

# Estrategias a nivel de macro y micro-planificación en un prototipo de verbalización de ontologías

Martín Bermudez and Sandra Roger

Grupo de Investigación en Lenguajes e Inteligencia Artificial (GILIA),  
Facultad de Informática. Universidad Nacional del Comahue  
Buenos Aires 1400, (8300) Neuquén  
martin.bermudez@est.fi.uncoma.edu.ar  
roger@fi.uncoma.edu.ar

**Resumen** En este trabajo se presenta un primer prototipo de un trabajo de generación de lenguaje natural capaz de procesar el contenido de una ontología y generar un texto en lenguaje natural. El texto resultante debe estar organizado y expresado con una sintaxis aceptable, para que sea accesible a cualquier usuario cuyo interés sea el dominio modelado, pero no tenga las herramientas necesarias para comprender las lógicas subyacentes.

**Keywords:** Verbalización, Ontología, OWL, Generación del Lenguaje Natural

## 1. Introducción

La Web Semántica puede ser vista como una colección de estándares y tecnologías que permiten a las máquinas entender el significado (semántica) de la información disponible en la Web [10]. Para describir el dominio de interés en la Web Semántica, se adoptó el uso de ontologías. Algunos lenguajes conocidos para representar ontologías son RDF y OWL2.

Como las ontologías estructuran los datos de manera formal, son poco comprensibles por usuarios no expertos, y no brindan una cómoda visualización de la información para aquellos que quieran beneficiarse del uso de las tecnologías semánticas. Por este motivo, expresar el contenido formal en Lenguaje Natural (LN) resulta atractivo, brindando la capacidad de documentar y expresar ontologías en un lenguaje accesible por usuarios no entrenados. Un Sistema de Generación de Lenguaje Natural (SGLN) puede tener diferentes objetivos en cuanto a la salida que producen, ya sea enfocarse en la facilidad de lectura, la capacidad de poder generar diferentes oraciones para expresar la misma información, enfocarse en la organización del contenido, entre otras. En el ámbito de las ontologías existen algunos trabajos que involucran el procesamiento de lenguaje natural [4][5][8]. Como veremos más adelante, cada enfoque posee ventajas y desventajas, por ejemplo en cuanto a la integración con las ontologías, debido

a la necesidad de modificar la ontología subyacente o de requerir convenciones de nombrado; como en su potencial para ser de utilidad al usuario final, debido a la complejidad de las oraciones, la falta de flexibilidad léxica, y la organización del contenido del texto, lo que resulta en una pobre comprensión del texto.

Teniendo en cuenta estas ventajas y desventajas, en este trabajo se propone el primer prototipo de un sistema de generación de texto que mantenga un equilibrio entre la dificultad de integrar el sistema con las ontologías, y la capacidad del sistema de generar texto que resulte útil al usuario final. El Sistema de Generación de Texto debe ser capaz de recibir una ontología de la Web Semántica y generar un documento de texto organizado y en lenguaje natural. No será necesario agregar información a la base de conocimiento, mientras que se espera alcanzar una salida que sea sintácticamente aceptable.

Los dos puntos claves en este trabajo son: la ORGANIZACIÓN DE LA INFORMACIÓN, para estructurar el texto con base en las relaciones semánticas, con el fin de reducir la carga cognitiva que exige reconocer las relaciones del dominio; y la GENERACIÓN DEL TEXTO EN LENGUAJE NATURAL, maximizando la cohesión de las oraciones.

## 2. Organización de la información

En esta sección describimos el diseño de un módulo para organizar y estructurar la información de una ontología de forma más significativa, de manera que favorezca la estructura inicial de un texto, teniendo en cuenta el preprocesamiento para el análisis e interpretación de datos de Reiter E. [7] dando como resultado resultado una jerarquía de tópicos que será usado para poder abordar la coherencia global del texto.

Considerando la suposición de la Teoría de Veins [2], las referencias entre unidades textuales son posibles si se encuentran en una relación estructural. Por lo cual, creamos una estructura jerárquica que facilite la relación entre unidades textuales evitando que una entidad en un nivel  $N$  haga referencia a entidades que se encuentren en niveles  $N+1$  (niveles inferiores).

