INTRODUCCIÓN

1.1 GENERALIDADES

Como es sabido, el lenguaje natural es el medio de comunicación de las personas, y por tanto, las consultas y muchos textos que se hacen y se encuentran en la web están expresados en algún tipo de lenguaje natural, tal como el español o el ingles. Por otro lado, normalmente las consultas en la web son tratadas sintácticamente, desaprovechando importante información tácita del perfil del usuario o contenida en la web; así, existe gran cantidad de información en crudo conteniendo opiniones, ideas, hechos, y teorías de diversos dominios, en lenguaje natural, esperando ser explotada. De aquí la importancia y la necesidad de tratar los documentos web y la consulta del usuario a un nivel más allá del meramente sintáctico, a saber, a un nivel semántico y pragmático.

Recientemente, en el CEMISID se ha venido trabajando en una propuesta de un marco ontológico dinámico semántico para la web semántica, que consiste en la especificación ontológica del procesamiento de consultas, expresadas en lenguaje natural, para la web [1]. Dicho marco requiere la definición de una ontología lingüística del lenguaje español, extendida para el uso coloquial del lenguaje, y una meta-ontología interpretativa sobre el perfil del usuario y/o la usabilidad de la web semántica. El marco ontológico es dinámico en el sentido que evoluciona a través de mecanismos de aprendizaje automático, para adaptarse a los cambios de la web y de los usuarios. Algunos de los objetivos que se persiguen con el marco ontológico semántico son: usar el lenguaje natural español para realizar las consultas, hacer procesos de razonamiento automáticos basados en marcos ontológicos con el fin de explotar el contenido semántico sobre la web, aprender sobre el uso y contenido de la web, entre otras cosas.

Para lograr el proceso de adaptación del MODS se requiere de un componente de aprendizaje que soporte al proceso de adquisición de conocimiento. En otras palabras, responder a preguntas tales como: *¿Cómo aprender de manera automática información lingüística y ontológica a partir de una unidad léxica y/o recursos recuperados de la web?, ¿Cómo este nuevo conocimiento entrara a formar parte del marco ontológico de manera racional?,* es decir, de qué forma asegurar que la actualización se realice de manera correcta y se incluya en la estructura adecuada (ontología interpretativa, ontología lingüística, lexicón, onomasticon[1]), lógica y consistentemente.

En la tesis se trata de responder a estas interrogantes a través del desarrollo de un componente de aprendizaje automático definido inicialmente para el MODS, pero que puede ser incorporado y extendido a otros sistemas dinámicos basados en conocimiento. Sin este componente de aprendizaje, el MODS se reduce a un simple sistema restringido y limitado de interpretación de consultas, basado en estructuras de conocimiento dependientes del ingeniero de conocimiento. Además, se elabora un prototipo para dos casos de aprendizaje en concreto, a saber: aprendizaje de información léxica, y aprendizaje de relaciones de dominio o relaciones no taxonómicas.

Esta investigación contribuye, entre otras cosas, al desarrollo de futuros sistemas de información basados en conocimiento, al tratamiento del lenguaje natural, al estudio de formas de aprendizaje automático de ontologías, y al desarrollo de agentes inteligentes especializados en dominios de conocimiento que requieran un aprendizaje automático.

1.2 ANTECEDENTES

Una reciente teoría que afronta el desafío de interpretar los textos y o sentencias en lenguaje natural se denomina Semántica Ontológica (Ontological Semantics:

[1] Un onomasticon es una lista de nombres propios

Ontología Semántica) [2]. La Semántica Ontológica revela la posibilidad de una interpretación automática de una sentencia o texto, aproximadamente a como lo haría un hablante nativo. En Semántica Ontológica, S. Nirenburg y V. Raskin introducen un enfoque completo para la representación del significado de un texto por computador. Ellos argumentan que el uso del significado o sentido de las palabras es clave para el procesamiento del lenguaje natural. La representación del significado del texto propuesto por la Semántica Ontológica se basa en fuentes de conocimiento estático (incluye una ontología, un repositorio de hechos y el lexicón), más procesos implicados en el análisis de texto y la adquisición de conocimiento.

Por otra parte, los actuales sistemas basados en conocimiento usan ontologías computacionales para modelar el conocimiento y lograr un mayor nivel de interoperabilidad entre sus actores (software, personas). Las ontologías han mostrado ser una buena respuesta para estructurar el conocimiento, proporcionar un vocabulario formal y sofisticado de un dominio particular, compartido por un grupo de personas y "comprensible" por las maquinas [3]. Actualmente, hay resultados en este campo que se han convertido en estándares/recomendaciones, como RDFS[2] y OWL[3]. OWL está provisto de semántica formal y razonamiento a través de lógica. La lógica de predicados y la lógica descriptiva han sido usadas para este propósito.

Gracias a las bondades de expresividad, deducción y objetividad que las ontologías tienen, se han convertido en la tecnología usada para compartir y reusar conocimiento y ser la base de la web semántica, así como de un gran número de sistemas de gestión de conocimiento. La construcción manual de ontologías es, sin embargo, un significativo cuello de botella. Actualmente, el aprendizaje de ontologías (*ontology learning:* OL) se ha visto como un paradigma

[2] RDFS (Resourse Description Framework Schema: RDFS) introduce algunos conceptos ontológicos sencillos. Es una extensión semántica de RDF, proporciona mecanismos para describir grupos de recursos y las relaciones presentes entre estos.

[3] OWL (Web Ontology Languaje:OWL) es un lenguaje que permite la definición sofistica de ontologías, un requerimiento fundamental en la integración de contenidos de información heterogéneos.

promisorio para construir ontologías de manera semi-automática. OL se ha enfocado en dos problemas fundamentalmente [4]:

1. El desarrollo de métodos, metodologías, herramientas y algoritmos para *integrar* ontologías existentes. La integración se realiza de las siguientes maneras:

a. Por fusión/*merging*) de ontologías, para crear una única ontología coherente [5].
b. Por alineación/*aligning* de ontologías, para establecer asociaciones entre ellas y reusar información de ambas [6].
c. Por mapeo/*mapping* de ontologías, para encontrar elementos correspondientes a cada una [7].

Algunos ejemplos de integración se muestran en SENSUS[4] y Cyc[5] [8], para crear una ontología de alto nivel sobre conocimiento del mundo, y en PROMPT[6] [9], para la fusión y alineación de ontologías.

2. El desarrollo de métodos, metodologías, herramientas y algoritmos para el aprendizaje y adquisición de ontologías semi-automáticamente.

A continuación se presentan algunos sistemas de aprendizaje de ontologías relacionados con la tesis, (una lista donde se incluyen otros sistemas, se puede ver en la Tabla 1 del Capítulo 2).

