se desarrolló a finales del siglo pasado a partir del análisis de las arquitecturas empleadas por diver-

sos sistemas de *GLN* existentes en el mercado<sup>3</sup>. El proyecto no buscaba definir un estándar, si no más bien establecer un punto de referencia. Su propósito principal era favorecer la reusabilidad y la comparación de las tecnologías empleadas en *GLN*, así como favorecer la evaluación de las mismas. Del análisis realizado, se extrae una clasificación de las tareas similar a la de [81], pero divergente en algunos matices. Así, en [13], se describen las siete fases que determina *RAGS*, que suelen aparecer agrupadas en tres conjuntos:

- *Lexicalización y generación de expresiones de referencia*, encargándose de la elección de las palabras finales y de la planificación de la descripción de los objetos.
- *Agregación y segmentación*, responsables de la combinación de varias estructuras en una más compleja o de la división de la información en oraciones y párrafos.
- *Estructuración retórica, ordenamiento y centrado/prominencia/tema*, a cargo de la determinación de las relaciones retóricas entre los mensajes, de la elección del orden de los elementos del texto y de la relación tanto de las oraciones entre sí como de éstas con el tema del discurso. La *teoría del centrado* se explicó con más detalle en la sección 3.1, y hacía alusión a la determinación del foco o centro local al que se referían otros elementos cercanos. En cuanto a la *prominencia*, ésta indica, en relación a una entidad o hecho, su relevancia respecto al tema del discurso. En la tarea de generación constituye un criterio para determinar la incorporación de ciertos elementos al texto final.

Sin embargo, no se presenta este conjunto como un grupo estático de componentes, si no como el conjunto de tareas más identificadas en los sistemas de *GLN* que analizaron. Lo que concluyen los autores de *RAGS* es que en los sistemas reales una tarea puede implementarse en diferentes módulos, no está exclusivamente asociada a

<sup>3</sup>No es el propósito de este artículo describir las fuentes del proyecto *RAGS*, puede consultarse el estudio completo en el informe técnico [74].

un módulo concreto, e incluso puede ser implementada en varios módulos del mismo sistema, poniendo de manifiesto la dificultad asociada a una especificación concreta de las interfaces correspondientes.

Si bien la idea que prevalece hasta ahora es que ni las fronteras entre módulos ni tan siquiera los módulos mismos están claramente definidos, sí es posible tomar un punto de partida que ayude a determinar la forma que corresponda a un sistema de *GLN* en lo más general. En este artículo ese punto de partida se encuentra planteado en un texto escrito en el año 2000 por Ehud Reiter y Robert Dale [81] que se ha convertido en texto de referencia en el campo de la *GLN*. De acuerdo con el mismo, las funcionalidades que corresponden a un sistema de *GLN*, que se desarrollarán más abajo, se distribuyen en siete tareas y la relación que se establece entre ellas se puede representar mediante una arquitectura básica de tres módulos:

- *Macro planificación*. El primer módulo del sistema debe determinar qué decir y organizarlo en una estructura coherente, dando lugar a un plan del documento. Lo hace mediante dos tareas:
  - Selección de contenido
  - Estructuración del documento
- *Micro planificación*. Partiendo del plan del documento que llega como entrada desde el módulo anterior, se generará una planificación del discurso. Se seleccionan las palabras y las referencias adecuadas, se dota a los mensajes de una estructura lingüística y se agrupa la información en oraciones. Las tareas que intervienen son las siguientes:
  - Agregación de sentencias
  - Lexicalización de estructuras sintácticas
  - Generación de expresiones referenciales
- *Realización*. A estas alturas del proceso, se dispone de una representación de las oraciones que van a conformar la salida del sistema. El módulo de realización genera la salida final, sea ésta texto o habla, las oraciones concretas que la conforman así como la estructura

que hayan de presentar. Dos tareas se suceden finalmente:

- Realización lingüística
- Realización de la estructura

En la figura 3 se muestra esta distribución de funcionalidades en tres bloques principales. En ocasiones se añade un bloque de preprocesado de datos, pero esto se comentará con más detalle en la siguiente sección. Antes de detenernos en la descripción de cada una de las tareas, un apunte más respecto a la arquitectura de un sistema *GLN*.

Las diferentes tareas aludidas se relacionan en el planteamiento de [81] de forma secuencial. Los procesos de transformación sobre la información se suceden unidireccionalmente, lo que impide que se produzcan revisiones o que se pueda modificar lo establecido en fases superadas. Se trata en este caso de una *arquitectura secuencial*. Pero existen otras arquitecturas posibles, tal y como vemos en [31]. Una *arquitectura integrada o monolítica* apenas establece separación entre los módulos. Las partes del sistema no se pueden reutilizar pero, a cambio, el sistema es eficiente [48]. En [43] se contempla un ejemplo de *arquitectura interactiva*, en la que la información ya no fluye en una dirección única y decisiones posteriores pueden modificar la salida de los módulos. En *arquitecturas de pizarra* los módulos modifican la información, que está ubicada en una zona común, hasta que el resultado es aceptable [14]. Por último, se empleará la caracterización de *arquitectura basada en revisión* cuando se dé el caso de que la información circule cíclicamente por los diferentes módulos del sistema hasta que se obtenga el resultado.

Hasta ahora se ha hablado de la posible clasificación de los sistemas de *GLN* según su entrada (datos o texto) o según su propósito (sección 2). También se han introducido los posibles enfoques desde los que se afronta la tarea de la *GLN* (sección 3) y se ha planteado una arquitectura de referencia. El marco definido conduce al análisis de las técnicas en que se concretan todos los aspectos mencionados. Sin embargo, esto no se verá hasta la sección 5 dado que se hace imprescindible para comprender el desarrollo de tales técnicas y, más

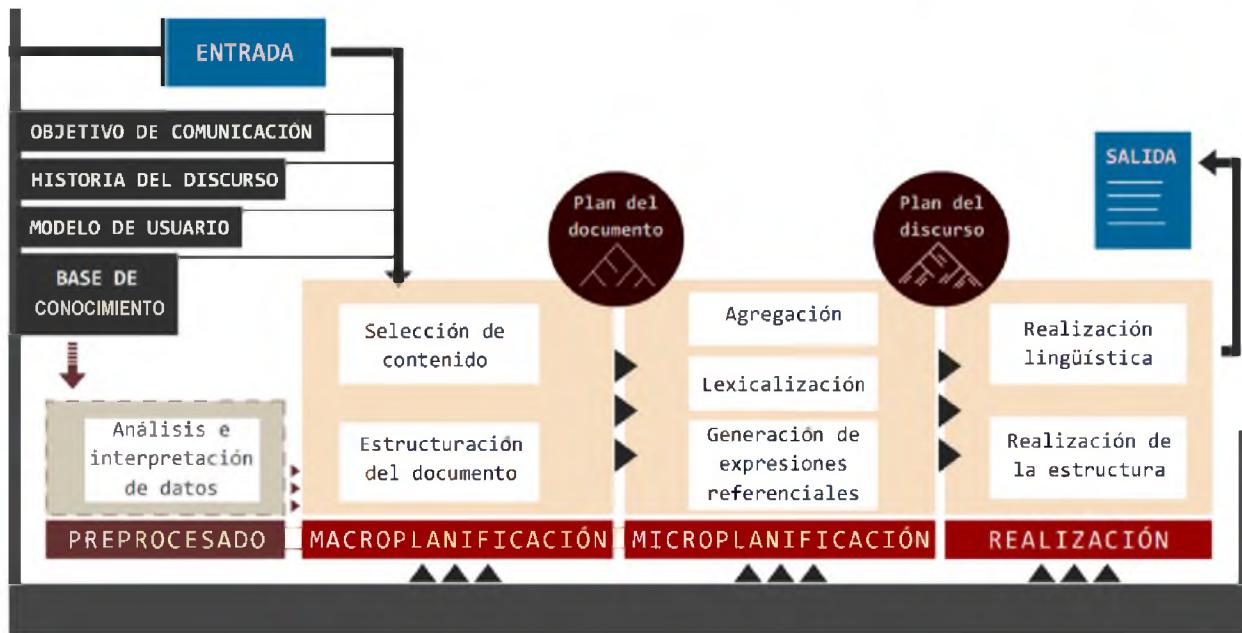


Fig. 3. Arquitectura de referencia para un sistema general de GLN

aún, para seleccionar las más adecuadas en el diseño de un sistema de *GLN*, un conocimiento de las tareas específicas que pretenden resolver. Tal es el cometido de las próximas secciones.

#### 4.1. Macro planificación

La fase de macro planificación es también conocida como fase de planificación del documento. Comprende la toma de decisiones respecto a qué información incluir en la salida del sistema y la determinación de la estructura que adoptará (la forma de organizar dicha información).

En la sección 2.1 se han clasificado los sistemas de *GLN* según el tipo de entrada. En este sentido, se establecía una diferencia entre aquellos sistemas en los que la entrada era un texto y aquellos cuya entrada estaba conformada por un conjunto de datos estructurados. Profundizando en esta distinción inicial, lo que se constituya como entrada puede adoptar muchas formas. La manera más sencilla de verlo es a través de algunos ejemplos. Si el sistema va a transmitir conclusiones extraídas de una encuesta, la entrada la conformarán las

respuestas de los usuarios a las preguntas (sistema *STOP* [82]). Si se trata de un sistema de recomendaciones, la entrada la constituye tanto un conjunto de descripciones de los elementos a recomendar como las anteriores consultas del usuario, sus preferencias o la pregunta concreta que realiza al sistema, por ejemplo, la solicitud “*Compara restaurantes en el centro de Manhattan*” que se le hace al sistema *MATCH* [46]. Un ejemplo más: el de un sistema que proporciona el resumen meteorológico de un mes y que cuenta con una base de datos con abundante información acumulada (sistema *SUMGEN-W* [81]). En este sistema se decide que la información entre como un conjunto de registros diarios (véase figura 4).

Lo que se observa a través de estos ejemplos es que la tarea de formalizar la entrada al sistema, y a la fase de macro planificación por ser la primera del proceso, debe llevarse a cabo desde una perspectiva amplia. De esta forma, tal y como aparece reflejado en la figura 3, se ha de considerar junto a la fuente de conocimiento de la que se nutre el sistema (ya sea corpus, ontología, base de datos o texto) tanto el objetivo comunicativo como el contexto del discurso (la historia del discurso, lo

```

type: dailyweatherrecord
  date: [day: 17
         month: 05
         year: 1995]
  temperature: [minimum: [unit: degrees-centigrade
                         number: 10.1]
                maximum: [unit: degrees-centigrade
                           number: 21.3]]
  rainfall: [unit: millimetres
             number: 12]

```

**Fig. 4.** Entrada para el sistema *SUMGEN-W* de resúmenes meteorológicos. Registro diario

generado anteriormente). Y también el modelo de usuario: a quién va dirigido el texto. Pues todo ello condiciona tanto el resultado final como el de las etapas intermedias.

Por otro lado, como se indicó más arriba, algunos autores añaden una fase de preprocesamiento previa a la fase de macro planificación [78]. Esta fase sería necesaria cuando los datos han de ser analizados e interpretados. El análisis se ocupa de extraer patrones de los datos mientras que la etapa de interpretación se realiza para inferir mensajes útiles en el dominio de la aplicación. El registro que se ha mostrado para el sistema *SUMGEN-W* (figura 4) se obtendría, por ejemplo, tras el procesamiento de un conjunto de datos como el de la figura 5.

Respecto a la salida de este módulo, suele adoptar la forma de árbol con mensajes en sus nodos terminales, es el llamado plan del documento. Los mensajes son unidades elementales del discurso procedentes del dominio susceptibles de ser expresadas a través de oraciones. Y junto a los mensajes se incorpora información relativa al modo en que se relacionan. La figura 6 es el plan del documento asociado a la tabla 2. Pertenece a un sistema que procesa información meteorológica, *SumTime*, que se ha mencionado anteriormente en la sección 2.2.

**Tabla 2.** Entrada para *SumTime*. Previsión de viento para 19 de septiembre, 2000

Hora	Dirección	Velocidad del viento
06:00	SE	11
09:00	SSE	13
12:00	SSE	14
15:00	SSE	15
18:00	SE	18
21:00	SE	23
00:00	SE	28



**Fig. 6.** Plan del documento de *SumTime*

Como se puede observar en la figura 6, el nodo raíz indica que la información deberá estar contenida en un párrafo. También facilita información respecto al orden que se establece entre los nodos y al valor de los parámetros que comparten.

#### 4.1.1. Selección de contenido

La selección de contenido es la tarea que permite al sistema elegir y obtener la información que debería ser comunicada en el texto final: la más relevante para el usuario acorde con el objetivo comunicativo y la situación, que incluye aspectos tan diversos como el tamaño que corresponde a la salida del sistema, el nivel de conocimiento del usuario o la historia del discurso hasta el momento.