Las entidades que tienen mayor impacto en la ontología aparecen en los niveles superiores de la jerarquía, con el objetivo de usarlas como hilo conductores para estructurar el texto. Por ello, trataremos dos problemas principales: la obtención de los tópicos más relevantes para tratar en el texto y el recorrido del grafo para obtener las unidades de información adecuada que maximicen la relación semántica respecto a los tópicos del texto (para abordar la coherencia global).

### 2.1. Diseño

Para organizar la información, se crearon cuatro módulos que transforman la ontología de entrada en un árbol con toda la información contenida en la ontología. **Traductor**: recibe la entrada y la traduce a una representación interna de la ontología. **Clasificador de Entidades**: clasifica y extrae las entidades más

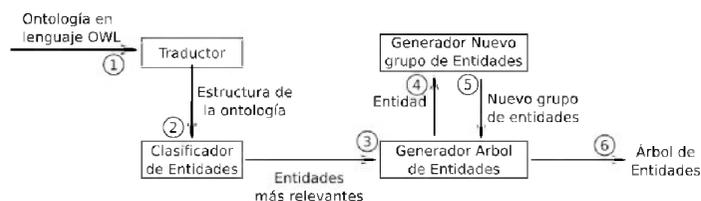


Figura 1: Módulos que componen el Organizador de Información

relevantes, utilizando como criterio las medidas de centralidad. **Generador de Nuevo Grupo de Entidades:** recibe una Entidad y recorre sus relaciones para crear un nuevo grupo de entidades relacionado semánticamente. **Generador del Árbol de Entidades:** se encarga de crear los niveles del Árbol de Entidades.

En la Figura 1 se muestra la transformación de la ontología OWL a través de los módulos. Los pasos 4 y 5 son iterativos, el proceso finaliza cuando no queden más Entidades que explorar, lo cual es definido por el Generador del Árbol de Entidades.

*Criterio de clasificación* Consideramos que una clase tiene más información que otra (y por lo tanto que es más relevante) si está presente en más cantidad de dominios de *ObjectProperties*. Esto permite reconocer las clases que tengan mayor cantidad de conexiones con otras clases, y que a su vez sean el núcleo de la relación. El cálculo de información de cada clase, utilizamos el grafo subyacente a la ontología y calculamos el *outdegree*<sup>1</sup> de cada nodo que represente una clase. Para el cálculo se tiene en cuenta únicamente la relación *rdf:domain*.

Algunas ventajas de este enfoque son: sencillez de implementación<sup>2</sup>; no se requiere agregar información extra al dominio; no es necesario utilizar sobre la ontología un razonador que requiera una complejidad computacional que sea intratable. La única observación es que hay que inferir el dominio y rango de las *ObjectProperties*. Sin embargo, la inferencia de dominio y rango se realiza teniendo en cuenta la jerarquía de *ObjectProperties*, sin necesidad de inferir clases equivalentes o disjuntas.

La eficacia de este enfoque depende fuertemente de que las *ObjectProperties* más importantes tengan el dominio explícito.

*Seleccionando las principales clases* Para obtener el tema principal (o temas principales), se seleccionan las clases que superen el valor promedio entre cantidad de propiedades y cantidad de clases. Si ninguna supera el valor promedio, se selecciona la o las clases con el valor más alto. Cuando se selecciona más de una clase que pertenece a una misma jerarquía, se eliminan las subclases, dejando como representante a la superclase en común.

<sup>1</sup> El *outdegree* es la medida de la cantidad de conexiones salientes de un nodo.

<sup>2</sup> Únicamente se debe recorrer el grafo calculando el valor de cada nodo. El recorrido del grafo tiene a lo sumo una complejidad polinomial.

*Agrupando la información de las Entidades* El tercer módulo crea nuevos grupos de entidades para agregar como un nuevo nivel al Árbol. Los elementos de cada grupo serán insertados como hijos de la Entidad correspondiente.