WEBKB: es un sistema para el aprendizaje de ontologías, cuyo objetivo es el aprendizaje de instancias y reglas desde documentos de la Web, combinando

[4] SENSUS es una ontología resultado de la fusión manual del modelo PENMAN, WordNet, y otras ontologías.
[5] Una gran base de conocimiento con conocimiento de sentido común creada por Microelectronics Y Computer Technology Corporation (MCC) en la década de los 80.
[6] PROMPT, la principal asunción de este método es que las ontologías a ser fusionadas/merged son formalizadas con un modelo del conocimiento de sentido común basado en marcos.

métodos estadísticos (Aprendizaje Bayesiano) y lógicos (aprendizaje de reglas de lógica de predicados) [10].

KAWB: es un sistema para el aprendizaje de ontologías, cuyo objetivo es la "adquisición de conocimiento de dominio a partir del lenguaje natural". Tiene dos fases, la de representación del texto y la de análisis. En la fase de representación se utiliza herramientas de anotación para caracterizar secuencias de caracteres, y en la fase de análisis estas anotaciones son explotadas para descubrir correlaciones en el texto [11].

SYNDIKATE: es un sistema para el aprendizaje de gramáticas y ontologías. Una ontología es incrementalmente actualizada con nuevos conceptos tomados de textos del mundo real. El proceso de adquisición se centra alrededor de la "calidad" conceptual y lingüística de conceptos hipotéticos. Sobre la base de la calidad de la evidencia (esta calidad es más una heurística, un tanto subjetiva, del ingeniero del conocimiento, quien dice que tan importante es un concepto), los conceptos son ordenados de acuerdo a la credibilidad, y los más favorecidos son seleccionados para su ingreso a la base de conocimiento [12].

Seguidamente se describen algunos sistemas para el aprendizaje de información morfosintáctica desde el lenguaje natural, particularmente, de extracción de términos[7] (una lista donde se incluyen otros sistemas, se puede ver en la Tabla 2 del Capítulo 2):

ANA (Adquisición Automática Natural): es un sistema para el aprendizaje de términos, el cual se basa en una simplificación de textos y la observación de patrones recurrentes. ANA está compuesta de dos módulos: un primer modulo de confianza, familiaridad o conocimiento, y otro modulo de descubrimiento. El primer módulo usa un conocimiento base (palabras preestablecidas, palabras específicas del dominio) para la detección de nuevos términos. El segundo modulo consiste en

[7] Términos "una unidad léxica que designa un concepto en un dominio dado"

un proceso de adquisición gradual o incremental de nuevos términos a partir del conocimiento existente. [13].

CLARIT: es un sistema para el aprendizaje de términos, cuyo principal objetivo es la indexación de documentos. El sistema realiza un procesamiento textual para determinar términos complejos, con el fin de lograr una descripción más apropiada del documento. Es por ello que se incluyen entre los sistemas detectores de términos [14].

ACABIT: su objetivo es la extracción de términos. Utiliza un analizador híbrido compuesto de un analizador sintáctico y un filtro estadístico, a partir de textos anteriormente etiquetados. El analizador es una máquina finita de estados que utiliza las categorías gramaticales del texto etiquetado, y otras características morfológicas, tal como morfología de las palabras (para formar clases de palabas), morfemas (para formar frases compuestas) etc. [15].

Por último, con relación al aprendizaje de relaciones no-taxonómicas, que son de interés para nuestro trabajo, tenemos los siguientes trabajos (una lista donde se incluyen otros sistemas, se puede ver en la Tabla 3 del Capítulo 2):

ASIUM: es un sistema que soporta el aprendizaje de estructuras verbales y conocimiento taxonómico. Se basa en un análisis estadístico y un analizador/parsing sintáctico de textos. Usa aprendizaje de máquina, agrupamiento o clustering [16].

DODDLE: es un sistema que soporta el aprendizaje de relaciones taxonómicas y no taxonómicas usando métodos estadísticos (análisis de co-ocurrencia)[8]. Usa WordNet[9] y textos específicos de dominio [17].

[8] Análisis de coocurrencia es un método automático que permite, a través del análisis de documentos por medio de un programa informático, establecer el número y grado de apariciones simultáneas de palabras o grupos de palabras en conjuntos de documentos, así como la distancia a la que ocurren. La co-ocurrencia

CAMELEON: es un sistema que encuentra relaciones léxicas taxonómicas y no taxonómicas en textos plano, por medio de patrones léxico-sintácticos para enriquecer un modelo conceptual [18].

1.3 OBJETIVOS

1.3.1 Objetivo General

Se plantea como objetivo principal del proyecto diseñar un componente de aprendizaje de ontologías, capaz de adquirir nuevo conocimiento en función de la información suministrada por el proceso de análisis de una consulta en lenguaje natural, y/o de la información recuperada desde la web por dicha consulta, para potenciar con este nuevo conocimiento las estructuras semánticas del "Marco Ontológico Dinámico Semántico para la WEB Semántica".

1.3.2 Objetivos Específicos

- Análisis de las metodologías y/o métodos más relevantes para el diseño y construcción de ontologías, y estudio de las teorías sobre semántica ontológica, ingeniería ontológica y aprendizaje de ontologías.

- Definición formal del componente de aprendizaje de ontologías, capaz de adquirir nuevo conocimiento en función de la información suministrada por el proceso de análisis de una consulta en lenguaje natural, y/o de la información recuperada desde la web por dicha consulta.

- Elaboración de un prototipo del componente de aprendizaje con dos métodos de aprendizaje diferentes; uno para el aprendizaje de nuevos términos, que

---

permite establecer términos de indización en proporción directa a la frecuencia de aparición de los términos.

9 WordNet es una enorme base de datos léxica del idioma inglés. Agrupa las palabras en conjuntos de sinónimos llamados 'synsets', proporcionando definiciones cortas y generales, y almacenando las relaciones semánticas entre estos conjuntos de sinónimos.

impactara el lexicón, y otro para el aprendizaje de relaciones no-taxonómicas, que impactara las ontologías de dominio interpretativa y lingüística. Ambos esquemas de aprendizaje tendrán fuentes de información distintas, la primera derivada del proceso de análisis de la consulta, y la segunda derivada de la información recuperada desde la web por la consulta.