Dado que esta fase es la que menos relación guarda con el procesamiento lingüístico, algunos autores la han situado fuera del sistema de *GLN*. Es el caso de [28], donde los autores proponen una nueva frontera que deja fuera de la disciplina aquellas acciones cuya naturaleza no sea estrechamente lingüística, considerando que las estrategias de selección no lo son. Un planteamiento más

```

95.121,1,5,1.00,100,-12.4,-4.406,-1.72,-1.298,1016,17.22,17.45,20.09,-13.58,-5.044,102.3,.2,149.9,92.8,0,.019
95.121,1,5,1.25,115,-12.49,-7.15,-2.166,-4.877,1016,17.13,17.45,20.1,-12.94,-4.867,103.1,.2,23.3,.195.0,.026
95.121,1,5,1.50,130,-12.93,-6.463,-2.053,-4.583,1016,17.02,17.22,20.15,-13.21,-4.801,101.9,.2,359.3,24.8,0,.043
95.121,1,5,1.75,145,-12.31,-5.145,-1.764,-3.092,1016,16.75,17.04,20.19,-12.98,-4.607,102.0,.2,330.6,1.416,0,.056
95.121,1,5,2.00,200,-14.71,-5.679,-2.034,-2.545,1016,16.52,16.94,20.21,-12.99,-4.658,104.3,.221,330.2,1.059,0,.066
95.121,1,5,2.25,215,-15.44,-6.819,-2.178,-4.025,1016,16.34,16.69,20.25,-13.62,-4.819,103.6,.2,329.9,0,0,.067
95.121,1,5,2.50,230,-12.69,-8.59,-2.624,-7.07,1016,16.26,16.65,20.29,-14.03,-4.859,104,.2,329.9,0,0,.073
95.121,1,5,2.75,245,-10.04,-4.78,-1.758,-1.46,1016,16.08,16.46,20.35,-14.59,-4.945,103.9,.2,329.9,0,0,.073

```

**Fig. 5.** Datos iniciales que preprocesa *SUMGEN-W*

flexible se propone en [66], el autor establece una división entre dos aplicaciones: un generador, que se haría cargo del procesamiento lingüístico y otra, a la que llama de *el speaker*, cuya función sería la de determinar qué decir pasando esta información al generador. Aun con tal separación, se considera que formarían parte de un mismo sistema que las necesita a ambas.

#### 4.1.2. Estructuración del documento

Para conseguir un texto coherente es preciso que los elementos que lo configuran estén debidamente estructurados. Cohesión y coherencia son los principios que permiten que un conjunto de oraciones constituyan un discurso. Hacen referencia al modo en que las unidades textuales se relacionan entre sí y permiten que se puedan realizar inferencias a partir de la información proporcionada o que se puedan identificar de forma no ambigua los elementos correferentes. Esta coherencia compete tanto a las oraciones como a los mensajes que las componen.

Por todo ello, se hace necesaria una tarea que, ya sea durante el proceso de seleccionar los mensajes o después de haberlo hecho, determine la estructura que vaya a tener en el texto final, la relación que guardan unos elementos con otros, dado que tal ordenación supone el primer paso hacia el discurso correcto.

De nuevo, tal y como ocurre con el resto de etapas en *GLN*, tanto para seleccionar las técnicas con que afrontar esta fase como para determinar el tipo de estructura que se necesita, se han de considerar aspectos extralingüísticos. No tendrá la misma estructura un texto que explique un proceso

que uno que compare dos propuestas, esto atendiendo al objetivo comunicativo. Si se considera el contexto, se debe mostrar una continuidad con las estructuras precedentes para no desorientar al usuario.

#### 4.2. Micro planificación

En el módulo de micro planificación se toma como entrada el plan del documento, que es el producto de la macro planificación en el que se indican los mensajes que deben formar parte del texto final así como la relación que se establece entre ellos, su estructura. Las operaciones que se llevan a cabo a partir de ese plan en las diferentes etapas que competen a este módulo de micro planificación son eminentemente lingüísticas. Para conseguir su cometido pueden tomar como fuente bases de conocimiento u ontologías y consideran, para realizar adecuadamente sus elecciones, tanto el objetivo comunicativo como el modelo de usuario, esto es, la caracterización del receptor.

La salida de este módulo es la especificación del texto o plan del discurso. El texto que se ha de generar deberá estar completamente caracterizado en tal especificación. La forma de la salida de nuevo adopta la forma de árbol en el que se presentarán los límites de las oraciones, sus relaciones sintácticas, las palabras que contienen o las relaciones de correferencia.

Son las decisiones que se han de tomar en este módulo las que determinan las etapas a seguir:

- *Agregación*, o cómo agrupar las estructuras procedentes del plan de documento para constituir conjuntos coherentes de mensajes.

- *Elección léxica o lexicalización*, pues se deben determinar las palabras que se utilizarán para expresar los conceptos y los hechos contenidos en el plan del documento.
- *Generación de Expresiones Referenciales (GER)*, dado que un mismo concepto o entidad puede aparecer en diferentes ocasiones a lo largo del texto y se debe elegir de qué modo referenciarlo o describirlo en cada aparición.

#### 4.2.1. Agregación

En la etapa de agregación se han de determinar las combinaciones que se realizarán sobre los elementos informativos que incorpora el plan del documento. También es competencia de esta fase establecer un orden entre el resultado de tales combinaciones. Para algunos autores, el objetivo sería eliminar la redundancia [22], mientras que para otros lo que se persigue es la combinación de los mensajes [18]. En cualquier caso, sea cual sea la perspectiva adoptada, el resultado hará prever la concisión y la simplicidad sintáctica para producir un texto coherente [8].

Se puede llevar a cabo la agregación de dos oraciones siguiendo distintos mecanismos. Se muestran a continuación algunos ejemplos sencillos:

- *Conjunción simple*: no cambia el contenido léxico ni sintáctico de los componentes.  
“Ana es de Murcia. Luis es de Albacete.”  
“Ana es de Murcia y Luis es de Albacete.”
- *Conjunción mediante componentes compartidos*: si se producen modificaciones, se busca que un elemento repetido aparezca una única vez:  
“Ana compró cebollas. Luis compró cebollas.”  
“Ana y Luis compraron cebollas.”
- *Inclusión*: desde la perspectiva lingüística, la forma más compleja de agregación, interviniendo oraciones subordinadas. En [18] aparece el siguiente ejemplo:  
“La casa está cerca del puente. La casa es bonita.”  
“La casa que está cerca del puente es bonita.”

- *Agregación léxica*: cuyo objetivo es expresar con un solo término el significado de un conjunto de términos. Se relaciona a su vez con la lexicalización, pues se debe elegir el componente léxico que sustituirá:  
“Reacciona con el fiúor, el cloro, el bromo y el yodo.”  
“Reacciona con los halógenos.”

En un sistema de *GLN* se seleccionarán los mecanismos adecuados para realizar estas tareas. Como en otras etapas del sistema, no se puede dejar de considerar aspectos como el perfil del usuario (puede requerir textos más o menos complejos), los requisitos del sistema (disponer de un espacio limitado favorece la concisión del texto), etc.

#### 4.2.2. Lexicalización

La lexicalización es la etapa de la *GLN* que se encarga de seleccionar las palabras específicas o estructuras sintácticas concretas con las que referirse al contenido seleccionado en fases anteriores.

Cuando se dispone de varias opciones deben considerarse aspectos como el conocimiento y preferencias de los usuarios, la consistencia tanto con el léxico ya empleado como con la historia del discurso o la relación con las tareas de agregación y Generación de Expresiones Referenciales (*GER*) (véase sección 4.2.3) propias de esta fase.

La variedad de opciones que se puede plantear para un mismo mensaje puede ser tanto sintáctica como semántica [81]. Si el mensaje que se quiere transmitir es la escasez de lluvia durante un mes, se dispone de múltiples posibilidades:

- *Variaciones en la categoría sintáctica*:  
Sujeto: “la lluvia fue muy pobre”  
Sintagma adjetival: “mucho más seco que la media”
- *Variaciones semánticas*:  
Valores absolutos: “muy seco, una lluvia muy pobre”  
Valores comparativos: “unas medidas muy por debajo de la media, mucho más seco que la media”

Cuando esto ocurre y se dispone de varias opciones, deben considerarse aspectos como el conocimiento y preferencias de los usuarios, el nivel de formalidad (p.ej. "padre" frente a "papá"), la consistencia tanto con el léxico ya empleado como con la historia del discurso (p.ej. se ha de expresar contraste, si ya aparece "*sin embargo*", se utiliza "*no obstante*") o la relación con las tareas de agregación y *GER* propias de esta fase de micro planificación.

#### 4.2.3. Generación de expresiones referenciales

En determinados contextos lingüísticos, la selección de una representación u otra para un concepto puede generar ambigüedad. Dentro del discurso se debe poder diferenciar las entidades y encontrar las características particulares que contribuyan a satisfacer los objetivos comunicativos. Determinar el modo en que se hace referencia a las entidades y conceptos que forman parte del plan del documento para evitar que se produzca ambigüedad es la función de la *GER*. Así, siguiendo a [81], se asumirá que las expresiones referenciales deben incluir la información que nos permite una identificaciónívoca de un referente en el contexto del discurso, evitando redundancias o exceso de información.

La definición del problema que ocupa a la *GER* es una de las más consensuadas en el ámbito de la *GLN*.

La arquitectura que se está tomando como referencia en este artículo es secuencial. Esto implica que la *GER* debe llevarse a cabo a partir de los contenidos seleccionados en la primera fase del sistema. Por tanto, cuando se produzca la selección, ésta debe adelantar lo que vaya a necesitar la *GER*, considerando que la forma que adopta la descripción de una entidad depende del lugar que ocupa en el contexto del discurso. En este sentido, es habitual la distinción que se lleva a cabo entre la primera alusión a una entidad en el discurso (referencia inicial) y cualquier otra referencia en el resto del discurso, que habrá de tomar en cuenta lo dicho hasta ese momento. Existe la posibilidad de

que sea la *GER* la que solicite al selector del contenido lo que necesite para construir una descripción adecuada. Esto implicaría una comunicación bidireccional que no es posible en la arquitectura secuencial aquí propuesta, sin embargo, sí lo es desde otras perspectivas, tal y como se presenta en trabajos como [42], en el que se plantea la generación de descripciones guiada por la *GER* frente a la guiada por la selección del contenido.

### 4.3. Realización

Las tareas asociadas al módulo de realización van a tener como objetivo final generar las oraciones reales que formarán parte de la salida del sistema, así como su estructura y formato dependiendo de los requisitos de la aplicación que contenga el módulo. La sintaxis, la morfología y la ortografía son aspectos que se trabajan en las fases de este módulo, que emitirá finalmente un texto gramaticalmente correcto y susceptible de recibir el postprocesamiento que le otorgue el formato precisado.

La entrada sobre la que trabaja esta fase es la especificación del texto o plan del discurso, producida por el módulo de micro planificación, que es un conjunto de especificaciones relativas a las oraciones y a su estructura en el discurso final. Se puede pensar en el módulo de realización como el conjunto de tareas que van a traducir esas especificaciones en la salida que un usuario va a recibir.

Con el objetivo de diferenciar la etapa que convierte las especificaciones en oraciones y la que les da un formato, se distinguen dos subtareas dentro de esta fase final: la realización lingüística y la realización de la estructura.

#### 4.3.1. Realización lingüística

La realización lingüística va a determinar el modo en que las representaciones abstractas de las oraciones se convierten en texto real.

En la figura 7 aparece la entrada para una herramienta llamada *RealPro*<sup>4</sup> [59]. El resultado en

<sup>4</sup>*RealPRO* es una herramienta empleada en *GLN* centrada en el ámbito de la realización. Se analizará más ampliamente en la sección 6.1.

este caso sería la oración “Marzo tuvo algunos días lluviosos”.

```
blow [class:verb mode:imperative]
(I you [class:pronoun]
 II nose [class:common-noun number:sing]
 (ATTR you [class:pronoun case:genitive]))
```

Fig. 7. Representación sintáctica para *RealPRO*

#### 4.3.2. Realización de la estructura

El último paso del proceso de *GLN* está totalmente condicionado por la aplicación. En este punto el resultado de los pasos anteriores recibirá un formato para aparecer en un medio determinado. Puede ocurrir que se vaya a mostrar en una página web y precise de etiquetas *HTML* o que deba convertirse en voz en un diálogo con un usuario. Son solo dos ejemplos pero hay múltiples posibilidades. Por tanto, son propias de esta etapa acciones como la inclusión de marcas en el documento (*HTML*, *LATEX*, *RTF*, *SABLE*<sup>5</sup>) o la creación de un árbol que incluya atributos propios del recipiente final (puntuación, viñetas, etc...). Un ejemplo de este tipo de árbol se muestra en la figura 8.

Respecto a la realización de la presentación mediante marcas, en la figura 9 se puede contrastar cómo se definiría para *HTML* y cómo para *LATEX*:

Un ejemplo de sistema con salida multimodo en el que se pone de manifiesto la necesidad de una etapa como ésta es *MATCH* [46], una aplicación que hace recomendaciones y proporciona información sobre restaurantes en la ciudad de Nueva York. En este sistema, la realización de la estructura facilita el geoposicionamiento de los locales en un mapa y determina el formato de texto y la voz. Frente a una petición como “*Show me Italian restaurants in the West Village*”, en primer lugar mostrará el mapa con las ubicaciones. El objetivo del usuario es decidir entre las opciones que le proporciona, de modo que solicita al sistema: “*Compare these restaurants*” trazando un círculo

<sup>5</sup> *SABLE* es un lenguaje de marcado *XML* empleado para anotar texto cuyo objetivo es la síntesis de voz.