Dada una Entidad  $E$ , debe recorrer las relaciones de  $E$  en la ontología, para seleccionar Entidades que serán agregadas al nuevo grupo. El contenido de cada grupo depende del tipo de  $E$ . Si  $E$  es una Clase, el nuevo grupo estará constituido por Propiedades, Individuos y Subclases. Si  $E$  es una Propiedad, el nuevo grupo estará constituido por Subpropiedades cuyo dominio contenga a  $E$ , o por las clases del Rango de  $E$  si no posee Subpropiedades. Si por el contrario  $E$  es un Individuo, no se creará ningún grupo nuevo.

Cuando existen más de una Propiedad en un grupo, se busca reemplazarlas por propiedades en común de más alto nivel en la jerarquía de propiedades.

*Generador Árbol de Entidades* El cuarto módulo crea la estructura del Árbol, sincronizando la interacción entre los demás módulos. Su objetivo es formar la jerarquía, comenzando desde las Entidades más Relevantes, y agregando nuevos niveles con ayuda del Generador de Nuevo Grupo de Entidades. Si hay clases que no fueron utilizadas, invoca un algoritmo para agregar estas clases en una rama especial.

### 3. Generación del documento de texto

Para el proceso de generación del documento de texto, se usó el Árbol de Entidades del organizador de información detallado anteriormente para dar una estructura inicial al texto, se agrupó el contenido de cada entidad de la ontología en párrafos y secciones, y luego se verbalizaron los axiomas, teniendo en cuenta la coherencia local dentro de los párrafos generados. Para ello, se crearon tres módulos principales: **Macroplanificador**, **Microplanificador** y **Realizador**, los cuales implementan los procesos asociados a las etapas de Macroplanificación, Microplanificación y Realización gramatical [9], respectivamente.

Para mantener la coherencia local según la Teoría de Centrado [6], y evitar cambiar continuamente los focos de atención, para cada tópico tratado en el texto, agrupamos toda su información en párrafos consecutivos. Dado una entidad  $E$ , para recuperar la información que la describe, se recorre la estructura de grafo a partir del nodo que representa a  $E$ , navegando por las relaciones del nodo, a solo un nivel de profundidad.

La relación entre el Organizador de Información y el Generador del Documento de Texto se puede ver en la arquitectura general de nuestro sistema (Figura 2).

#### 3.1. Diseño Macroplanificador

Para crear el documento en la etapa de Macroplanificación, se reemplazó el módulo Generador del Árbol de Entidades del Organizador de Información, por un nuevo módulo específico del Macroplanificador.

La representación interna del documento de texto tiene en cuenta los siguientes criterios: Cada *Entidad* presente en la estructura que resulta del Organizador de Información es considerada un tópico. A cada *Tópico* del primer nivel le corresponde una sección. Los tópicos anidados (Subclases, Individuos y Relaciones) se corresponden a subsecciones (secciones anidadas). Cada *Sección* está compuesta de al menos un párrafo y un título. Cada *Párrafo* trata un único tópico, el cual puede ser una clase o un individuo. También, un párrafo puede o no tener alguna sentencia. Esto se debe a que no todas las entidades tienen asociados axiomas que la describan.

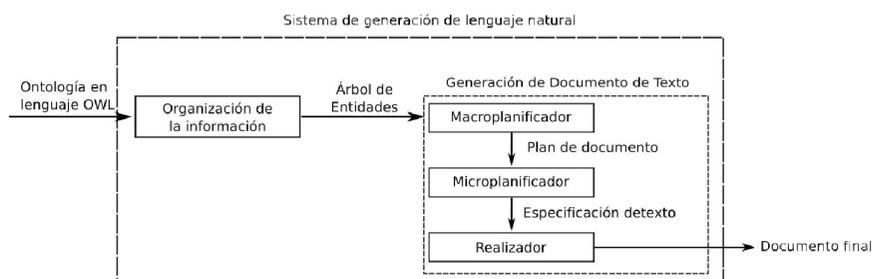


Figura 2: Arquitectura de los módulos que componen nuestro Sistema de Generación de Lenguaje Natural

**Secciones y párrafos** Dentro de una sección para una Clase, se verbalizarán sus axiomas y se enumerarán sus Individuos, Subclases y Relaciones, todo en un párrafo. En una sección asociada a una Relación, solo habrán subsecciones acerca de las Entidades que pertenecen a la Relación. En una sección para Individuos, se creará un párrafo para describir sus propiedades.