1.4 Organización del trabajo

El documento se estructura en cinco partes. Este capítulo 1 es la introducción del trabajo, donde se presenta generalidades sobre el contexto de la investigación que nos ocupa, los antecedentes de los trabajos afines con la investigación, y cuáles son los objetivos pretendidos en la tesis. En el capítulo 2 se presenta el marco teórico de base de nuestra propuesta, que consiste en la teoría sobre aprendizaje ontológico y las variantes usadas en este trabajo, y la arquitectura propuesta del marco ontológico dinámico semántico. En el capítulo 3 se describe la arquitectura propuesta del componente de aprendizaje; sus módulos y funcionamiento con el MODS. En el capítulo 4 se presenta la implementación del prototipo de prueba, con dos métodos de aprendizaje: uno para aprender unidades léxicas de orden superior (verbo, sustantivo, adjetivo y adverbio), que impactará el conocimiento léxico (lexicón-onomasticon y ontología lingüística); y otro para aprender relaciones no-taxonómicas a partir de la información recuperada de la web, que impactará las ontologías interpretativa y lingüística. En el capítulo 5 se presenta un caso de estudio, y por último, las conclusiones.

2. MARCO TEORICO

2.1 APRENDIZAJE ONTOLÓGICO

2.1.1 Aprendizaje de ontologías en general

En dominios poco complejos ("de juguete" dicen algunos autores), el problema de la representación de conocimiento no es importante, y es fácil encontrar un vocabulario[10] consistente [19]. Por otro lado, dominios complejos, como lo es la consulta de información en lenguaje natural a la web (entorno cambiante), requieren de representaciones generales y flexibles. Para la representación general de conocimiento se tienen las ontologías computacionales, y para su flexibilidad o adaptabilidad el aprendizaje de ontologías. Hoy, las ontologías se utilizan en el desarrollo de un gran número de aplicaciones en diversas áreas, como la gestión de conocimiento, procesamiento de lenguaje natural, recuperación de la información, y más recientemente en la Web Semántica. En general, una ontología *es una especificación formal de una conceptualización compartida.* Las ontologías tienen los siguientes componentes:

Clases: que representan conceptos tomados en su sentido muy amplio. Por ejemplo, el concepto *categoría_lexica*[11]. En ontología, los conceptos están normalmente organizados en taxonomía, por medio de las cuales se pueden aplicar mecanismos de herencia.

Relaciones: que representan un tipo de asociación entre los conceptos del dominio. Por ejemplo, la relación *calificar_sustantivo* establece la relación entre un adjetivo que califica/determina al sustantivo.

[10] Palabras y reglas de formación de un lenguaje.
[11] Son categorías léxicas, los verbos, sustantivos, adjetivos, adverbios, entre otras.

Axiomas: sirven para modelizar afirmaciones que son siempre ciertas. Un ejemplo es el axioma *una proposición/sentencia es la concatenación de una frase nominal y una frase verbal*.

Instancias: que se utilizan para representar elementos o individuos en una ontología. Ejemplos de instancias del concepto *verbo_regular* son *"estudiar", "amar".*

Construir y ensamblar manualmente todos estos elementos en un todo ontológico ha sido un significativo cuello de botella para el Ingeniero del conocimiento[12]. La dificultad radica a nivel del conocimiento que se requiere para su diseño, el tiempo y esfuerzo necesario, y potenciales errores en el diseño (por ejemplo, inconsistencias, etc.), lo que ha dado origen al aprendizaje de ontologías.

El proceso de aprendizaje de ontologías se propone como una tecnología que resuelve parte de este problema (aún hay participación del ingeniero en su funcionamiento), y tiene que ver con la extracción de elementos ontológicos[13] a partir de diferentes fuentes, para construir o actualizar una ontología existente de manera significativa[14].

A continuación se describen varios sistemas de aprendizaje de ontologías. Estos sistemas se caracterizan por usar diferentes técnicas de aprendizaje y diversas fuentes de información.

[12] Un ingeniero de conocimiento u ontológico, es alguien que investiga un dominio concreto, aprende que conceptos son los más importantes en ese dominio, y crea una representación formal de los objetos y relaciones del dominio.
[13] Son considerados elementos ontológicos, concepto/palabra, relación, función, axioma, instancia.
[14] Significativa quiere decir que se estable una conexión con el conocimiento existente y no como islotes de nueva información.

Tabla 1. Sistemas de aprendizaje de ontologías

Nombre del Sistema	Descripción del Sistema
ONTOLEARN	Es un sistema que permite enriquecer una ontología de dominio con conceptos y relaciones. Usa aprendizaje de máquina (el principal objetivo del Aprendizaje de Máquina (ML por sus siglas en Inglés) es el desarrollo de sistemas que puedan cambiar su comportamiento de manera autónoma basados en su experiencia. El ML ofrece algunas técnicas para el descubrimiento de conocimiento (patrones) tal como cadenas de markop[15]) y procesamiento de lenguaje natural para este objetivo [20].
DLLEARNER	Es un sistema que aprende una definición de concepto en lógicas descriptivas. Este sistema se basa en programación lógica inductiva (PLI) y lógica descriptivas[16]. El sistema recibe como entrada una base de conocimiento y un conjunto de entrenamiento con ejemplos positivos y negativos de instancias del concepto de interés [21].
SVETLAN'	Es un sistema para construir una ontología a partir de jerarquías de nombres. Toma dominios semánticos como unidades temáticas, y construye dominios estructurados para clasificar los nombres de acuerdo con los verbos y las relaciones entre ellos [22].
KEA	Es un sistema para el descubrimiento automático de sentencias (keyphrases) en documentos de texto. La técnica de aprendizaje usada por el sistema es aprendizaje de máquina, estadística, y procesamiento léxico [23]
HASTI	Es un sistema para la construcción automática de ontologías, usa como entrada información no-estructurada en la forma de texto en lenguaje natural persa. HASTI no usa conocimiento previo, es decir, construye las ontologías desde cero. Usa un lexicón[17] que se encuentra inicialmente vació, y va creciendo incrementalmente en la medida que va aprendiendo nuevas palabras. HASTI aprende conceptos, relaciones conceptuales taxonómicas, no-taxonómicas, y axiomas, con lo cual va construyendo la ontología sobre un nucleo/kernel inicial. HASTI usa un enfoque simbólico hibrido (una combinación de lógica, lingüística, y métodos heurísticos, entre otros) [24].

[15] Una cadena de Márkov es una serie de eventos, en la cual la probabilidad de que ocurra un evento depende del evento inmediato anterior. En efecto, las cadenas de este tipo tienen memoria. "Recuerdan" el último evento y esto condiciona las posibilidades de los eventos futuros.

[16] Las lógicas descriptivas (Descripción Logic:DL) son un formalismo de representación de conocimiento, base para los lenguajes ontológicos [38]

2.1.2 Aprendizaje de información morfosintáctica desde el lenguaje natural

El aprendizaje de información morfosintáctica consiste en una serie de métodos de aprendizaje para la adquisición, ampliación y refinamiento de la información morfológica y sintáctica contemplada en el lexicón[18], y sin los cuales el marco ontológico dinámico semántico no podría robustecerse en su capacidad de interpretación del lenguaje natural.