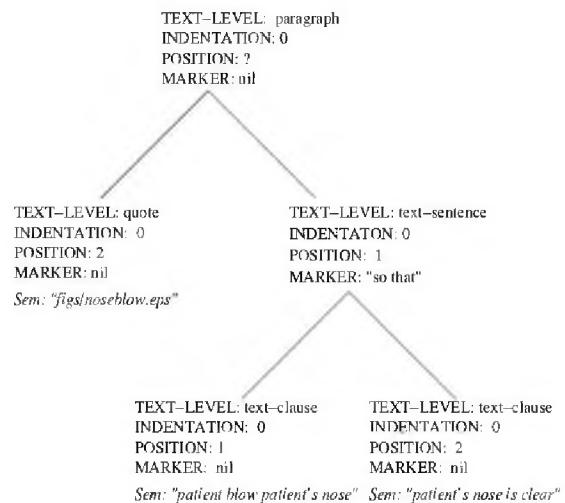


Fig. 8. Indicaciones para generar la salida [69]

<p>	(list introduction)
<ul>	\begin{itemize}
<li> (first list element)	\item (first list element)
<li> (second list element)	\item (second list element)
</ul>	\end{itemize}
</p>	

Fig. 9. Estructuras lógicas para web(*HTML*) y para *LATEX*

alrededor de los restaurantes sugeridos, tal y como aparece en la figura 10. De nuevo *MATCH* genera una salida, pero esta vez con forma de texto/voz: “*Compare-A: Among the selected restaurants, the following offer exceptional overall value. Uguale's price is 33 dollars. It has excellent food quality and good decor. Da Andrea's price is 28 dollars. It has very good food quality and good decor. John's Pizzeria's price is 20 dollars. It has very good food quality and mediocre decor*”.

#### 4.4. Ejemplos del proceso de generación

De acuerdo con la arquitectura presentada en esta sección, consistente en las fases de macro planificación, micro planificación y realización, se exponen a continuación dos ejemplos del proceso de generación implementados por sendos sistemas de *GLN* que plantean la secuenciación de



**Fig. 10.** Resultado del sistema *MATCH* para localizar restaurantes en West Village y solicitud de comparación

tareas desde puntos de vista diferentes. Por una parte, en *BabyTALK* [34] se emplea para el proceso de *GLN* la arquitectura de fases tal y como se ha descrito previamente. Por otra parte, en *PESCaDO* [100] el proceso se reduce a dos etapas, una similar a la macro planificación y otra que realiza el resto de generación lingüística.

En la sección 6.1 se revisarán otras herramientas empleadas en *GLN* así como sistemas que acometen subtareas de las fases.

#### 4.4.1. BabyTALK

*BabyTALK* es un sistema que genera informes sobre el estado de pacientes neonatales dependiendo del tipo de usuario al que van dirigidos (médicos, enfermeros, padres, etc.). Este sistema está basado en la arquitectura de tres módulos explicada anteriormente con el añadido de dos fases de preprocesamiento de datos. Esto es así debido a que la entrada al sistema, como puede verse en la figura 11, procede de un conjunto de canales que hacen llegar información fisiológica del paciente neonatal, como el ritmo cardiaco, la presión de la sangre o la temperatura.

En la primera fase de preprocesamiento de datos, se realiza un análisis de la señal de entrada que dará lugar a la identificación de eventos médica mente significativos y a patrones o tendencias a corto y largo plazo.

En la fase de interpretación de datos, partiendo de la identificación obtenida en la fase anterior

(figura 11), se agrupan un conjunto de eventos comunes a uno de más alto nivel y se interpretan. Como resultado de esta fase, se obtiene el conjunto de eventos interpretados.

Estos eventos son tomados por la fase de macro planificación y son posteriormente ordenados formando un árbol de eventos, siendo éste la salida de la fase.

Finalmente, en este sistema, se realiza conjuntamente las fases de micro planificación y realización, donde, primero se convierte el árbol de eventos de la fase anterior en una estructura de evento a la que se le irán añadiendo diversos conceptos como la vinculación de los eventos entre sí, y finalmente, partiendo de esta estructura se genera el texto final. La salida final para el ejemplo visto en la figura 11 es:

*You saw the baby between 16:40 and 17:25. Heart Rate (HR) = 155. Core Temperature (T1) = 36.9. Peripheral Temperature (T2) = 36.6. Transcutaneous Oxygen (TcPO2) = 9.0. Transcutaneous CO2 (TcPCO2) = 7.4. Oxygen Saturation (SaO2) = 94. Over the next 24 minutes there were a number of successive desaturations down to 0. Fraction of Inspired Oxygen (FIO2) was raised to 100 %. There were 3 successive bradycardias down to 69. Neopuff ventilation was given to the baby a number of times. The baby was re-intubated successfully. The baby was resuscitated. The baby had bruised skin. Blood gas results received at 16:45 showed that PH = 7.3, PO2 = 5, PCO2 = 6.9 and BE = -0.7. At 17:15 FIO2 was lowered to 33 %. TcPO2 had rapidly decreased to 8.8. Previously T1 had rapidly increased to 35.0.*

#### 4.4.2. PESCaDO

*PESCaDO* (*Personalised environmental service configuration and delivery orchestration*) es un proyecto desarrollado para ofrecer información medioambiental personalizada según el perfil de usuario, sus preferencias y su ubicación. Genera un informe partiendo de una base de conocimiento medioambiental básico que combina con datos procedentes de servicios web así como información relativa a otros perfiles de usuario. Esto le permite realizar las tareas de selección de contenido y

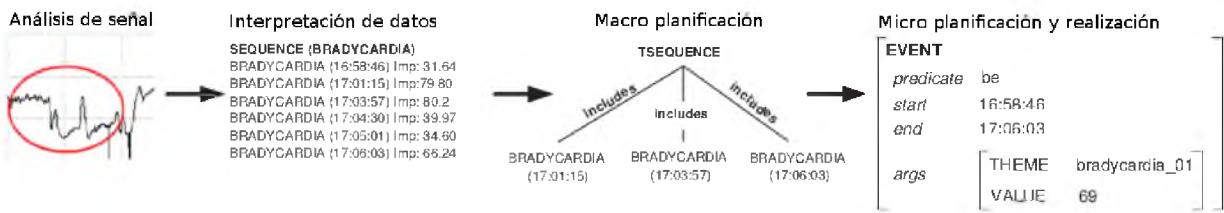


Fig. 11. Salidas y entradas de las fases en BabyTALK

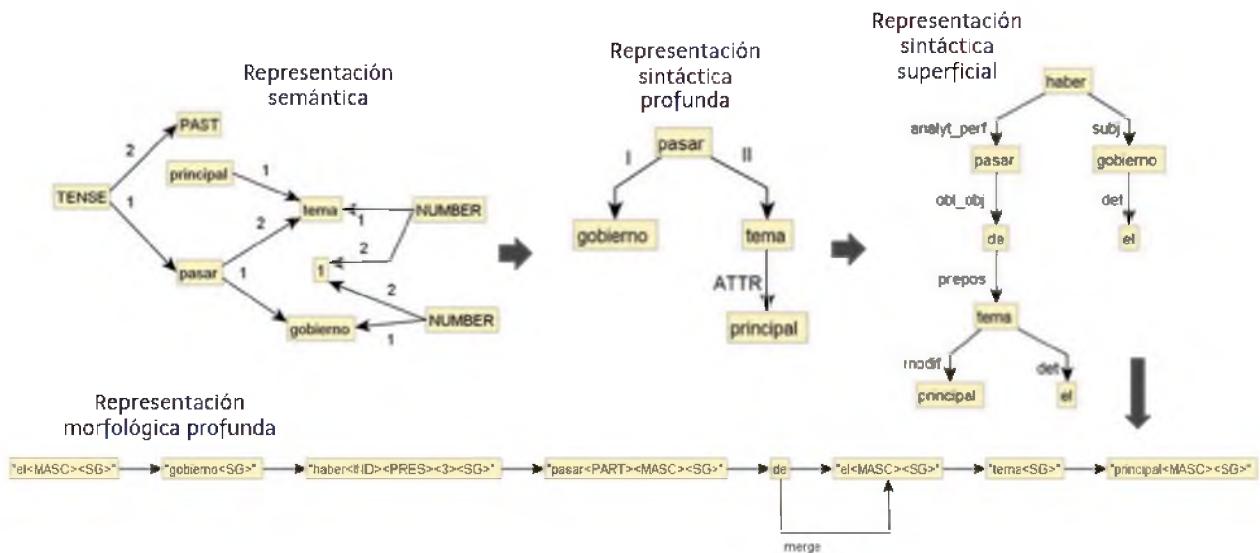


Fig. 12. Diferentes estratos de transformación en un sistema que se fundamenta en la MTT

de evaluación de calidad de los datos apoyándose en información relevante y actualizada.

Respecto a la fase de macro planificación, la selección de contenido se lleva a cabo a partir de la solicitud del usuario que da lugar a la población de una ontología multinivel. Los nodos que la constituyan serán agrupados por temas, de modo que se puedan extraer mensajes como unidades elementales del discurso susceptibles de ser asociados a un conjunto de esquemas que determinarán la estructura del mismo.

El resto del procesamiento no se ciñe exactamente a la arquitectura enunciada por Reiter, aunque sí lleva a cabo un proceso de generación lingüística a partir del plan del documento obtenido. Se introduce este sistema para contrastar su implementación con la de otros sistemas más

cercanos al modelo clásico. En el presente caso, el módulo encargado de transformar aquella estructura abstracta en una salida adecuada toma como base teórica la MTT (véase la sección 3.1). El proceso consiste en mapear las diferentes estructuras lingüísticas adyacentes empleando un conjunto de reglas de transición para cada nivel de transformación, mediante una herramienta llamada MATE [99].

La diferencia más evidente respecto a sistemas que implementan los tres módulos es que según este planteamiento, el proceso de generación se lleva a cabo en dos fases, la que genera el plan del documento y la que lo realiza empleando herramientas lingüísticas o técnicas estocásticas.

En la figura 12 se puede observar un ejemplo de las diferentes estructuras formando parte de un

proceso de transformación que lleva desde el plan del contenido, que aparece como una representación de mensajes semánticos, hasta la linearización y morfologización que produce finalmente una oración del informe.

En cuanto a la entrada y a la salida del sistema, las primera se lleva a cabo a través de una interfaz interactiva, introduciendo datos como la fecha sobre la que se requiere el informe, el lugar o algún tipo de dato medioambiental específico. Un ejemplo de salida producido por el sistema en inglés se muestra en la figura 13.

## 5. Técnicas aplicadas a las fases

En esta sección se describirán algunas de las técnicas empleadas en *GLN* considerando las fases en las que se utilizan comúnmente. En ocasiones se pondrá de manifiesto que esa relación no está estrictamente definida y que a la hora de diseñar el sistema, en ese sentido, la aplicación de las técnicas es bastante flexible. Se ofrecerá una descripción general de las técnicas así como ejemplos concretos de sistemas que las emplean, señalando su relación con los enfoques estadísticos y basados en conocimiento que se plantearon en la sección 3.

### 5.1. Técnicas de macro planificación

La macro planificación es la fase inicial de un sistema de *GLN*. Como se indicó anteriormente, en esta fase se lleva a cabo tanto la selección del contenido como la estructuración del documento.

Se ha aludido a una posible clasificación general de las técnicas asociadas a esta fase atendiendo a si la selección de contenido antecede o sucede a la creación de la estructura. Como resultado, se habla de un *enfoque de arriba a abajo* cuando se dispone de la estructura y se selecciona el contenido para completarla. En sentido contrario, se habla de *enfoques de abajo a arriba* cuando se dispone de los mensajes, de las unidades informativas, y es su composición la que da lugar a la estructura del documento [8]. Concretando un poco más los mecanismos empleados, *McDonald* en [66] propone una triple división de las técnicas.

El primer mecanismo que plantea *McDonald* es la planificación por refinamiento progresivo del mensaje. La esencia del refinamiento progresivo es que un módulo encargado de planificar el documento añade información adicional a un esqueleto básico. En este mecanismo, también llamado “*reemplazamiento directo*”, se empieza con una estructura de datos que gradualmente se transforma en un texto. La coherencia semántica del texto final procede de la coherencia semántica presente en la estructura original. El problema principal de este mecanismo es su carencia de flexibilidad, dado que una vez se ha generado la estructura, no es posible modificarla si en el proceso posterior se requieren variaciones.

El segundo mecanismo que plantea *McDonald* es la planificación usando operadores retóricos. Los operadores o predicados retóricos son los precedentes de la *RST* (véase sección 3.1), y también establecen relaciones retóricas entre los elementos sobre los que se ha de generar el texto. El procedimiento parte de un análisis de los objetivos comunicativos y consiste en la expansión del objetivo principal para alcanzar una estructura de árbol jerárquica en la que los nodos terminales sean las proposiciones y los operadores las reglas de derivación del mismo.

Por último, se presenta la técnica de los esquemas de texto. El origen de este mecanismo se remonta al trabajo de [67] quien lo propuso y empleó en un sistema llamado *TEXT*. La autora detectó regularidades en la forma de construir textos, tras haber analizado un gran número de ellos. Lo que observó es que, dado un objetivo comunicativo, se tendía a transmitir la misma información y en el mismo orden. Para reflejar tal peculiaridad, acuñó el concepto de “*schemata*” y lo combinó con el uso de predicados retóricos. Los *schemata* o esquemas determinan las posibles combinaciones de predicados, formando patrones o plantillas. Por tanto, a partir de un objetivo como *describir* o *comparar*, el sistema era capaz de seleccionar un esquema que proporcionaba un plan, indicándole cuándo y qué cabía decir.