**Subsecciones** Las subsecciones son secciones anidadas dentro de otras secciones. Cuando surge a partir de un Individuo, la subsección trata como tópico al Individuo; cuando surge a partir de una Subclase, el tópico principal es la Subclase; y cuando surge a partir de una Relación, el tópico principal resulta ser el rango de la Relación, es decir, otra Clase.

**Oraciones** Las oraciones pertenecen a los párrafos por lo que tratan un solo tópico. Cada oración que tenga como tópico a una Clase, aborda un solo tipo de información, por lo que existe una oración para las clases disjuntas, una oración para las clases equivalentes, y así con los demás tipos de información. Cada oración correspondiente a un Individuo contiene las propiedades con sus valores declaradas sobre el Individuo.

Al tratar un solo tipo de información por oración, se maximiza la cohesión dentro de cada oración.

### 3.2. Diseño Microplanificador

En esta etapa se llevarán a cabo las tareas para describir una Entidad en lenguaje natural a partir de la información que contiene cada grupo creado en la Macroplanificación. Para simplificar, se asoció a cada constructor OWL solo algunas formas de verbalización. Se utilizó *Stanford POS Tagger*<sup>3</sup> para etiquetar las palabras y poder trabajar con patrones de la gramática del lenguaje humano.

El enfoque adoptado para crear la sintaxis de las oraciones se basa en un recorrido bottom-up de la jerarquía de un *Axioma*. Se comienza desde las hojas, construyendo oraciones parciales a partir de las *Entidades*, luego se procede a subir por los niveles a través de los constructores de *Expresiones de Clases*, componiendo nuevas oraciones parciales, hasta alcanzar la raíz de la jerarquía, donde se termina de construir la oración final. Adicionalmente, se realizan algunos tratamientos morfológicos para agregar fluidez y coherencia al texto, tal como inserción de artículos, y manejo de concordancia de género y número.

Con el objetivo de comenzar a formar las oraciones parciales, el primer paso a realizar es la verbalización de las *Entidades*, ya que están directamente relacionadas con las IRIs (o tienen acceso al *label*, en caso de extraer los nombres desde los *labels*). En el nivel de constructores de *Expresiones de Clase*, es posible generar oraciones complejas subordinando las expresiones que se encuentran anidadas, o componer nuevas oraciones parciales teniendo en cuenta los tipos de componentes involucrados. En el nivel más abstracto se encuentran los constructores de *Axiomas*. Estos constructores tienen la característica de no ser recursivos entre ellos, por lo que es posible generar oraciones independientes, yuxtapuestas o coordinadas.

**Componentes y oraciones parciales** Durante la creación de una oración, se crean oraciones parciales y se componen entre ellas. Se asoció a cada constructor un tipo de componente oracional para evitar una exhaustiva programación de composiciones. Los tipos de componentes son los siguientes: *Término (T)*: caracterizado por no poseer verbo. Pueden contener adverbios, sustantivos y adjetivos. *Sintagma Verbal (SV)*: debe poseer un verbo. *Oración Negativa (ON)*: representa la negación de un componente. *Unión*: representa una disyunción de oraciones parciales. *Intersección*: representa una adición o subordinación de oraciones parciales.

Los componentes que retorna cada constructor se definen a continuación: Componente Término (T): (Clase, Individuo). Componente Sintagma Verbal (SV): (Propiedad, Constructores de cuantificación, Constructor *hasValue*, Constructores con cardinalidad). Componente Unión: (*UnionOf*, *OneOf*). Componente Intersección: (*IntersectionOf*). Componente Oración Negativa (ON): (*ComplementOf*).

**Verbalización de constructores OWL** Esta tarea se encarga de componer las oraciones según la información de cada constructor OWL. En algunos constructores se diseñó más de una forma de composición, mejorando la variabilidad, fluidez e interpretación de las oraciones.