Siguiendo a Moreno y col. [25], cualquier sistema automático de procesamiento de lenguaje natural utiliza las estructuras del lenguaje natural para generar los procesos de análisis. Dentro de las estructuras del lenguaje natural que tiene que ver con al aprendizaje morfosintáctico, se tienen a las unidades léxicas de alto nivel (verbos, sustantivos, adjetivos, adverbios).

Existen varios métodos para el aprendizaje de información morfosintáctica (en adelante "términos") a partir, principalmente, de información no-estructurada. Varios de estos enfoques se basan en métodos de recuperación de información, y procesamiento de lenguaje natural y terminológico [26].

Tabla 2. Sistemas de aprendizaje de términos

Nombre del Sistema	Descripción del Sistema
DAILLE	Su objetivo es la extracción de términos. La principal idea de este sistema es combinar conocimiento lingüístico con cálculos estadísticos. Primero, un corpus contiene toda la información morfológica. Luego se genera una lista de términos candidatos de acuerdo con unos patrones de formación de términos. Esta información usa métodos estadísticos para filtrar esta lista [27].

[18] Un lexicón es una lista de palabras en un lenguaje-un vocabulario-junto con algún conocimiento del uso de cada palabra.

FASTR	Su principal objetivo consiste en la detección de variantes de términos. El objetivo de este sistema es detectar variantes de los términos desde un conjunto de términos previamente conocidos. Estos términos de inicio pueden estar disponibles en una base de datos o un programa de adquisición de términos [28].
HEID	Su principal objetivo está en la extracción de términos. Para este sistema se parte del hecho que un sistema de extracción de términos automático tiene varias aplicaciones, siendo las más importantes la construcción de diccionarios o glosarios. Para la construcción de un diccionario desde corpus lingüísticos se distinguen dos fases: una de pre-análisis lingüístico y otra de identificación de términos [29].
NEURAL	Su objetivo es extraer términos. Es un sistema de naturaleza hibrida. Usa conocimiento lingüístico y estadístico. Del conocimiento lingüístico se tienen patrones morfológicos y una lista de sufijos específicos del dominio, y del conocimiento estadístico la frecuencia, etc. [30].
LEXTER	Utiliza una máquina de estados finita para obtener los sintagmas nominales[19] máximos. Una vez extraídos estos sintagmas, los divide en sub-sintagmas, para obtener los que aparecen en el corpus en situaciones no ambiguas [31].
TERMINO	Está formado por tres fases secuenciales: filtrado de texto, análisis de sintagmas nominales basados en gramática, y construcción del conjunto de términos. El analizador morfológico utilizado se basa en reglas gramaticales en vez de diccionarios, y se centra en el análisis de los sintagmas nominales. A diferencia de ACABIT y LEXTER, TERMINO posee su propio analizador morfológico (POS-Tagger[20]), y permite ambigüedades gramaticales en el análisis morfológico [32].
TERMS	Se basa en la idea que los términos suelen repetirse en documentos técnicos con más frecuencia que sintagmas nominales no terminológicos, y que los sintagmas nominales que representan términos tienen una estructura diferente que los sintagmas nominales que no los representan. Normalmente, los sintagmas que representan términos no suelen incluir modificadores ni determinantes, al contrario que los sintagmas nominales comunes. Así, TERMS utiliza un filtro estadístico que rechaza las cadenas que aparecen sólo una vez en el documento [33].

[19] Un sintagma nominal (SN) designa alguno de los participantes en la predicación verbal. Está constituido por un nombre (sustantivo) o adjetivo. También se le llama frase nominal (FN).

[20] Part-of-speech tagging (POS tagging o POST), también llamado etiquetado gramatical, es el proceso de asignar (o etiquetar) las palabras. Usado para obtener la categoría gramatical de cada palabra en la frase actual.

2.1.3 Aprendizaje de relaciones no-taxonómicas

De la misma manera que el aprendizaje de información morfosintáctica es clave para el lexicón, el aprendizaje de información semántica es fundamental para la ontología interpretativa (dominio) y lingüística. El aprendizaje de información semántica cubre el aprendizaje de conceptos, relaciones taxonómicas, relaciones de dominio o no-taxonómicas, propiedades y axiomas, sin los cuales el marco ontológico dinámico semántico no podría robustecerse en su capacidad de interpretación del lenguaje natural.

El aprendizaje de relaciones no-taxonómicas consiste en el descubrimiento de verbos relacionados con el dominio, la extracción de conceptos que están relacionados no taxonómicamente, relaciones de marcado (por ejemplo, las etiquetas puestas a un producto, que hablan sobre el producto), entre otras cosas.

En general, las relaciones no-taxonómicas se refieren a cualquier relación entre conceptos, excepto la relación *"es_un",* tales como sinónimos, meronimos[21], antónimos, etc. Un grupo de relaciones de interés para este trabajo son:

Relación de equivalencia: establece la igualdad o equivalencia entre dos conceptos aparentemente diferentes. Esta relación suele aparecer en el lenguaje natural con expresiones parecidas a las siguientes:

• *"A equivale a B", "A es igual a B".*

Relación de dependencia: es un tipo especial de relación de asociación a través de responsabilidades, parentesco, propiedades, etc. Esta relación suele aparecer en el lenguaje natural en las formas siguientes:

• *"A está asociado a B", "A es responsable de B", "A depende de B".*

[21] Se denomina *merónimo* a la palabra cuyo significado constituye una parte del significado total de otra palabra.

Relación topológica: describe la distribución espacial de conceptos físicos, e interconexiones entre esos conceptos. Esta relación suele aparecer en el lenguaje natural en las formas siguientes:

• *"A está a la derecha de B", "A está encima de B", "A está debajo de B", "A está dentro de B", "A contiene a B", "A está conectado a B", "A está al lado de B",*

Relación causal: Este tipo de relación describe como, dados unos estados o acciones, se induce a otros estados o acciones. Esta relación suele aparecer en el lenguaje natural de la siguiente manera:

• *"A es la causa de B", "A necesita a B", "A activa a B"*

Relación funcional: describe las condiciones para las acciones y reacciones que tienen lugar, y las posibles consecuencias de las acciones. Esta relación suele aparecer en el lenguaje natural en las formas siguientes:

• *"A permite a B", "A necesita a B", "A activa a B"*

Relación cronológica: Esta relación se conoce también como relación temporal, y describe la secuencia de tiempo en la que ocurren eventos. Esta relación suele aparecer en el lenguaje natural en las formas siguientes:

• *"A ocurre antes de B", "A ocurre después de B", "A y B ocurren simultáneamente", " A ocurre durante B", "A comienza cuando B termina".*