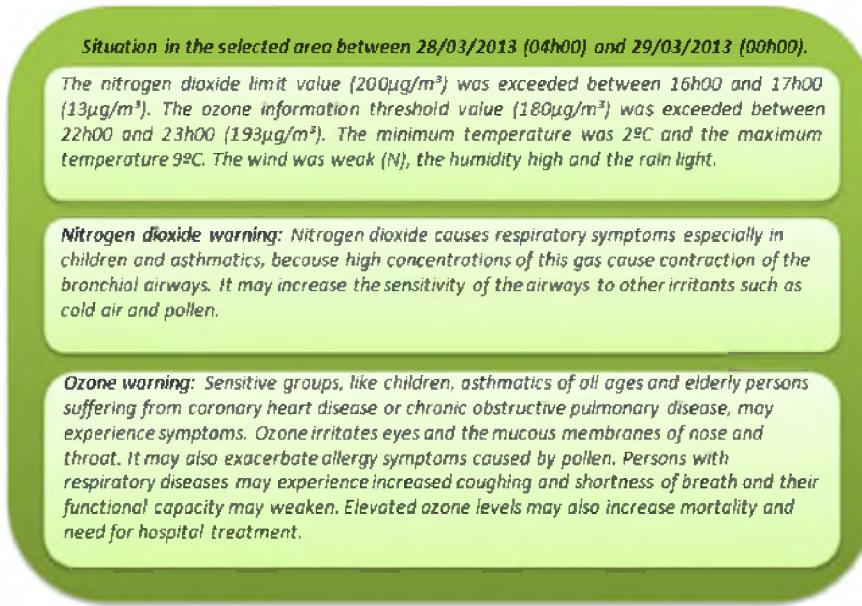


Fig. 13. Un ejemplo de la salida de PESCaDO [45]

## 5.2. Técnicas de micro planificación

La fase de micro planificación lleva a cabo las tareas de agregación de mensajes, lexicalización y generación de expresiones referenciales.

La agregación consistiría, según lo visto anteriormente, en la combinación de mensajes con el propósito de reducir la redundancia y aportar coherencia y fiudez al texto. Esta tarea generalmente requiere de una serie de reglas de composición mediante las que genera un conjunto de salidas posibles, por lo que necesita, cuando ése es el caso, de una función de selección. Se exponen a continuación diversos modos de afrontar el proceso.

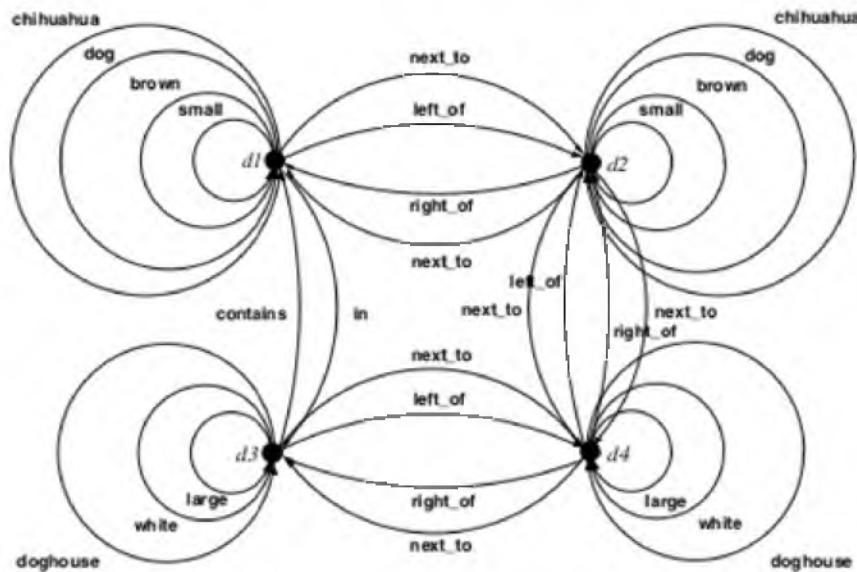
Como se ha indicado anteriormente, algunos sistemas generan varias alternativas pero también los hay que aportan una salida única a partir del conjunto de reglas y las unidades de información. Es lo que ocurre, por ejemplo, en el sistema ASTROGEN<sup>6</sup> [22], escrito en PROLOG en 1996. Frente a éste, SPOT [97] es un sistema capaz de seleccionar la opción más adecuada después de

generar varias posibilidades. El sistema incorpora técnicas de aprendizaje, de modo que las diferentes *agregaciones* son puntuadas mediante una función aprendida a partir de un corpus anotado. También hay sistemas en los que se emplean *algoritmos evolutivos*<sup>7</sup> [17], [41]. Se pueden encontrar trabajos que se basan en el recorrido de árboles de dependencias y en la RST [92] y, entre los planteamientos más actuales, aquellos que interpretan la agregación como un problema de particionado de hipergrafos [5].

La siguiente tarea que se suele incluir en la micro planificación es la lexicalización. Simplificando mucho, esta etapa consistía en asociar palabras (verbos, nombres, adjetivos) a los conceptos que producen las fases anteriores. De nuevo las técnicas son muy variadas. La lexicalización basada en plantillas asocia directamente una producción a un mensaje. SUMGEN-W, por ejemplo, selecciona el texto "very much warmer than average"

<sup>6</sup><http://people.dsv.su.se/~hercules/ASTROGEN/ASTROGEN.html>

<sup>7</sup>Los algoritmos evolutivos son un conjunto de técnicas inspiradas en los procesos evolutivos biológicos. Se aplican sobre una población de individuos que sufren una serie de transformaciones, como *mutaciones* o *cruces*, cuyos resultados son evaluados mediante una *función de aptitud* que proporciona finalmente el ejemplar adecuado.



**Fig. 14.** Grafo representando una escena de un perro que se encuentra cerca de una caseta. Algoritmo basado en grafos para resolución de la fase *GER*

cuando la temperatura media se encuentra en el intervalo [2.0 .. 2.9]. Otro tipo de sistemas emplean diccionarios o tesauros de los que toman sinónimos para elegir los términos adecuados [27] u otras bases de conocimiento como *WordNet* [72] o *FrameNet* [11] para realizar inferencias más complejas. Se pueden crear reglas explícitas o, en esa línea, árboles de decisión. Por ejemplo, si se ha de expresar un periodo entre dos fechas, se determina por su comparación si el mensaje de salida será “entre el *i* y el *j*” o “el *i* y el *j*”, si *i* es el día inmediatamente anterior a *j* (también en *SUMGEN-W*). Pero este tipo de planteamiento restringe mucho las posibilidades expresivas. Los métodos estocásticos, por otro lado, basados en corpora, mejoran en ese aspecto pero, por otro lado, hacen depender sus resultados tanto del tipo como de la cantidad de textos que componen el corpus que se emplee. Un ejemplo de este tipo de sistemas se encuentra en [2], donde se desarrolló un generador estadístico que, como parte del procesamiento general, es capaz de realizar la selección de los términos correspondientes a un conjunto de representaciones semánticas mediante clasificadores (Support Vector Machines), tomando como base el *AnCora-UPF* treebank [70].

Debido a la aparición de cierto tipo de competiciones o *desafíos* (que se verán más detenidamente en la sección 7.4), la tercera etapa de la micro planificación, la *GER*, ha sido una de las más desarrolladas estos últimos años. En esta etapa el objetivo es producir las descripciones o expresiones referenciales que permitan identificar una entidad del discurso. En principio, tal identificación debe ser unívoca, pero se han desarrollado algunas técnicas que relajan este requisito, dado que tales niveles de exigencia no siempre son necesarios [32]. Se dice, en ese caso, que se generan expresiones referenciales *simples*. Sin embargo, no es la norma común, de modo que aquí se verán los algoritmos o métodos que sí buscan esa univocidad de la expresión.

Uno de los mecanismos más utilizados es el *algoritmo incremental* [20] o alguna de sus variantes, como el algoritmo sensible al contexto de *Krahmer y Theune* [53]. La entrada de este algoritmo la conforma tanto la entidad que ha de ser referenciada como un conjunto de entidades denominado como *conjunto de contraste*. Todas las entidades tienen atributos asociados. Se itera sobre una lista de atributos, entre los que se seleccionan para formar la expresión final aquellos que descarten entidades

del *conjunto de contraste*. Sus autores crearon este algoritmo con la intención de que los resultados fueran similares a los que produciría un humano en el mismo contexto, habiendo considerado para su generación anteriores resultados relacionados con evidencias psicolingüísticas y análisis de diálogos.

También es muy conocido y empleado el *algoritmo basado en grafos* [54]. El algoritmo trabaja sobre un grafo dirigido mediante el que se representa una escena. A partir del grafo genera las expresiones referenciales. Tal y como se observa en la figura 14, las aristas del grafo están etiquetadas con propiedades y atributos. La idea básica sobre la que se diseñó el algoritmo es que es posible generar sistemáticamente todos los subgrafos de un grafo dirigido. Empezando por el subgrafo que solo contiene el vértice representando al objeto a referenciar, se lleva a cabo la expansión recursiva del mismo, añadiendo las aristas adyacentes al subgrafo que esté activo en el momento. Este procedimiento permite garantizar que el resultado será un subgrafo conexo. Finalmente, el algoritmo proporciona, si existe, el subgrafo distintivo de menor coste computacional que hace referencia al vértice del objeto a referenciar.

Por último, destacar que se han llevando a cabo una serie de trabajos que realizan la tarea desde enfoques estadísticos, motivados por aquellos desafíos mencionados anteriormente, y para ello emplean corpus de entrenamiento. En [94] encontramos una revisión del algoritmo basado en grafos incorporando en el proceso un modelo generado a partir de un corpus de descripciones, mientras que *mCRISP* es un sistema que realiza la generación de las expresiones empleando clasificadores, que se han obtenido también a partir de corpus [33].

### 5.3. Técnicas de realización

Uno de los primeros trabajos que presentaron técnicas estadísticas basadas en corpus para llevar a cabo la fase de realización fue el elaborado por *Langkilde y Knight* en 1998 [57]. El sistema utilizaba este enfoque también en las fases anteriores. Empleaba un modelo de n-gramas (véase

sección 3.2) que, en lo relativo a la etapa de realización, determinaba las transformaciones de palabras (usar el plural o no, el género,...). Se seleccionaban aquellas producciones cuya probabilidad fuera mayor.

Otro de los planteamientos dirigidos a resolver la fase de realización consiste en la aplicación de un algoritmo de *Spanning tree*<sup>8</sup> [98]. El procedimiento parte de un conjunto de palabras que constituyen un grafo. Esta técnica consigue convertirlo en un grafo acíclico que se recorre con un algoritmo voraz con el fin de ordenar los nodos hermanos usando un *LM* de n-gramas. De este modo se obtienen las distintas estructuras de árbol posibles. Para completar el proceso, se emplea un modelo de satisfacción argumental mediante el que se seleccionará la estructura que mejor ordene gramaticalmente las palabras, permitiendo la generación del texto final.

Considerando otra perspectiva, la fase de realización ha sido interpretada desde la *MTT* (véase la sección 3.1) como un paso final en una secuencia de transformaciones realizadas sobre representaciones lingüísticas. Tal transformación puede ser abordada tanto mediante gramáticas o reglas que permiten la traducción de grafos [99], como mediante el empleo de métodos estadísticos. Tal es el caso de un sistema nombrado más arriba, el planteado en [2], que mantiene esta filosofía en cada una de las transformaciones.

## 6. Herramientas y corpus existentes

En la sección 4 se han descrito las diferentes fases que constituyen un sistema de *GLN* genérico y en la sección 5 se han revisado las técnicas más frecuentes que se utilizan para abordar cada fase. A continuación, se van a analizar una serie de herramientas que se emplean en esas fases.

Existen muchas herramientas y aplicaciones gratuitas en la Web, aunque sobre todo se encuentran orientadas a la fase de realización. Numerosos sistemas y recursos están disponibles en la web de *Bateman* y *Zock*<sup>9</sup>, y documentación abundante relativa a técnicas, teoría subyacente,

<sup>8</sup><http://www.cisco.com/c/en/us/tech/lan-switching/spanning-tree-protocol/index.html>

<sup>9</sup><http://www.nlg-wiki.org/systems>

evaluación y nuevos retos se puede encontrar en las actas tanto del *International NLG Conference* como del *European NLG workshop*. Ambas conferencias son de carácter internacional, la primera a nivel mundial y la segunda a nivel europeo. La *INLG* celebró en 2014 su octava edición y publicó 28 de los 36 artículos que recibió (tres de ellos como demos). La conferencia europea, el *ENLG*, se celebra desde 1987 y su última edición fue la número 15, en septiembre de 2015.

### 6.1. Herramientas de *GLN*

Se exponen a continuación una serie de herramientas que han sido seleccionadas ya por su relevancia (*SimpleNLG*), ya por su actualidad (*NaturaOWL*) o por estar asociadas a cada una de las fases tal y como han sido abordadas en el presente trabajo. Éstas serán las primeras entonces: *SPUR* en el contexto de la macro planificación, *SPARKY* en micro planificación y *RealPRO* como la herramienta empleada en la fase de realización. Se puede ver un resumen de las mismas en la tabla 3.