<sup>3</sup> Un software para asignar la categoría gramatical a las palabras de un texto <https://nlp.stanford.edu/software/tagger.html>

Para reconocer cómo componer las oraciones en cada constructor, se revisó empíricamente los axiomas de algunas ontologías, y se buscó utilizar oraciones que sean lo más genéricas posibles, que permitan la comprensión de los axiomas.

**Expresiones de referencia** Las expresiones de referencia solo fueron implementadas para las *Clases*. Para llevar a cabo la referenciación, se tuvo en cuenta el uso de pronombres demostrativos (“este/esta”), a veces, acompañados por el primer sustantivo (si posee) del nombre de la *Clase*.

**Agregación de sentencias** Basaremos la agregación en la conjunción por componentes compartidos [1]. El objetivo es que los elementos en común aparezcan una sola vez, mediante elipsis del componente repetido. La agregación de sentencias ocurre en dos lugares. Uno es durante el proceso de producción de una oración en el constructor *N-Ary*, es decir dentro de una oración; el otro es durante el proceso de creación de párrafos, es decir, entre oraciones.

### 3.3. Diseño del realizador

Esta última etapa se encarga de dar formato al Documento de Texto para presentar todas las secciones y párrafos de manera más agradable. Reduciremos las secciones en párrafos, para que el Documento Final sea más compacto, tratando de mejorar la estética y el proceso de lectura, sin perjudicar la coherencia y la segmentación de la información.

Para llevar a cabo la reducción, se midió la información verbalizada, usando como métrica la cantidad de oraciones presentes en la descripción de un tópico.

**Criterio de reducción de Secciones** Para decidir si un tópico es planificado como una Sección, se utilizan los siguientes criterios: debe poseer cinco o más oraciones; debe poseer alguna Subsección con cinco o más oraciones; en cualquier otro caso será planificado como un Párrafo.

Cuanto menor sea la cantidad de oraciones requeridas para ser considerada una sección, la planificación del documento será más similar al Documento Inicial; mientras que si la cantidad de oraciones requeridas es muy grande, la planificación perderá niveles de jerarquía, y únicamente contemplará como secciones a los tópicos del primer nivel. Ninguno de los dos extremos es conveniente: o se pierde coherencia, o se obtiene un texto con demasiados niveles de secciones, que resulta antinatural. Sin embargo, un documento en el que predominan las subsecciones (aunque tengan poca información), ayuda a los humanos a reconocer y establecer relaciones entre los tópicos, siendo el objetivo principal no desaprovechar la semántica del dominio modelado, y teniendo como intención principal que el lector comprenda el dominio modelado en la ontología.

## 4. Una aplicación sobre la ontología *wine*

Analizamos el comportamiento del sistema utilizando como entrada la ontología Wine. Esta ontología tiene como objetivo describir un dominio de vinos y comidas<sup>4</sup>, por lo que esperamos que las entidades que se consideren más rele-

<sup>4</sup> [https://protege.stanford.edu/publications/ontology\\_development/ontology101-noy-mcguinness.html](https://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html)

vantes sean aquellas afines a los vinos y comidas. A continuación se presentan algunos fragmentos del documento generado.