Relación de similaridad: establece que conceptos son iguales o análogos, y en qué medida. Esta relación suele aparecer en el lenguaje natural en la forma siguiente: *"A es similar a B"*

Relación condicional: define las condiciones en las cuales ciertas cosas tienen lugar. Esta relación suele aparecer en el lenguaje natural en la forma siguiente:

• *"A tiene como condición B", "A está condicionada por B", "Si A entonces B".*

Relación de propósito: Esta relación establece el por qué y el para qué de los conceptos. Esta relación suele aparecer en el lenguaje natural en la forma siguiente:

•*"A ocurre con la finalidad de B", "A ocurre para B", "A nace con el propósito de B".*

A continuación se presentan algunos sistemas que descubren algunos de los tipos de las relaciones de domino antes expuestas:

Tabla 3. Sistemas que soportan el aprendizaje de relaciones no-taxonómicas

Nombre del Sistema	Descripción del Sistema
TEXT-TO-ONTO	Su principal objetivo consiste en el aprendizaje de relaciones taxonómicas y no-taxonómicas. Se basa en análisis estadístico y un analizador/*parsing* sintáctico de textos en francés, más reglas de asociación y de poda/acotación[22]. Aprende desde texto libre, semi-estructurado, desde otras ontologías y bases de datos. El resultado del proceso de aprendizaje es una ontología de dominio, que contiene conceptos específicos del dominio bajo estudio. Todo el proceso es supervisado por el ingeniero ontológico [34].
MARSID	Es una propuesta para el descubrimiento de reglas de asociación entre un conjunto de Ítems en una gran base de datos. Dada una base de datos de las transacciones de los clientes de un almacén, y cada transacción consiste la canasta de compra de un cliente en una visita, se plantea un algoritmo en este trabajo que genera todas las reglas de asociación significativa entre los artículos reportados en la base de datos. El algoritmo incorpora técnicas de estimación y de poda [35].

[22] La técnica de *poda* se suele interpretar como un árbol de soluciones, donde cada rama nos lleva a una posible solución posterior a la actual. La característica de esta técnica (y a la que debe su nombre) es que el algoritmo se encarga de detectar en qué ramificación las soluciones dadas ya no están siendo óptimas, para «podar» esa rama del árbol y no continuar malgastando recursos y procesos en casos que se alejan de la solución óptima.

2.2 MARCO ONTOLÓGICO DINAMICO SEMÁNTICO PARA LA WEB SEMÁNTICA: MODS

El MODS es una propuesta novedosa para el análisis de consultas en lenguaje natural, para la Web. La consulta para el MODS, más que una petición de información, es un elemento cargado de información útil, para formarse una idea del tipo de usuario, y aproximarse de manera sucesiva a una respuesta que cada vez mas cubra las necesidades del mismo.

MODS tiene el desafío de permitir interpretar y formalizar la consulta realizada por el usuario en lenguaje natural, y refinar sus esquemas internos frente a la dinámica de la web (para lo cual requiere de mecanismo de adaptación), para tratar futuras consultas.

Antes de adentrarnos en la arquitectura, miremos el siguiente caso. Si un usuario realiza la siguiente pregunta en lenguaje natural a la Web: *"Se quiere conocer todas las universidades venezolanas que ofrecen la materia Sistema Operativos Avanzados en modalidad a Distancia".* Una persona puede interpretar (según el marco ontológico mediante el cual representa y entiende el mundo que lo rodea) lo siguiente: el dominio es Educación, está interesado en Universidades de Venezuela, tanto a nivel de pregrado y postgrado, y lo que la persona se presupone quiere saber es las Universidades que dictan la materia Sistema Operativos Avanzada en modalidad a distancia. Pero las máquinas carecen de ese marco ontológico que poseen los humanos, por lo tanto, se tienen que usar técnicas y herramientas de diferentes áreas (procesamiento del lenguaje natural (PLN), ontologías, etc.) que permitan representar la consulta en una formalización entendible por la Web, y a partir de esa formalización obtener el resultado deseado por el usuario.

Actualmente existen trabajos y teorías reportadas en la literatura especializada con una finalidad semejante: transmitir la habilidad de "comprensión" del lenguaje

natural a las maquinas, a un nivel aproximado a como lo haría un hablante nativo. A continuación nombramos algunos de ellos.

El sistema MESIA (Modelo Computacional para Extracción Selectiva de Información de textos cortos) es un modelo computacional para extracción selectiva de información de textos cortos [36]. El objetivo es mejorar los resultados de la búsqueda de documentos en la Web de los buscadores tradicionales basados fundamentalmente en métodos estadísticos. MESIA recibe la consulta del usuario escrita en lenguaje natural, y posteriormente la convierte en una consulta booleana. Durante este proceso se produce una expansión de la consulta mediante recursos lingüísticos. Después de la búsqueda, MESIA incorpora información sobre el dominio al proceso, permitiendo la expansión semántica de resultados. Por otro lado, una vez identificado el tema de la consulta, a los resultados obtenidos se le añaden enlaces sobre asuntos relacionados con dicho tema].

El sistema QACID[23] (Sistema de Búsqueda Respuesta, basado en Ontologías) es un sistema para buscar una respuesta ajustada/contextual a los requerimiento del usuario [37]. A QACID los constituyen una ontología de dominio, los datos estructurados, la base de conocimiento de usuario, y el modulo de implicación textual. La *ontología* usada por QACID es implementada en OWL[24], los *datos* (información sobre el dominio) son almacenados en RDF (Resourse Description FrameWork) y consultados con SPARQL[25], la Base de Conocimiento se creó a partir de hechos reales vía preguntas representativas sobre el dominio, y el modulo de *implicación textual* implementa técnicas de implicación textual con el objetivo de inferir deducciones semánticas a partir de las preguntas de entrada y la base de conocimiento previamente obtenidas.

[23] Acrónimo en ingles, Question Answering on CInema Domain
[24] OWL (del ingles, Ontology Web Language) es un lenguaje de marcado para publicar y compartir datos usando ontologías, www.w3.org/TR/owl-features/.
[25] SPARQL es un lenguaje est´andar de consulta para la recuperaci´on de informaci´on desde datos RDF, (www.w3.org/TR/rdf-sparql-query).