- *SPUR* [96] se encarga de la fase de macro planificación. La salida que proporciona es un plan de contenido. La herramienta se emplea en un sistema que recomienda o compara objetos, y esto determina la estructura que produce. Los atributos que se incluyen en la salida son los más relevantes para el usuario, partiendo de sus preferencias y la solicitud que realiza. Dependiendo del entrenamiento, *SPUR* puede producir diferentes planificaciones.
- *SPARKY* [96] se encarga de la fase de planificación de oraciones basándose en un sistema de plantillas. El proceso que realiza está dividido en dos fases o módulos:
  - *Sentence Plan Generator (SPG)*. En una primera fase, se genera un conjunto de árboles de planificación que contienen las relaciones (nodos internos) y las aserciones (nodos hoja) que aparecen en el texto final. En la segunda fase, asigna esas aserciones a oraciones y las organiza.

- *Sentence Plan Ranker (SPR)*. Evalúa los distintos planes generados por el *SPG* basándose en un modelo basado en las valoraciones del usuario en la fase de entrenamiento.

- *RealPRO* [59] es una herramienta que ejecuta la fase de realización (véase sección 4.3). La entrada de la herramienta tiene que estar en formato *ASCII*, *HTML* o *RTF* por lo que se considera un sistema del tipo *D2T*. El procesamiento se realiza sobre una base de conocimiento lingüístico, inicialmente sólo para la lengua inglesa, pero ampliable a otras lenguas. Los datos de entrada se estructuran en un diagrama en forma de árbol de dependencias. Este diagrama tiene dos componentes:

- Relaciones sintácticas, que se representan con etiquetas en los arcos que relacionan los nodos.
- Lexemas, que se representan con una etiqueta en cada uno de los nodos. Además sólo se almacenan los lexemas que aportan significado. La herramienta no es capaz de realizar un análisis sintáctico, por lo que todos los lexemas que aportan significado tienen que estar especificados.

Una vez tenemos el árbol, la herramienta se encarga de añadir la palabras funcionales, generando así un segundo árbol. Con este segundo árbol, basándose en las etiquetas de los arcos, se crean reglas de precedencia linear que son utilizadas, posteriormente, para la conjugación de los elementos de la oración. Finalmente, se añaden los signos de puntuación y se generan las instrucciones necesarias para adaptar la salida al formato seleccionado.

- *SimpleNLG* [35], es una herramienta centrada en la fase de realización desarrollada exclusivamente para la lengua inglesa. Se puede encontrar en forma de biblioteca escrita en lenguaje java y su función es ayudar a escribir frases gramaticalmente correctas.

La herramienta se ha construido bajo tres principios básicos:

**Tabla 3.** Herramientas de GLN. \*Los sistemas marcados se han explicado en la sección 4.4, para ilustrar las diferentes fases del proceso de generación completo

Herramienta	Fases	Entrada	Salida
SPUR [96]	Macro planificación	Atributos que tiene el objeto que se va a comparar.	Plan de documento con los atributos más importantes.
SPARKY [96]	Micro planificación	Plan de documento (generado por SPUR).	Plan de discurso con las aseveraciones que se van a representar en la salida.
RealPRO [59]	Realización	Plan del discurso (nodos como lexemas y las aristas como relaciones sintácticas) (D2T)	Texto en forma de oración sintáctica y semánticamente correcta.
SimpleNLG [35]	Realización	Estructura sintáctica de la oración (D2T)	Oración sintáctica y semánticamente correcta.
NaturalOWL [1]	Macro planificación Micro planificación Realización	Ontología OWL (D2T)	Oración sintáctica y semánticamente correcta.
BabyTALK*	Macro planificación Micro planificación Realización	Señales procedentes de dispositivos médicos	Texto completo formado por oraciones sintáctica y semánticamente correctas.
PESCaDO*	Macro planificación Realización (que incluye procesos de Micro planificación)	Ontología multinivel y solicitud del usuario	Informe como discurso bien formado.

- Flexibilidad. *SimpleNLG* es una combinación de sistema *enlatado* (basado en esquemas) y sistema avanzado. Mediante la combinación de ambos, se consigue una mayor cobertura sintáctica.
- Robustez. Cuando una entrada está incompleta o es errónea la herramienta generará una salida, a pesar de que lo más probable es que no sea la esperada.
- Independencia. Las operaciones morfológicas y sintácticas están claramente diferenciadas y separadas.

La biblioteca proporciona una interfaz con la que interactuar desde el código java. A partir de un elemento base, que equivale al verbo principal de la oración, se van concatenando otros elementos que van a tomar parte en la acción principal. Una vez están los elementos agrupados, se indicará el tiempo verbal de la oración y la forma en la que se construye (interrogativo, infinitivo...). Finalmente, la herramienta genera una oración basándose en

los parámetros que se le han indicado. Un ejemplo de entrada y salida se encuentra en la tabla 4.

- *NaturalOWL* [1] es una herramienta *D2T* que a partir de una ontología (*OWL*) genera un texto con la información contenida en ésta. *OWL* es un estándar para especificar Ontologías en la Web Semántica. Para generar el texto, realiza las tres fases comentadas en la sección 4.

Para la fase de macro planificación, la herramienta recoge todas las declaraciones de la ontología que se consideran relevantes y las convierte a un formato más simple de expresar (tripletas). Posteriormente, selecciona qué tripletas van a ser mostradas en el texto. Cada una de las tripletas se intentan mostrar como una frase simple. Para ello se ordenan estas tripletas en lugar de las oraciones correspondientes. Esto es debido a que *NaturalOWL* no tiene en cuenta la coherencia global ya que la mayoría de las oraciones sólo aportan infor-

**Tabla 4.** Ejemplo de funcionamiento de *SimpleNLG*

Verbo	Parámetros	Modo, "Salida"
	tiempo = pasado	Interrogativa (where, object): "Where did the boys leave?"
"Leave"	objeto = "the house"	Interrogativa (sí, no): "Did the boys leave the home?"
	sujeto = "the boys"	

mación adicional al núcleo o como mucho a los núcleos de segundo nivel. Por esto último, no se utiliza la representación en forma de árbol que se suele usar en esta fase.

En la fase de micro planificación, *NaturalOWL* permite que el usuario configure el número máximo de frases a agregar. Generalmente los sistemas de *GLN* agregan el máximo de frases posibles para mejorar la legibilidad, pero esta herramienta permite configurar el número máximo de frases que se desean concatenar.

Por último, en la fase de realización, *NaturalOWL* toma la salida de la micro planificación y la representa añadiendo los símbolos de puntuación y las letras mayúsculas necesarias. La entrada de esta última fase, como en la mayoría de los sistemas basados en esquemas, contiene el formato y el orden final en el que van a aparecer cada una de las palabras en el texto final, por lo que no hace falta añadir nueva información, es más un proceso de transformación de los datos obtenidos al formato de salida. Un ejemplo de una entrada y salida en *NaturalOWL* queda ilustrado en la figura 15.

## 6.2. Corpus y conjuntos de datos

Para determinadas estrategias de desarrollo de sistemas de *GLN* es habitual el uso de corpus, ya sea intrínsecamente en el proceso de generación (sección 3) o extrínsecamente en el proceso de

CLASE	
Laptop, tecraA8	
PROPIEDADES	
manufacturedBy	toshiba
hasProcessor	intelCore2
hasMemoryInGB	2
hasHardDiskInGB	110
hasSpeedInGHz	2
hasPriceInEuro	850
SALIDA	
"Tecra A8 is a Laptop, manufactured by Toshiba. It has an Intel Core 2 processor, 2 gb ram and a 110 gb hard disk. Its speed is 2 ghz and it costs 850 Euro."	

**Fig. 15.** Ejemplo de funcionamiento de *NaturalOWL*

evaluación (sección 7.3). Los corpus empleados pueden estar etiquetados con información de diversa índole y serán seleccionados dependiendo de la tarea que se deba resolver, siendo distinto el tipo de información incluida en un corpus necesario en la etapa de selección de contenido del requerido para la etapa de *GER*. Es posible encontrar sistemas que utilizan corpus generales y sistemas que emplean corpus especialmente diseñados para *GLN*. Más concretamente, estos últimos, suelen ser creados *ad hoc* para una etapa de una aplicación concreta o en el seno de una competición que plantea resolver una tarea muy delimitada (véase sección 7.4).

En la Tabla 5 se expone un conjunto de corpus creados específicamente para la *GLN*. Se han clasificado atendiendo a las tareas descritas en la sección 4. Todos ellos están en inglés y la información que presentan es dispar. Por un lado,

<sup>10</sup><http://inf.abdn.ac.uk/research/sumtime>

<sup>11</sup><http://www.classic-project.org/>

<sup>12</sup><http://jetteviethen.net/research/spatial.html>.

<sup>13</sup><http://inf.abdn.ac.uk/research/tuna/corpus>

<sup>14</sup><http://www.pitt.edu/~coconut/coconut-corpus.html>

<sup>15</sup>[http://mcs.open.ac.uk/nlg/old\\_projects/pills/corpus/PIL/](http://mcs.open.ac.uk/nlg/old_projects/pills/corpus/PIL/)

**Tabla 5.** Corpus *GLN*

Corpus	Información	Tipo de info.	Tareas
SumTime <sup>10</sup>	Predicciones de parámetros meteorológicos	Datos ( <i>D2T</i> )	Selección de contenido y agregación
WOZ <sup>11</sup>	Selección de atributos de un restaurante de Edimburgo	Texto ( <i>T2T</i> )	Selección de contenido y agregación
GRE3D7 <sup>12</sup>	Expresiones referenciales sobre objetos en 3D	Datos ( <i>D2T</i> )	Generación de expresiones referenciales
TUNA <sup>13</sup>	Referencias a objetos en dominios visuales	Datos ( <i>D2T</i> )	Generación de expresiones referenciales
COCONUT <sup>14</sup>	Diálogos automatizados	Datos ( <i>D2T</i> )	Generación de expresiones referenciales
PIL <sup>15</sup>	Información de pacientes	Datos ( <i>D2T</i> )	Realización

los corpus orientados a selección de contenido y agregación contienen conjuntos de datos, tanto numéricos como textuales, de los que seleccionar la información para posteriormente procesarla. Los corpus orientados a generar expresiones referenciales contienen información sobre objetos reales o expresiones referenciales en sí. Por último, se incluye un corpus que se emplea en la fase de realización, siendo lo característico de este tipo de corpus que contiene datos estructurados.

## 7. Evaluación de la *GLN*

Si en algo están de acuerdo los autores que han trabajado el tema de la evaluación en *GLN* es en la dificultad que entraña tal tarea debido a sus peculiaridades [93]. Frente a otros sistemas desarrollados en la *LC*, en este ámbito la evaluación del sistema se llevará a cabo considerando que, para empezar, lo que deba ser la entrada, ya al sistema ya a los módulos, no está adecuadamente especificado, que la salida correcta no es única y que no hay un criterio definido que permita evaluar la bondad/calidad de la misma.

### 7.1. Tipos de evaluación

Cuando se evalúa un sistema de *GLN* se puede optar por diferentes estrategias [85]. Por un lado es posible evaluar el impacto que el sistema produce en los usuarios o en otras tareas. Se trata en este caso de una *evaluación extrínseca*, centrada en los efectos externos del sistema. Por otro lado,

se puede evaluar el rendimiento y eficacia del sistema en sí mismo, caso en el que se estaría realizando una *evaluación intrínseca*. También se distingue la *evaluación manual* de la *automática*. Generalmente la primera es más costosa y más difícil de organizar, e incluso puede tomar mucho tiempo completarla. Es usual que en la evaluación de los sistemas de *GLN* la evaluación extrínseca se realice manualmente y la intrínseca automáticamente [85]. Un ejemplo de evaluación extrínseca y manual fue la llevada a cabo para el sistema *STOP* (véase sección 2.2). Se utilizaron encuestas para supervisar la eficacia de la tarea: cuántos usuarios habían dejado de fumar, el tiempo que les llevó, ... La evaluación de *STOP* necesitó 20 meses y tuvo un coste de 75.000 libras [82]. Respecto a la evaluación intrínseca, la cual valora las propiedades del sistema sin considerar los efectos externos del mismo, suele llevarse a cabo comparando las salidas del sistema o de alguna de sus partes con textos de referencia o con corpus, empleando métricas o escalas de puntuación.

### 7.2. Aspectos relevantes en la evaluación de sistemas de *GLN*

En la tarea de evaluación de un sistema de *GLN* caben muchos aspectos que es preciso delimitar y definir. Se pueden considerar aspectos relacionados con el funcionamiento del sistema completo, por un lado, y con el funcionamiento de los módulos del sistema, por otro [79].

En cuanto a la evaluación del sistema, se tiene en cuenta la adecuación de la salida al objetivo comunicativo, a la historia del discurso o a la petición del usuario. También la cobertura sintáctica y la corrección del estilo, así como la coherencia, la ambigüedad y la calidad del vocabulario [7]. En ocasiones se mide el esfuerzo necesario para post-editar la salida o se realizan experimentos con los usuarios que tienen que completar pruebas de lectura y comprensión o puntuar las salidas, por ejemplo.