1 Wine A wine is a potable liquid. A wine has wine sugar as sugar, has wine flavor as flavor, has wine body as body, has wine color as color, has wine descriptor and made from grape a wine grape. This located in some region. Also, wine has only winery maker. Also, wine has exactly one maker, made at least one wine grape, wine body, wine color, wine flavor and wine sugar. There are the following kinds of wines: italian wine, . . . , late harvest and alsatian wine.  
 Full bodied wine: a full bodied wine is a wine that has full body.  
 Burgundy: a burgundy is a wine that located in bourgogne region. This has dry sugar. There are the following kinds of burgundies: white burgundy and red burgundy.  
 . . . Otros vinos realizados como párrafos.  
 1.1 Italian wine  
 A italian wine is a wine that located in italian region. chianti is the only kind of wine.  
 Chianti: a chianti is a italian wine. This has only light or medium body. Also, chianti has red color, moderate flavor and dry sugar. located in chianti region. made from grape sangiovese grape. chianti classico is a type of chianti.  
 Chianti classico: a chianti classico has mc guinnesso maker and medium body.  
 1.2 ... 1.7  
 1.8 Red wine  
 A red wine is a wine that has red color. There are the following kinds of wines: dry red wine, red burgundy, port and red bordeaux.  
 Port: a port is a red wine. This has full body, strong flavor and sweet sugar. located in portugal region. taylor port is a type of port. Taylor port: a taylor port has taylor maker.  
 1.8.1 Red burgundy  
 A red burgundy is a burgundy and red wine. This made from grape pinot noir grape. Also, red burgundy made at most one wine grape. cotes d or is the only kind of burgundy. Cotes d or: a cotes d is a red burgundy that located in cotes d or region. This has moderate flavor. clos de vougeot cotes d or is a type of cotes d or. Clos de vougeot cotes d or: a has clos de vougeot maker.  
 . . . Otros vinos realizados como secciones  
 1.14 Wines descriptor  
 Wine sugar: a wine sugar is a wine taste. a wine sugar is a sweet, off dry or dry. there are the wines sugar: dry, off dry and sweet.  
 Wine color: a wine color is a wine descriptor. a wine color is a rose, red or white. there are the wines color: white, red and rose.  
 Wine flavor: a wine flavor is a wine taste. a wine flavor is a moderate, delicate or strong. there are the wines flavor: moderate, strong and delicate.  
 Wine body: a wine body is a wine taste. a wine body is a light, medium or full. there are the wines body: medium, full and light.

Para identificar el formato del texto, a las secciones se le incluye el número de sección a la izquierda del título de la sección, y cada párrafo comienza con sangría.

La ontología Wine cuenta con 138 clases y 16 propiedades. De las 16 propiedades, 12 fueron usadas para verificar qué clases superan el promedio para ser seleccionadas como las principales. El promedio fue de 2.5 y la única clase que lo superó fue Wine, lo que cumple parcialmente el resultado esperado, ya que parte del objetivo de la ontología es describir el dominio de los vinos. Respecto a la sección que describe las comidas, quedó desplazada a *otras secciones*, siendo un resultado no esperado según el objetivo de la ontología. Sin embargo, analizando manualmente la ontología, se puede apreciar que el porcentaje de información que describe a las comidas es significativamente menor en relación a la información referida a los vinos, factor por el cual resulta aceptable que no aparezcan como sección principal.

Analizando la coherencia global y el sentido del texto, la estructuración resultó razonable, enfocada en los vinos y las propiedades de los vinos.

La implementación de la sintaxis presenta algo de redundancia y repetitividad, aunque las oraciones son claras. El resultado de la realización final mantiene la coherencia global y establece las relaciones pertinentes entre los tópicos.

A continuación se presenta una comparación de la verbalización de la clase *Beaujolais* de la ontología *Wine*, entre el trabajo [3] y el nuestro. En [3] proponen listar sus características de la siguiente manera: “*A Beaujolais is a Wine that:*

- *is made from at most 1 grape, which is Gamay Grape,*
- *has Delicate flavor,*
- *has Dry sugar,*
- *has Red color,*
- *has Light body ”*

En nuestro caso, la oración resultante es la siguiente: “**Beaujolais has light body, red color, delicate flavor and dry sugar. Made from grape gamay grape.**”

Existe clara diferencia, no solo en el criterio de presentación lista frente a prosa, sino también en el hecho de elegir cuánta elipsis emplear (en nuestro caso, además de omitir *Beaujolais*, también omitimos el verbo “has” para evitar redundancia), y de separar en diferentes oraciones las características asociadas al “has” y las asociadas al “made from”. Decidir qué representación usar depende de muchos factores, por ejemplo, la versión en prosa no resulta muy compleja, pues la omisión del “has” solo ocurre en tres características, por lo que no recae tanta carga sobre el lector. Quizá en descripciones más largas sea conveniente un formato en lista, pero también hay que tener en cuenta cómo impactaría visualmente representar todas las clases de una gran ontología de esta forma. De acuerdo con [3], pueden existir métodos de generación más complejos que mejoren la calidad del texto de salida.