Específicamente, en [1] se propone una arquitectura para la interpretación de una consulta en lenguaje natural para la Web Semántica, llamada MODS. A continuación presentamos la arquitectura de MODS

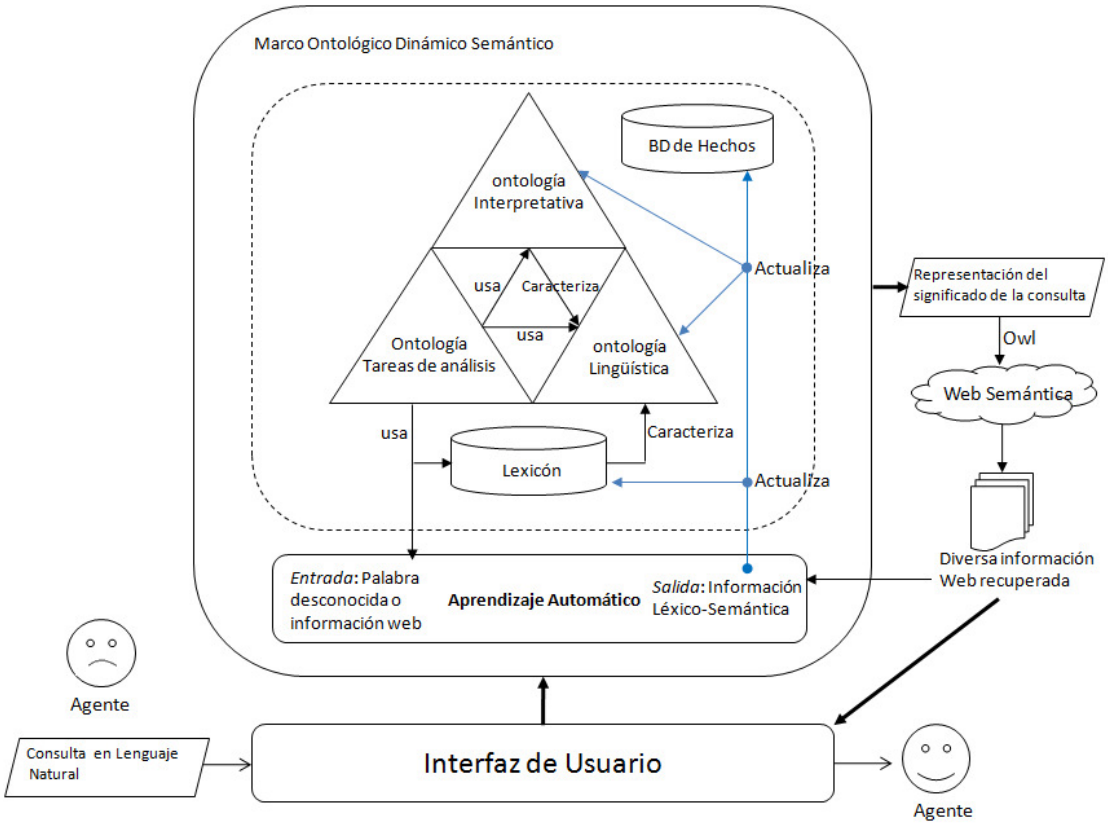

Figura 1. Marco Ontológico Dinámico Semántico

A groso modo, el MODS se activa por una consulta en lenguaje natural, la cual es recibida a través de una interfaz de usuario. A la consulta se le sustrae el significado con el MODS, luego, este significado es especificado en OWL para ser enviado a la Web. La información recuperada de la petición semántica en OWL a la Web, junto con la información proveniente del análisis de la consulta, son usados como entrada para el componente de aprendizaje.

2.2.1 Pasos generales para la interpretación de la consulta en el MODS

1. El usuario expresa su consulta en lenguaje natural al sistema
2. La ontología de tareas analiza las oraciones proporcionadas, a nivel léxico,morfológico y sintáctico; es decir, si las frases contienen palabras compuestas por morfemas, si las estructuras de las oraciones son correctas, etc., utilizando para ello los lexicones y la ontología lingüística.

3. El siguiente paso de la ontología de tareas es analizar las oraciones semánticamente, es decir saber cuál es el significado de cada oración, y asignar el significado de estas a expresiones lógicas, utilizando como insumo al lexicón y la ontología lingüística.

4. Una vez realizado el paso anterior, se realiza el análisis pragmático de la consulta, para lo cual, la estructura de representación obtenida en el paso anterior se reinterpreta para determinar su significado real y puntual dentro del contexto específico. Para ello se utiliza la meta-ontología interpretativa. Una vez realizado ese paso, ya se tiene la expresión final.

5. La expresión final es la representación del significado de la consulta, ella es transformada en OWL, para realizar la consulta propiamente dicha hacia la web semántica.

De manera general, MODS transforma la consulta a un lenguaje ontológico, utilizando los diferentes componentes que lo componen: el lexicón, la ontología lingüística, la ontología de tareas y la ontología de dominio. De esta manera, el MODS utiliza mecanismos de la semántica ontológica y herramientas del procesamiento de lenguaje natural, para el procesamiento de las consultas de los usuarios.

Como ya se ha comentado, la arquitectura la componen tres sub-marcos ontológicos, a saber (Figura 1): *la ontología de tareas*, esta ontología modela las tareas de procesamiento de la consulta en lenguaje natural (análisis léxico-

morfológico, análisis sintáctico, análisis semántico, análisis pragmático); *la ontología lingüística*, la cual especifica la gramática del lenguaje español, junto con una extensión de derivaciones coloquiales (ejemplos de sentencias coloquiales que se incluyen en esa extensión son frases como "booteo del equipo", préstamo lingüístico desde el idioma inglés muy usado en la jerga informática, u otros más nativos como "mamar gallo").La ontología lingüística cuenta con un *lexicón*, el cual caracteriza al lenguaje español, que a su vez contiene un onomasticon[26] para el manejo de nombres propios, términos especializados, y/o coloquiales. El último marco ontológico es la *meta-ontología interpretativa*, que modela el conocimiento sobre el contexto específico del usuario. Ella es una ontología de alto nivel, con especializaciones/extensiones basadas en ontologías de dominio, externas al MODS. También, en la ontología interpretativa se encuentra la ontología del usuario, la cual describe el uso del sistema que va haciendo cada usuario, lo que permite incorporar a la consulta formal las características propias del usuario (contextualización), para intentar delimitar la respuesta de la web.

Finalmente, otro componente clave para la adaptabilidad del MODS a la dinámica de la web y del usuario, es el componente de aprendizaje de ontologías que se propone en este trabajo, cuyo fin es permitir que las ontologías evolucionen a la par con la usabilidad del sistema.

2.2.2 Lexicón del MODS

El lexicón para un lenguaje (en este caso, el español) es una colección de entradas que son indexadas desde el lexema de la palabra, y describe todos sus posibles usos [4]. El lexicón del MODS contiene información sobre su raíz, categoría, ortografía y morfología. Así, cada entrada del lexicón está compuesta por un conjunto de tipos de información léxica. Estos tipos son, como se dijo antes, los siguientes:

[26] Un onomasticon es una colección de nombres propios y/o términos.