Respecto al funcionamiento de cada uno de los módulos o tareas desempeñadas por el sistema de *GLN*, cada etapa debe ser evaluada considerando sus responsabilidades propias [7]:

- *Selección del contenido*: calidad de la información mostrada
- *Estructuración del documento*: cohesión
- *Agregación de sentencias*: cohesión y redundancia
- *Lexicalización*: calidad y cobertura del vocabulario
- *Generación de expresiones referenciales*: calidad de la información, ambigüedad, redundancia
- *Realización lingüística*: cobertura sintáctica, fluidez, claridad
- *Realización de la estructura*: esfuerzo para post-editar la salida, legibilidad, claridad

### 7.3. Métricas para evaluar sistemas de *GLN*

Se ha comentado anteriormente la posibilidad de evaluar la salida de un sistema de *GLN* comparándola con un texto ideal creado por un experto o con un corpus de referencia, sea éste generado por humanos o por otros sistemas de *GLN*. La evaluación puede hacerse en este caso en términos cuantitativos empleando métricas realizadas sobre tales comparaciones. Las métricas que se suelen emplear proceden de otros ámbitos de la *LC* y se han adoptado porque tuvieron buenos resultados en sus campos respectivos. Este tipo de evaluación automática basada en corpus es

atractiva en *GLN*, del mismo modo que en otras ramas de la *LC*, por su velocidad, reproductibilidad y bajo precio [80].

Algunas de las métricas que se han utilizado en generación proceden del ámbito de la traducción, como *BLEU* o *NIST*. Otras, como *ROUGE*<sup>16</sup>, de la generación de resúmenes. *BLEU* [75] es una métrica de precisión empleada en traducción que evalúa la proporción de n-gramas que comparte la salida del sistema con varias traducciones. *NIST* [25] es una adaptación de *BLEU* que añade un peso a los n-gramas más informativos. También se utiliza en traducción la herramienta *METEOR*<sup>17</sup> [58], que más allá de cotejar los n-gramas, compara significados a partir de fuentes como *WordNet*<sup>18</sup> [72]. En el ámbito de los resúmenes, se usa la métrica *ROUGE*, con un funcionamiento paralelo a *BLEU* y diversas métricas como *ROUGE-1*, *ROUGE-2* o *ROUGE-SU4* [61].

Estas estrategias son discutidas en la comunidad por diversos factores: en sistemas de *GLN* no existe una única salida buena (*gold-standard*), que sí tiene sentido en otros campos de la *LC*. Tampoco existen muchos corpus específicos para la tarea, las métricas que se están empleando proceden de otras ramas de la *LC* [89] o se apunta a la dificultad de interpretar los resultados proporcionados por las métricas (qué implica una variación de los resultados, cómo comparar los resultados con la evaluación humana) [76]. Éstas son algunas de las razones que se aluden para desconfiar de este tipo de evaluación. Sin embargo, el tema se mantiene abierto y en constante revisión y encontramos estudios, como el llevado a cabo en [6], en el que tras emplear las métricas citadas sobre la salida de *SumTime* y cotejar los resultados con evaluaciones humanas, se concluye que es apropiado hacer uso de las métricas en *GLN*, pero bajo ciertas condiciones si se espera obtener buenos resultados: por un lado se precisa de corpus amplios y de alta calidad y, por otro, de métricas que permitan evaluar ciertos aspectos lingüísticos del texto que van más allá de la mera

<sup>16</sup><http://www.berouge.com>

<sup>17</sup><http://www.cs.cmu.edu/~alavie/METEOR/index.html>

<sup>18</sup>*WordNet* es una base de datos léxica del idioma Inglés <http://wordnet.princeton.edu/>

comparación de n-gramas, como la estructura de la información.

#### 7.4. Evaluación colaborativa y competiciones

Determinar el modo en que se evalúan los sistemas, adscritos a una determinada disciplina, se considera un aspecto de crucial importancia para que la investigación avance y la disciplina progrese. El debate al respecto en *GLN* se remonta a la última década del siglo XX, cuando comenzaba a diferenciarse del resto de áreas de la *LC* [68]. Sin embargo, ha sido en los últimos años cuando se ha realizado un mayor esfuerzo dirigido a definir la metodología adecuada en evaluación y cuando se han impulsado iniciativas para determinar marcos de referencia comunes y disponer espacios adecuados para discutir sobre la cuestión de la evaluación. Como respuesta al creciente interés en el tema, en 2006 se llevó a cabo una primera sesión especial en la Conferencia Internacional en Generación de Lenguaje Natural (*INLG'06 Special Session on Sharing Data and Comparative Evaluation*). Allí se sentaron las bases para nuevos proyectos centrados en la evaluación en *GLN*, anticipando la celebración de las siguientes reuniones<sup>19</sup>. Pero lo que marcó la diferencia con respecto al modo de orientar la evaluación de los sistemas fue la creación de una organización cuyo cometido sería fomentar competiciones relacionadas con diferentes tareas de los sistemas de *GLN*. El grupo en cuestión se denominó *Generation Challenges* y su labor ha dado lugar a lo que se conoce como *STECs* (*Shared task evaluation challenges*) en el entorno de la *LC*, es decir, trabajos de evaluación colaborativa a partir del planteamiento de un problema concreto, referido a una tarea de la *GLN*, cuya resolución han de afrontar varios equipos de trabajo, comparando finalmente los resultados obtenidos [93]. Los llamados *desafíos*.

Aunque la puesta en marcha de tales competiciones se remonte en la actualidad a unos pocos años, la variedad de los retos propuestos ha dado lugar a muy diversas convenciones. Se muestra en la tabla 6 una relación de algunos de ellos.

<sup>19</sup> *Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation 2007, Workshop: Using Corpora for Natural Language Generation + Evaluation 2011*

Sin embargo, no se debe perder de vista que el empleo de *STECs* en el ámbito de la *GLN* no ha dejado de discutirse desde el momento en que aparecieron en escena [21]. Debido a la naturaleza compleja de la disciplina, que se ha puesto de manifiesto a lo largo del artículo, se deben considerar cuestiones respecto al tipo de tareas que se hayan de evaluar mediante estos métodos, el tipo de métricas a utilizar, las bases metodológicas necesarias (tanto en cuanto al planteamiento de la competición como a la comparación de los resultados), etc. Y también se plantean alternativas. Según Walker [95], “Casi cualquier recurso compartido será bueno para el avance científico de la *GLN*”. No obstante, el autor defiende con mayor vehemencia que lo que propiciaría un verdadero avance en el campo sería la generación de recursos de calidad para cada módulo concreto, con sus interfaces claramente identificadas. En este sentido, realiza la siguiente propuesta: que sean los propios investigadores los que pongan a disposición de la comunidad los recursos utilizados en su trabajo con una especificación adecuada (aunque al mismo tiempo asume la dificultad de esta tarea por el coste asociado a publicar y mantener ese tipo de producto en el contexto de la actividad investigadora).

### 8. Conclusiones y prospectiva de la *GLN*

En este artículo se ha realizado una revisión exhaustiva de la tarea de *GLN*, abarcando desde las etapas que componen el proceso de generación, pasando por las técnicas utilizadas, corpus y herramientas existentes más relevantes, hasta las competiciones que se han organizado a lo largo de los años para poner en común los resultados obtenidos por la comunidad investigadora en este campo. De todo lo expuesto, en esta sección se van a extraer las conclusiones más importantes, así como las líneas futuras que la investigación en este campo seguirá con mayor probabilidad.

Entre los distintos sistemas de *GLN* existentes en el mercado se puede realizar una clasificación atendiendo bien al tipo de entrada o bien al tipo de objetivo que persigan. El primer grupo se puede

**Tabla 6.** Competiciones en el ámbito de la *GLN*. Estas tareas se desarrollan o bien enfocadas a resolver una fase/etapa concreta o bien a generar un sistema completo. Se indican las fases involucradas: macro planificación (Ma), micro planificación (Mi) y/o realización (Re)

Año	Competición	Objetivo	Tareas/Fases <i>GLN</i>
2013	<i>GIVE</i>	Construcción de un sistema que genere instrucciones de orientación y manipulación para que un usuario se mueva en un edificio virtual y consiga un trofeo.	Proceso completo [Mi]
2013	<i>GRUVE</i> Generación de instrucciones de orientación en entornos virtuales inciertos.	Construcción de un módulo que ayude a la navegación en Google Streetview.	Proceso completo [Mi]
2012	Primer desafío en selección de contenido desde la web semántica abierta	Construcción de un sistema que recupere, de un conjunto de triples RDF referidos a una celebridad, aquellos que aparecen implícitos en un texto objetivo.	Selección de contenido [Ma]
2012	<i>HOO</i> <i>Helping Our Own</i>	El propósito de esta tarea es mejorar la calidad de textos en inglés, sobretodo los escritos por hablantes no nativos.	Proceso completo [Ma+Mi+Re]
2012	<i>KBGen</i> Generación desde bases de conocimiento	Generación de una descripción coherente de entidades biológicas, procesos y conexiones entre ellos.	Proceso completo [Ma+Mi+Re]
2011	<i>GQ</i> <i>Question Generation Shared Task</i>	Generación de preguntas desde oraciones y desde párrafos tomando como fuentes la Wikipedia, OpenLearn y Yahoo! Answers.	Proceso completo [Ma+Mi+Re]
2011	<i>SR-11</i> Realización de la estructura	Generación de oraciones a partir de representaciones sintácticas y semánticas.	Proceso completo [Re]
2008	<i>REG-08</i> <i>Referring Expression Generation</i>	Selección de atributos y realización.	Generación de expresiones referenciales [Mi]
2007	<i>ASGRE</i> Selección de atributos para GER	Generación de un conjunto de atributos que identifiquen un referente dado, ya sea el conjunto mínimo de atributos que permitan distinguirlo o el que seleccionaría un humano.	Generación de expresiones referenciales [Mi]

dividir a su vez en dos, obedeciendo al tipo de entrada que tenga el sistema, un *texto* (*T2T*: *Text-to-Text*) o un conjunto de datos (*D2T*: *Data-to-Text*), ya sean estos datos numéricos, representaciones semánticas, grafos, etc.

En cuanto al segundo tipo de clasificación, según el objetivo del sistema, no existe un número acotado de clases debido a la gran cantidad de aplicaciones para las que se puede utilizar un sistema de *GLN*: generación de textos informativos, generación de resúmenes, generación de textos simplificados, generación de diálogos, etc. Así pues, parece que en un futuro los sistemas de

*GLN* van a seguir dependiendo en gran medida del tipo de datos que contenga la entrada al mismo y del objetivo final para el que se esté construyendo.

Por otro lado, a la hora de afrontar la construcción de un sistema *GLN* se puede hacer desde dos enfoques distintos, el basado en conocimiento o el basado en métodos estadísticos, incluso se podría realizar una aproximación híbrida. Dada la envergadura de los sistemas de *GLN* y la dificultad que atañe a cada una de las tareas involucradas, cada vez es más común afrontar dichas tareas utilizando las técnicas que mejoran resultados dan en cada una de ellas, o subtareas, por separado.

De esta manera, es cada vez más habitual ver sistemas híbridos que utilizan una técnica para una tarea y otra muy distinta para otra. De todas formas, existe aún un amplio margen de mejora para estos sistemas, por lo que se antoja que deberán de emplearse nuevas técnicas para afrontar las distintas problemáticas que plantean los sistemas de *GLN* actuales.

Tal y como pone de manifiesto este artículo, ni los distintos sistemas de *GLN* existentes en el mercado ni las herramientas utilizadas para llevar a cabo las diversas tareas dentro de cada uno de estos sistemas siguen una estructura específica, más bien afrontan tareas concretas que han sido especificadas por sus diseñadores. Pese a ello, desde un punto de vista más arquitectónico, en este artículo se han agrupado las distintas tareas realizadas en estos sistemas y herramientas de *GLN* en tres fases secuenciales basándose en el trabajo realizado en [81]. Estas tres fases son: la *macro planificación*, encargada de diseñar la estructura del texto creando un plan de documento, la *micro planificación*, encargada de la planificación del discurso seleccionando las palabras y las referencias adecuadas y, por último, la *realización*, que se encarga de generar la salida final del sistema con las oraciones que la conforman así como las flexiones, concordancia y disposición oportunas. Pero, como ya se ha comentado, no existe un consenso sobre las distintas fases que deben participar en la creación de un sistema de *GLN* ni se acota el cometido de cada una de estas. Incluso el propio *Reiter*, cuya obra ha determinado que en el presente artículo se plantee una arquitectura de tres fases, en [79], añade una nueva fase de preprocesado de la entrada al sistema que se ha de implementar bajo ciertas circunstancias. Esto indica que la estructura de los sistemas de *GLN* está en continua reconstrucción.

La consecución de los objetivos propios de cada fase requiere de la aplicación de ciertas técnicas. Siguiendo esa distribución de funciones, se expusieron las técnicas más habituales:

- *Macro planificación*. Siguiendo a *McDonald* [66], se plantearon tres enfoques distintos para abordar las tareas: refinamiento progresivo, operadores retóricos y esquemas de texto.

- *Micro planificación*. Una de las problemáticas que más frecuentemente se pretende solventar en esta fase es la de la generación de expresiones referenciales. Por otro lado, para la generación del texto en esta fase se suelen emplear autómatas finitos ponderados [91] y gramáticas probabilísticas libre de contexto [51].