## 5. Conclusiones

Se diseñó un Sistema de Generación de Lenguaje Natural, orientado a generar un texto organizado a partir de una ontología (OWL) de la Web Semántica.

Para maximizar la relación semántica entre los tópicos, se propuso una solución basada en el análisis del grafo subyacente a la ontología, midiendo las relaciones entre sus nodos. De esta manera se obtuvieron las Entidades de la ontología que resultaban más sobresalientes, con el fin de usarlas como hilo conductor de los temas del texto generado. Esta solución se vio motivada por la característica que poseen las ontologías de la Web Semántica, de ser grafos libres de escala [11].

El resultado obtenido del caso de prueba retornó una estructura organizada, que se correspondía con los resultados esperados. Sin embargo, la solución propuesta depende de la cantidad de *ObjectProperties* y del contenido de sus dominios, por lo que queda pendiente poder analizar la solución con mayor diversidad de ontologías, variando la densidad de sus relaciones.

Además de maximizar la relación entre los tópicos, se buscó maximizar la relación entre los elementos de los párrafos. Para esto se usó como base la coherencia local, a través de los postulados de la Teoría de Centrado.

Como se mostró en la sección 4, la solución basada en el desarrollo de estos módulos resultó en textos legibles y organizados. Si bien validar textos puede resultar subjetivo, podemos apreciar que existe suficiente relación semántica en el orden de los temas que se presentan, tanto entre las Secciones como entre las Oraciones, por lo que cumplen con cierta Coherencia Global y Local.

Aún resulta conveniente explorar otras técnicas para desarrollar las gramáticas y la composición de las oraciones. Desarrollar las gramáticas manualmente es un trabajo costoso, y debido a la naturaleza recursiva de las construcciones de los axiomas y de la semántica de sus constructores, es difícil mantener el control de la oración compuesta, tanto a nivel sintáctico como semántico. Queda pendiente probar el sistema con otras ontologías que presenten mayor complejidad en sus axiomas, con el fin de desarrollar nuevas gramáticas que abarquen más casos de prueba.

## Referencias

1. S. Bernardos. Marco metodológico para la construcción de sistemas de generación de lenguaje natural. *Unpublished doctoral dissertation*. *Informatica*, 2003.
2. D. Cristea, N. Ide, and L. Romary. Veins theory: A model of global discourse cohesion and coherence. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, pages 281–285. Association for Computational Linguistics, 1998.
3. D. Hewlett, A. Kalyanpur, V. Kolovski, and C. Halaschek-Wiener. Effective nl paraphrasing of ontologies on the semantic web. In *Workshop on end-user semantic web interaction, 4th int. semantic web conference, galway, ireland*, 2005.
4. D. C. Moreno and M. V. Lombardo. Ontología y procesamiento de lenguaje natural. *KnE Engineering*, pages 492–501, 2018.
5. M. C. Pérez Hernández. Explotación de los corpórea textuales informatizados para la creación de bases de datos terminológicas basadas en el conocimiento. *Estudios de Lingüística del español*, 18:000–0, 2002.
6. M. Poesio, R. Stevenson, B. D. Eugenio, and J. Hitzeman. Centering: A parametric theory and its instantiations. *Computational linguistics*, 30(3):309–363, 2004.
7. E. Reiter. An architecture for data-to-text systems. In *Proceedings of the Eleventh European Workshop on Natural Language Generation*, pages 97–104. Association for Computational Linguistics, 2007.
8. M. Vállez. La web semántica y las tecnologías del lenguaje humano. In *Web semántica y sistemas de información documental*, pages 155–180. Trea, 2009.
9. M. Vicente, C. Barros, F. S. Peregrino, F. Agulló, and E. Lloret. La generación de lenguaje natural: análisis del estado actual. *Computación y Sistemas*, 19(4):721–756, 2015.
10. L. Yu. *A developer's guide to the semantic Web*. Springer Science & Business Media, 2011.
11. H. Zhang. The scale-free nature of semantic web ontology. In *Proceedings of the 17th international conference on World Wide Web*, pages 1047–1048. ACM, 2008.