Tabla 4. Tipos especificados para el lexicón

RAIZ	Lexema de la entrada léxica
CAT	Categoría léxica, las cuales son: verbo (V), nombre (N), adjetivo (ADJ), adverbio (ADV), conjunciones (CON), artículos (ART), pronombre (PRN) (la categoría se relaciona con la categoría de la ontología lingüística)
ORTH	Abreviación de la entrada léxica
MORPH	Información morfológica, formas irregulares, clase o paradigma y variantes

A modo de ejemplo, para la unidad léxica "*Golpear*" se tendría el siguiente registro en el lexicón.

Tabla 5. Llenado de tipos para verbo golpear

RAIZ	golpe
CAT	verbo transitivo (v tr)
ORTH	Golpear
MORPH	Formas no personales • Infinitivo: golpear • Participio: golpeando • Gerundio: golpeando Indicativo • Presente: golpeo golpeas golpea golpeamos golpeáis golpean Futuro Simple Null

Estos registros léxicos son extendidos de manera automática por medio del componente de aprendizaje léxico. Por ejemplo, para el caso del registro anterior,

el componente de aprendizaje lo complementaría con información del tiempo futuro, a saber:

Tabla 6. Extensión del tipo MORF en el tiempo Futuro para el verbo golpear

RAIZ	golpe
CAT	verbo transitivo (v tr)
ORTO	Golpear
MORF	Formas no personales • Infinitivo: golpear • Participio: golpeando • Gerundio: golpeando Indicativo • Presente: golpeo golpeas golpea golpeamos golpeáis golpean Futuro Simple golpearé golpearás golpeará golpearemos golpearéis golpearán

2.2.3 Ontología lingüística del MODS

Su función es apoyar el procesamiento del lenguaje natural. En ella se describe de manera general a las unidades léxicas como objetos lingüísticos en una base de datos léxica, y las relaciones que se dan entre estas unidades léxicas en una jerarquía conceptual (taxonomía ontológica). Por lo tanto, se puede decir que la ontología lingüística determina una representación de conceptos lingüísticos, y sus relaciones en un dominio específico, en este caso, el lenguaje español.

Tabla 7. Tipos definidos para la ontología lingüística

CAT	Categoría léxica: verbo, nombre, adjetivo, adverbio, conjunción, articulo, pronombre (las mismas categorías que las definidas en el lexicón)
ESTRUCTURA SINTÁCTICA	Conjunto de reglas de producciones de la gramática en español. Representa la estructura sintáctica del lenguaje.
ESTRUCTURA SEMÁNTICA	Cada regla de producción de la gramática representada en la estructura sintáctica, tiene una representación semántica (ellas constituyen las reglas de producción semántica)

A modo de ejemplo, para la unidad léxica: verbo transitivo.

Tabla 8. Tipos definidos para la ontología lingüística

CAT	Verbo transitivo (V_TR)
ESTRUCTURA SINTÁCTICA	1.3 FV ::= VR_TR AR OBJ[27] Existen otras reglas de producción.
ESTRUCTURA SEMÁNTICA	Agente[28] $var(S)[29] Tema[30] $var(OBJ)

[27] Frase verbal (FV) → Verbo transitivo (VR_TR) Articulo (Ar) Objeto (Obj)
[28] Agente: es la entidad que causa o es responsable de una acción (por ejemplo, humano, objeto, animal)
[29] $var(•) convención usada para variable.
[30] Tema: es la entidad manipulada por la acción (por lo general objetos (rara vez humanos)

El sistema hace uso de la ontología lingüística para obtener información a un nivel abstracto sobre una unidad léxica, y saber su rol en la consulta. Por ejemplo, para el caso de los sustantivos/nombre saber que es la entidad que causa o es responsable de una acción.

2.2.4 Ontología interpretativa del MODS

La ontología interpretativa tiene la siguiente estructura, la cual servirá para representar el conocimiento de algún dominio.

Tabla 9. Componente de una ontología computacional.

Conceptos	Son las ideas básicas que se intenta formalizar. Por ejemplo, los objetos, de un dominio dado: animales, material bibliográfico, etc.
Relaciones	Representa la interacción y enlace entre los conceptos del dominio. Suelen formar la taxonomía del dominio. Por ejemplo: subclase-de, parte-de, conectado-a. etc.
Instancias	Se utilizan para representar objetos determinados de un concepto.
Axiomas	Son teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología. Por ejemplo: "*Si A y B son de la clase C, entonces A no es subclase de B*", etc.

Como se ve en la tabla 9, la ontología interpretativa utilizada en el MODS tiene la estructura clásica de las ontologías. Además, desde ella se podrán hacer enlaces a otras ontologías de dominios específicos (por ejemplo, del área de universidades, biología molecular, por mencionar algunas), según los intereses de los usuarios.

Además, la ontología interpretativa modela el perfil del usuario. En el MODS se identifica el perfil del usuario y su situación, la localización relativa del usuarios, el momento en el que se encuentra, la tarea que está realizando o

quiere realizar, y por último, y más importante pero complejo, la finalidad u objetivo del usuario.

Tabla 10. Tipos definidos para la ontología interpretativa

Usuario	Representa el perfil de usuario en cuanto a: 1. **Quien:** Se debe conocer *Quienes* están presentes en el entorno, porque ellos serán en cualquier momento usuario del MODS. 2. **Donde:** La Ubicación de los usuarios que están en el entorno es indispensable en el MODS, nos permitirá determinar los servicios que se le ofrecerán al usuario. 3. **Cuando:** No solo la ubicación física es importante, sino además la ubicación en el tiempo de todos los elementos. Si bien se está preparado para que todas las actividades se realizan en cualquier momento, las actividades que se pueden realizar dependerán de quienes estén en ese momento. 4. **Por qué:** Si bien existen actividades cuya realización siguen un patrón estructurado, las cuales no representan mayor problema para el MODS; el MODS debe estar preparado al libre albedrío humano, y al hecho de que a veces realizamos cosas cuya razón solo nosotros conocemos. 5. **Qué.** Debe conocer *Que* es lo que realizan Quienes están en el entorno. O mejor dicho, cuales son las actividades que se pueden realizar dentro del MODS
Relación	Representa la interacción y enlace entre quien, donde, por qué, y qué. Forman la taxonomía de la ontología usuario.
Instancias	Se utilizan para representar objetos determinados de un usuario.
Axiomas	Son teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología.

2.2.5 Ontología de tareas del MODS

La ontología de tarea permite el procesamiento del lenguaje natural. La estructura de la ontología de tarea es la siguiente:

Tabla 11. Tipos definidos para la ontología de tareas

Tarea	Representa la tarea a realizar en cada fase del análisis, • Análisis–léxico-morfológico. • Análisis-sintáctico. • Análisis-semántico. • Análisis-pragmático.
Relación	Representa la interacción y enlace entre las tareas. Forman la taxonomía de la ontología de tareas, como se muestra en la Figura 5.
Instancias	Se utilizan para representar las tareas específicas de análisis de una consulta dada.
Axiomas	Son teoremas que se declaran sobre relaciones, que deben cumplir los elementos de la ontología de tarea.