- *Realización*. La técnica más recurrida, debido a su sencillez y la economización de los recursos, para la creación del texto final son los *LM*, aunque también es común la utilización de reglas y ontologías.

La incorporación de nuevas técnicas para resolver la problemática asociada a cada fase es constante. El trabajo en otras áreas de la *LC* y de la inteligencia artificial contribuye a la mejora de los recursos existentes, y los amplía, dado que en esos campos se emplean técnicas basadas en conocimiento y estadísticas similares a las utilizadas en la *GLN*.

Por otro lado, a la hora de diseñar e implementar un sistema de *GLN*, los mejores lugares para encontrar recursos de *GLN* son los portales Web de los dos grandes congresos en la materia: *International NLG Conference* y el *European NLG workshop*. A través de las publicaciones que generan es posible estar al tanto de las técnicas, los avances, las dificultades que enfrentar en el transcurso de la creación y prueba del nuevo sistema, etc. También es posible incluir en su generación herramientas disponibles en la web (la mayor parte de las que se ofrecen gratuitamente están asociadas a la fase de realización).

Al mencionar los enfoques estadísticos, se introdujo la idea de que tales técnicas empleaban modelos para realizar previsiones. Un modelo de lenguaje se obtiene, generalmente, de un *corpus*. Por otro lado, en la sección de evaluación también se mencionó la importancia de los mismos a la hora de medir la calidad de un sistema. En ese contexto, se puede considerar que tanto el diseño como el uso de *corpus* en la *GLN*, dependen de la tarea a la que se asocien. Los *corpus* orientados a la selección de contenido y agregación contienen conjuntos de datos, tanto numéricos como textuales, de los que seleccionar la información para posteriormente procesarla. Los *corpus* orientados

a generar expresiones referenciales contienen información sobre objetos reales o expresiones referenciales en sí. Por último, los corpus empleados en la fase de *realización* se caracterizan por contener datos estructurados. El número de corpus existentes y el tipo de estos irá creciendo en un futuro debido principalmente a las distintas competiciones existentes en la materia, en las que suelen ser los propios participantes los que ayudan a la creación de estos corpus.

Por último, en lo referente a la evaluación de los sistemas de *GLN* hay que destacar la dificultad de dicha tarea ya que ni la especificación de las entradas es definitiva ni existen salidas correctas de forma absoluta con las que comparar los resultados. La evaluación, por otro lado, puede ser intrínseca, considerando el funcionamiento interno del sistema o tarea, o extrínseca, atendiendo a los efectos externos que éste provoque, y debe considerar aspectos muy variados como la adecuación de la salida al objetivo comunicativo, a la historia del discurso o a la petición del usuario, la cobertura sintáctica y la corrección del estilo, etc. En cuanto a las distintas métricas que se suelen utilizar para evaluar un sistema de *GLN*, muchas de estas proceden del ámbito de la traducción u otros campos de la *LC*, entre las más comunes se pueden encontrar *BLEU*, *NIST* o *ROUGE*. Parece, sin embargo, que la tendencia en este ámbito se dirige hacia la evaluación colaborativa que se suele llevar a cabo a partir de competiciones o *workshops* y que se plantean bien para solventar un problema concreto, bien para evaluar un sistema completo. Para ello se insta a los participantes a generar recursos de calidad para cada módulo o tarea concreta.

A pesar de la dificultad que entraña la *GLN* y de la falta de consenso que en ocasiones se manifiesta en la comunidad investigadora, tanto la aplicabilidad como el potencial que comporta su investigación lo sitúan en un punto de desarrollo muy fructífero en el momento actual. Más aun, si bien es cierta la disparidad de enfoques, también lo es el constante intercambio de puntos de vista para tratar de estandarizar o acercar las diferentes perspectivas. Un indicador de esta tendencia es la proliferación de congresos y talleres en los que los investigadores y las empresas ponen en común la

evolución de sus trabajos y proyectos. Por ejemplo, los recientes *1st International Workshop on Data-to-Text Generation*<sup>20</sup> y *1st International Workshop on Natural Language Generation from the Semantic Web*<sup>21</sup>, celebrados a lo largo de 2015 en Edimburgo y en Nancy, respectivamente.

Lo que revela esta exposición es que, aunque los inicios de la disciplina como tal se remonten a mediados del siglo XX, queda mucho trabajo por realizar para conseguir que se normalice su aplicación a todos los ámbitos de la actividad humana (dado que en ese sentido se estaría cumpliendo su potencial, en tanto interfaz humano-máquina).

De modo que, por un lado, se hace manifiesta la necesidad de profundizar en la investigación de técnicas híbridas que permitan abordar las distintas fases de manera eficaz y eficiente desde un punto de vista computacional, haciendo especial hincapié en investigar y proponer arquitecturas genéricas que puedan llegar a ser independientes del dominio y de la aplicación. Las técnicas híbridas son aquellas que se basan tanto en estadística como en bases de conocimiento lingüístico. En ese sentido, y en cuanto a los recursos empleados en el procesamiento, ya sea asociados a una u otra vertiente (corpus anotados, gramáticas, diccionarios,...), hay una fuerte tendencia a incorporar consideraciones pragmáticas y semánticas bajo el supuesto de que tanto el significado del discurso que conforma la salida de un sistema de *GLN*, como el lenguaje en tanto instrumento de comunicación, únicamente adquiere su correcto significado cuando se sustenta sobre tal entramado de conocimiento [15].

Por otro lado, es patente la necesidad de establecer estándares que permitan una mayor comunicación y participación entre sistemas, o que favorezcan la evaluación de las diferentes propuestas de modo que la investigación pueda avanzar a otro ritmo. Hay varios proyectos que desarrollan esta línea de acción y que facilitan estructuras susceptibles de ser compartidas. Es el caso de *NIF* [40] o de *OLIA* [19], ambas vinculadas a la integración de datos a través de ontologías.

A medida que la comunidad investigadora vaya avanzando en el desarrollo de nuevos enfoques,

<sup>20</sup><http://www.macs.hw.ac.uk/InteractionLab/d2t/>

<sup>21</sup><http://www.loria.fr/~gardent/WebNLG2015/Home.html>

estos serán integrados cada vez más en aplicaciones reales orientadas a mejorar la interacción humano-máquina y viceversa. El resultado permitirá una comunicación más fluida entre estos actores, contribuyendo al desarrollo de herramientas que faciliten la gestión y manejo del inmenso volumen de datos y conocimiento que se produce hoy en día. Ya sea en sistemas de diálogo, en la generación de textos de diversa índole o como apoyo a otros sistemas automáticos que requieran una comunicación final en lenguaje natural, se hace patente la importancia y necesidad de desarrollar sistemas de *GLN* que implementen soluciones de calidad y resuelvan adecuadamente las dificultades asociadas a tales procesos.

## Agradecimientos

Esta investigación ha sido financiada por la Generalitat Valenciana a través del proyecto DIIM2.0: Desarrollo de técnicas Inteligentes e Interactivas de Minería y generación de información sobre la web 2.0 (PROMETEOII/2014/001). Además, ha sido parcialmente financiada por la Comisión Europea a través del proyecto SAM (FP7-611312); por el Ministerio de Economía y Competitividad del Gobierno de España mediante los proyectos: "Análisis de Tendencias Mediante Técnicas de Opinión Semántica" (TIN2012-38536-C03-03) y "Técnicas de Deconstrucción en la Tecnologías del Lenguaje Humano" (TIN2012-31224); y finalmente, por la Universidad de Alicante a través del proyecto "Explotación y tratamiento de la información disponible en Internet para la anotación y generación de textos adaptados al usuario" (GRE13-15). Nos gustaría agradecer a Manuel Palomar su apoyo en la investigación realizada, ya que sus ideas han contribuido a que este artículo fuera posible.

## Referencias

1. Androutsopoulos, I., Lampouras, G., & Galanis, D. (2013). Generating natural language descriptions from OWL ontologies: The natural OWL system. *Journal Artificial Intelligence Research*, Vol. 48, No. 1, pp. 671-715.
2. Ballesteros, M., Mille, S., & Wanner, L. (2014). Classifiers for data-driven deep sentence generation. *INLG 2014*, pp. 108.
3. Barzilay, R. & Lapata, M. (2005). Collective content selection for concept-to-text generation. *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, pp. 331–338.
4. Bateman, J. & Zoch, M. (2003). *Natural Language Generation*. Oxford University Press.
5. Bayyarak, H. S. (2011). Efficient algorithm for context sensitive aggregation in natural language generation. *RANLP*, pp. 84–89.
6. Belz, A. & Reiter, E. (2006). Comparing automatic and human evaluation of NLG systems. *11th Conference of the European Chapter of the Association for Computational Linguistics*, The Association for Computer Linguistics, pp. 313–320.
7. Bernardos, M. d. S. (2003). *Marco metodológico para la construcción de sistemas de generación de lenguaje natural*. Ph.D. thesis, Informatica.
8. Bernardos, M. d. S. (2007). ¿Qué es la generación de lenguaje natural? Una visión general sobre el proceso de generación. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, Vol. 11, No. 34, pp. 105–128.
9. Bilmes, J. A. & Kirchhoff, K. (2003). Factored language models and generalized parallel backoff. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Association for Computational Linguistics, pp. 4–6.
10. Binstead, K. & Ritchie, G. (1997). Computational rules for punning riddles. In *Humor*, volume 10. Walter de Gruyter & Co, pp. 25–76.
11. Bleeker, I. M. R. D. (2005). Towards an optimal lexicalization in a natural-sounding portable natural language generator for dialog systems.
12. Bouayad-Agha, N., Casamayor, G., & Wanner, L. (2014). Natural language generation in the context of the semantic web. *Semantic Web*, Vol. 5, No. 6, pp. 493–513.
13. Cahill, L. & Reape, M. (1999). Component tasks in applied NLG systems. Technical Report ITRI-99-05, ITRI, University of Brighton.
14. Calder, J., Evans, R., Mellish, C., & Reape, M. (1999). "Free choice" and templates: how to get

- both at the same time. Technical Report D-99-01, ITRI, University of Brighton.
15. Cambria, E. & White, B. (2014). Jumping NLP curves: A review of natural language processing research. *IEEE Computational Intelligence Magazine*, Vol. 9, No. 2, pp. 48–57.
  16. Cawsey, A. J., Jones, R. B., & Pearson, J. (2000). The evaluation of a personalised health information system for patients with cancer. *User Modeling and User-Adapted Interaction*, Vol. 10, No. 1, pp. 47–72.
  17. Cheng, H. & Mellish, C. (2000). Capturing the interaction between aggregation and text planning in two generation systems. *Proceedings of the first international conference on Natural language generation, Volume 14*, Association for Computational Linguistics, pp. 186–193.
  18. Cheng, H., Mellish, C. S., & O'Donnell, M. (1997). Aggregation based on text structure for descriptive text generation. *Proceedings of the PhD Workshop on Natural Language Generation, 9th European Summer School in Logic, Language and Information*.
  19. Chiarcos, C. (2012). Ontologies of linguistic annotation: Survey and perspectives. *LREC*, pp. 303–310.
  20. Dale, R. & Reiter, E. (1995). Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, Vol. 12, pp. 233–263.
  21. Dale, R. & White, M. (2007). Shared tasks and comparative evaluation in natural language generation. *Proceedings of the Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation*, pp. 1–6.
  22. Dalianis, H. (1996). Aggregation as a subtask of text and sentence planning. pp. 1–5.
  23. Davey, A. (1974). *The Formalisation of Discourse Production*. Ph.D. thesis.
  24. de Rosis, F. & Grasso, F. (2000). Affective natural language generation. pp. 204–218.
  25. Doddington, G. (2002). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. *Proceedings of the second international conference on Human Language Technology Research*, Morgan Kaufmann Publishers Inc., pp. 138–145.
  26. Dzikovska, M. O., Isard, A., Bell, P., Moore, J. D., Steinhauer, N., & Campbell, G. (2011). Beetle II: An adaptable tutorial dialogue system. *Proceedings of the SIGDIAL 2011 Conference*, pp. 338–340.
  27. Edmonds, P. & Hirst, G. (2002). Near-synonymy and lexical choice. *Computational linguistics*, Vol. 28, No. 2, pp. 105–144.
  28. Evans, R., Piwek, P., & Cahill, L. (2002). What is NLG? *Proceedings of Second International Natural Language Generation*, pp. 144–151.
  29. Ferres, L., Parush, A., Roberts, S., & Lindgaard, G. (2006). Helping people with visual impairments gain access to graphical information through natural language: The iGraph system. *Proceedings of the 10th International Conference on Computers Helping People with Special Needs*, Springer, pp. 1122–1130.
  30. Fiedler, A. (2005). Natural language proof explanation. In *Mechanizing Mathematical Reasoning*. Springer, pp. 342–363.
  31. García Ibáñez, C., Hervás, R., & Gervás, P. (2004). Una arquitectura software para el desarrollo de aplicaciones de generación de lenguaje natural. *Procesamiento del Lenguaje Natural*, Vol. 33, pp. 111–118.
  32. Garoufi, K. & Koller, A. (2010). Automated planning for situated natural language generation. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, pp. 1573–1582.
  33. Garoufi, K. & Koller, A. (2011). Combining symbolic and corpus-based approaches for the generation of successful referring expressions. *Proceedings of the 13th European Workshop on Natural Language Generation*, Association for Computational Linguistics, pp. 121–131.
  34. Gatt, A., Portet, F., Reiter, E., Hunter, J., Mahmood, S., Moncur, W., & Sripada, S. (2009). From data to text in the neonatal intensive care unit: Using NLG technology for decision support and information management. *AI Commun.*, Vol. 22, No. 3, pp. 153–186.
  35. Gatt, A. & Reiter, E. (2009). SimpleNLG: A realisation engine for practical applications. *Proceedings of the 12th European Workshop on Natural Language Generation*, Association for Computational Linguistics, pp. 90–93.
  36. Goldberg, E., Driedger, N., & Kittredge, R. I. (1994). Using natural-language processing to produce weather forecasts. *IEEE Expert*, Vol. 9, No. 2, pp. 45–53.

37. Grosz, B. J. & Sidner, C. L. (1986). Attention, intentions, and the structure of discourse. *Computational linguistics*, Vol. 12, No. 3, pp. 175–204.
38. Grosz, B. J., Weinstein, S., & Joshi, A. K. (1995). Centering: A framework for modeling the local coherence of discourse. *Computational linguistics*, Vol. 21, No. 2, pp. 203–225.
39. Halliday, M. A. (1985). *An Introduction to Functional Grammar*. Edward Arnold.
40. Hellmann, S., Lehmann, J., Auer, S., & Brümmer, M. (2013). Integrating NLP using linked data. In *The Semantic Web (ISWC)*. Springer, pp. 98–113.
41. Hervás, R. & Gervás, P. (2005). An evolutionary approach to referring expression generation and aggregation. *Proceedings of the Tenth European Workshop on Natural Language Generation, Aberdeen, Scotland*.
42. Hervás, R. & Gervás, P. (2008). Descripción de entidades y generación de expresiones de referencia en la generación automática de discurso. *Procesamiento de Lenguaje Natural*, Vol. 41, pp. 217–224.
43. Hovy, E. H. (1988). *Generating Natural Language Under Pragmatic Constraints*. L. Erlbaum Associates Inc.
44. Hutchins, W. J. (1986). *Machine translation: past, present, future*. Ellis Horwood Chichester.
45. Johansson, L., Epitropou, V., Karatzas, K., Waner, L., Karppinen, A., Kukkonen, J., Vrochidis, S., & Kompatsiaris, I. The extraction and fusion of meteorological and air quality information for orchestrated services.
46. Johnston, M., Bangalore, S., Vasireddy, G., Stent, A., Ehlen, P., Walker, M., Whittaker, S., & Maloor, P. (2002). MATCH: An architecture for multimodal dialogue systems. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, pp. 376–383.
47. Joshi, A. K. & Schabes, Y. (1997). Tree-adjoining grammars. In *Handbook of formal languages*. Springer, pp. 69–123.
48. Kantrowitz, M. & Bates, J. (1992). Integrated natural language generation systems. *Proceedings of the 6th International Workshop on Natural Language Generation: Aspects of Automated Natural Language Generation*, Springer, pp. 13–28.
49. Koller, A., Byron, D., Cassell, J., Dale, R., Moore, J., Oberlander, J., & Striegnitz, K. (2009). The software architecture for the first challenge on generating instructions in virtual environments. *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Demonstrations Session*, Association for Computational Linguistics, pp. 33–36.
50. Koller, A. & Petrick, R. (2011). Experiences with planning for natural language generation. *Computational Intelligence*, Vol. 27, No. 1, pp. 23–40.
51. Konstas, I. & Lapata, M. (2012). Concept-to-text generation via discriminative reranking. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, volume 1, Association for Computational Linguistics, pp. 369–378.
52. Konstas, I. & Lapata, M. (2013). A global model for concept-to-text generation. *J. Artif. Intell. Res. (JAIR)*, Vol. 48, pp. 305–346.
53. Krahmer, E. & Theune, M. (2002). Efficient context-sensitive generation of referring expressions. *Information sharing: Reference and presupposition in language generation and interpretation*, Vol. 143, pp. 223–263.
54. Krahmer, E., van Erk, S., & Verleg, A. (2003). Graph-based generation of referring expressions. *Comput. Linguist.*, Vol. 29, No. 1, pp. 53–72.
55. Kukich, K. (1983). Design of a knowledge-based report generator. *Proceedings of the 21st annual meeting on Association for Computational Linguistics*, Association for Computational Linguistics, pp. 145–150.
56. Laclaustra, I. M., Ledesma, J. L., Méndez, G., & Gervás, P. (2014). Kill the dragon and rescue the princess: Designing a plan-based multi-agent story generator. *5th International Conference on Computational Creativity*.
57. Langkilde, I. & Knight, K. (1998). Generation that exploits corpus-based statistical knowledge. *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, Association for Computational Linguistics, pp. 704–710.
58. Lavie, A. & Agarwal, A. (2007). Meteor: An automatic metric for MT evaluation with high levels of correlation with human judgments. *Proceedings of the Second Workshop on Statistical Machine Translation*, Association for Computational Linguistics, pp. 228–231.

59. Lavoie, B. & Rambow, O. (1997). A fast and portable realizer for text generation systems. *Proceedings of the Fifth Conference on Applied Natural Language Processing*, Association for Computational Linguistics, pp. 265–268.
60. Lim-Cheng, N. R., Fabia, G. I. G., Quebral, M. E. G., & Yu, M. T. (2014). Shed: An online diet counselling system. *DLSU Research Congress 2014*.
61. Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, Association for Computational Linguistics, pp. 74–81.
62. Liu, M., Calvo, R. A., & Rus, V. (2012). G-Asks: An intelligent automatic question generation system for academic writing support. Vol. 3, No. 2, pp. 101–124.
63. Lloret, E. & Palomar, M. (2013). COMPENDIUM: a text summarisation tool for generating summaries of multiple purposes, domains, and genres. *Natural Language Engineering*, Vol. 19, No. 2, pp. 147–186.
64. Mann, W. (1999). Introducción a la teoría de la estructura retórica (rhetorical structure theory: RST).
65. Mann, W. C. & Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text - Interdisciplinary Journal for the Study of Discourse*, Vol. 8, No. 3, pp. 243–281.
66. McDonald, D. D. (2010). *Natural Language Generation*, chapter 6. CRC Press, pp. 121–144.
67. McKeown, K. R. (1985). Discourse strategies for generating natural-language text. *Artificial Intelligence*, Vol. 27, No. 1, pp. 1–41.
68. Mellish, C. & Dale, R. (1998). Evaluation in the context of natural language generation. *Computer Speech & Language*, Vol. 12, No. 4, pp. 349–373.
69. Mellish, C., Scott, D., Cahill, L., Paiva, D., Evans, R., & Reape, M. (2006). A reference architecture for natural language generation systems. *Natural language engineering*, Vol. 12, No. 01, pp. 1–34.
70. Mille, S., Burga, A., & Wanner, L. (2013). AnCora-UPF: A multi-level annotation of Spanish. *DepLing 2013*, pp. 217.
71. Mille, S. & Wanner, L. (2008). Multilingual summarization in practice: the case of patent claims. *Proceedings of the 12th European Association of Machine Translation conference*, pp. 120–129.
72. Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, Vol. 38, No. 11, pp. 39–41.
73. Žolkovskij, A. & Mel'čuk, I. A. (1965). O vozmožnom metode i instrumentax semantičeskogo sinteza (towards a possible method and tools of semantic generation [in russian]). *Naučno-techničeskaja informacija*, Vol. 5, pp. 23–28.
74. Paiva, D. S. (1998). A survey of applied natural language generation systems.
75. Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318.
76. Paris, C., Scott, D., Green, N., McCoy, K., & McDonald, D. (2007). Desiderata for evaluation of natural language generation. *Proceedings of the Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation*, pp. 9–15.
77. Portet, F., Reiter, E., Hunter, J., & Sripada, S. (2007). Automatic generation of textual summaries from neonatal intensive care data. In *Artificial Intelligence in Medicine*. Springer, pp. 227–236.
78. Reiter, E. (2007). An architecture for data-to-text systems. *Proceedings of the Eleventh European Workshop on Natural Language Generation*, Association for Computational Linguistics, pp. 97–104.
79. Reiter, E. (2010). *Natural Language Generation*. Wiley-Blackbell.
80. Reiter, E. & Belz, A. (2009). An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, Vol. 35, No. 4, pp. 529–558.
81. Reiter, E. & Dale, R. (2000). *Building Natural Language Generation Systems*. Cambridge University Press.
82. Reiter, E., Robertson, R., & Osman, L. M. (2003). Lessons from a failure: Generating tailored smoking cessation letters. *Artificial Intelligence*, Vol. 144, No. 1, pp. 41–58.
83. Reiter, E., Sripada, S., Hunter, J., Yu, J., & Davy, I. (2005). Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, Vol. 167, No. 1, pp. 137–169.
84. Reiter, E., Turner, R., Alm, N., Black, R., Dempster, M., & Waller, A. (2009). Using NLG to help

- language-impaired users tell stories and participate in social dialogues. *Proceedings of the 12th European Workshop on Natural Language Generation*, Association for Computational Linguistics, pp. 1–8.
85. Resnik, P. & Lin, J. (2010). *Evaluation of NLP Systems*. Wiley-Blackwell, pp. 271–295.
  86. Robin, J. & McKeown, K. (1996). Empirically designing and evaluating a new revision-based model for summary generation. *Artificial Intelligence*, Vol. 85, No. 1, pp. 135–179.
  87. Rudnicky, A. I., Thayer, E. H., Constantinides, P. C., Tchou, C., Shern, R., Lenzo, K. A., Xu, W., & Oh, A. (1999). Creating natural dialogs in the Carnegie Mellon communicator system. *Eurospeech*.
  88. Sauper, C. & Barzilay, R. (2009). Automatically generating Wikipedia articles: A structure-aware approach. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Volume 1*, Association for Computational Linguistics, pp. 208–216.
  89. Scott, D. & Moore, J. (2007). An NLG evaluation competition? eight reasons to be cautious. *Proceedings of the Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation*, pp. 22–23.
  90. Singhal, A. (2001). Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, Vol. 24, No. 4, pp. 35–43.
  91. Soricut, R. & Marcu, D. (2005). Towards developing generation algorithms for text-to-text applications. *Proceedings of the 43rd Annual Meeting of the ACL*, Association for Computational Linguistics, pp. 66–74.
  92. Theune, M., Hielkema, F., & Hendriks, P. (2006). Performing aggregation and ellipsis using discourse structures. *Research on Language and Computation*, Vol. 4, No. 4, pp. 353–375.
  93. Viethen, J. & Dale, R. (2007). Evaluation in natural language generation: Lessons from referring expression generation. *Traitement Automatique des Langues*, Vol. 48, No. 1, pp. 141–160.
  94. Viethen, J., Dale, R., Krahmer, E., Theune, M., & Touretzky, D. (2008). Controlling redundancy in referring expressions.
  95. Walker, M. (2007). Share and share alike: Resources for language generation. *Proceedings of the Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation*, pp. 28–30.
  96. Walker, M., Stent, A., Mairesse, F., & Prasad, R. (2007). Individual and domain adaptation in sentence planning for dialogue. *Journal Artificial Intelligence Research*, Vol. 30, No. 1, pp. 413–456.
  97. Walker, M. A., Rambow, O., & Rogati, M. (2001). SPoT: A trainable sentence planner. *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pp. 1–8.
  98. Wan, S., Dras, M., Dale, R., & Paris, C. (2009). Improving grammaticality in statistical sentence generation: Introducing a dependency spanning tree algorithm with an argument satisfaction model. *Proceedings of the 12th Conference of the European Chapter of the ACL*, pp. 852–860.
  99. Wanner, L., Bohnet, B., Bouayad-Agha, N., Laureau, F., & Nicklaß, D. (2010). MARQUIS: Generation of user-tailored multilingual air quality bulletins. *Applied Artificial Intelligence*, Vol. 24, No. 10, pp. 914–952.
  100. Wanner, L., Vrochidis, S., Rospocher, M., Moßgraber, J., Bosch, H., Karppinen, A., Myllynen, M., Tonelli, S., Bouayad-Agha, N., Casamayor, G., Ertl, T., Hilbring, D., Johansson, L., Karatzas, K., Kompatsiaris, I., Koskentalo, T., Mille, S., Mountzidou, A., Pianta, E., Serafini, L., & Tarvainen, V. (2012). Personalized environmental service orchestration for quality of life improvement. In Iliadis, L., Maglogiannis, I., Papadopoulos, H., Karatzas, K., & Sioutas, S., editors, *Artificial Intelligence Applications and Innovations*, volume 382 of *IFIP Advances in Information and Communication Technology*. Springer Berlin Heidelberg, pp. 351–360.
  101. White, M., A. J. Clark, R., & D. Moore, J. (2010). Generating tailored, comparative descriptions with contextually appropriate intonation. *Computational Linguistics*, Vol. 36, No. 2, pp. 159–201.
  102. Williams, S. & Reiter, E. (2008). Generating basic skills reports for low-skilled readers. *Natural Language Engineering*, Vol. 14, No. 04, pp. 495–525.
  103. Yu, J., Reiter, E., Hunter, J., & Mellish, C. (2007). Choosing the content of textual summaries of large time-series data sets. *Natural Language Engineering*, Vol. 13, No. 01, pp. 25–49.