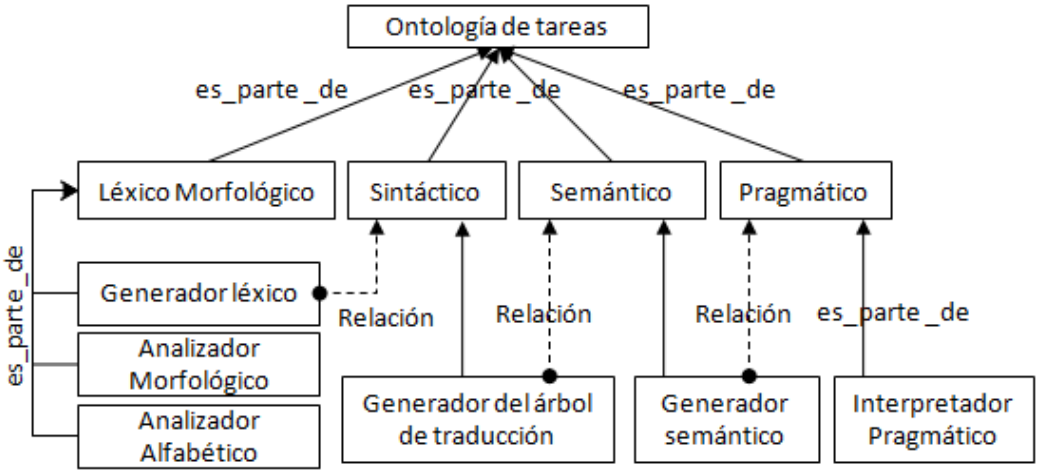

Figura 2. Taxonomía de la ontología de tareas

A continuación se describe la ontología de tareas y su relación con los otros componentes utilizados en el MODS: La primera tarea a realizar es el análisis léxico-morfológico, y se usa como componente de apoyo al lexicón. En esta tarea se obtiene la información morfológica de cada una de las palabras que se encuentran en la consulta, y el onomasticon para el manejo de nombres propios.

El análisis léxico-morfológico se divide en las siguientes sub-tareas: Análisis léxico, su principal función es de recorrer la consulta de entrada y separarla en componentes léxicos (tokens en inglés). La estructura de datos que se va utilizar para el análisis léxico es la siguiente:

lex(CL,TipoCL)

donde, lex: indica que se está en la tarea de análisis léxico, CL: Componente Léxico o Token, y TipoCL: Tipo de Componente Léxico: palabra, número, etc.

Por ejemplo dada la siguiente consulta “relación entre libro y revista”, se obtiene del análisis léxico:

lex(relación,palabra)
lex(entre,palabra)
lex(libro,palabra)
lex(y,palabra)
lex(revista,palabra).

El siguiente paso, con el apoyo del lexicón y la ontología lingüística, es el análisis morfológico, el cual consiste en determinar la forma, clase o categoría gramatical de cada palabra de la consulta. Para realizar este análisis se procede de la siguiente manera: teniendo como entrada el resultado del análisis léxico, se pasa a determinar a qué categoría pertenece cada palabra: nombre (Nom), adjetivo (Adj), pronombre (Pron), verbo (Ver), adverbio (Adver), preposición (Pre), articulo (Art), si se trata de una palabra variable, etc. La estructura de datos que se va utilizar es:

lex_mor(componente léxico, categoría, genero, número, modo, tiempo, aspecto, voz, persona, instancia_ontología_linguistica) donde:

Componente léxico: es una de las palabras de la consulta.

Categoría: es el tipo de palabra, unidades léxicas-superiores, unidades léxicas inferiores [31].

Género: indica si éste pertenece al masculino o al femenino.

Numero: indica si el objeto nombrado es uno o más de uno, en español hay dos números: singular y plural.

Modo: se refiere a la actitud del hablante con respecto a lo que dice. Enuncia el hecho de manera real y objetiva. Hay tres modos verbales: Indicativo (por ejemplo, he llegado a la ciudad). Subjuntivo (indica subordinación, por ejemplo, *presente el informe pronto*). Imperativo (para formular ordenes, por ejemplo, *cierre la puerta*).

Tiempo: es la capacidad que tiene el verbo para situar la acción en un contexto temporal determinado. Normalmente, un verbo expresa nociones que se sitúan en el presente, en el pasado, o en el futuro.

Aspecto: expresa si la acción del verbo ha acabado o tiene sentido durativo.

Voz: la voz nos indica si el sujeto realiza la acción (sujeto activo o agente), o sufre la acción que realiza otro (sujeto pasivo o paciente). En el primer caso, decimos que el verbo está en voz activa; cuando el sujeto es paciente, el verbo está en voz pasiva

Persona: primera, segunda o tercera; yo-tu-el-nosotros-vosotros-ellos

[31] Una unidad léxica superior/alto nivel se refiere a los verbos, sustantivos, adjetivos, adverbios, y las de orden inferior las restantes, tal como, pronombre, artículos, etc…

Siguiendo la consulta del ejemplo anterior, se encontraron los componentes léxicos en el lexicón y en la ontología lingüística para la palabra *relación*, que se muestran a continuación.

Tabla 12. Valores de argumentos de componente *lex_mor*(.) para *relación*.

relación	
Categoría	Nombre
Genero	Femenino
Numero	Singular
Instancia_Ontología_lingüística	Categoría::=N $var(N)::=relación

Por lo tanto, el resultado obtenido del análisis léxico-morfológico es el siguiente:

lex_mor(relación,nombre,femenino,singular,null,null,null,null,null,$var(N)::= relación)

Para el resto sería:

lex_mor(entre, preposición, null, null, null, null, null, null,null,$var(Prep) ::= entre)

lex_mor(libro,nombre,masculino,singular,null,null,null,null,null,$var(N) ::= libro)

lex_mor(y,conjunción,null,null,null,null,null,null,null,$var(Conj) ::= y)

lex_mor(revista,nombre,femenino,singular,null,null,null,null,null,$var(N) ::= revista)

Después del análisis léxico-morfológico se obtiene la siguiente regla de producción de la consulta del usuario, según la posición de cada componente léxico en la consulta.

Consulta ::= N Prep N Conj N

En este caso se encontraron todos los componentes léxicos en el lexicón, por eso se pudo generar dicha regla de producción, pero puede darse el caso en que no

se encuentren. Suponiendo en el ejemplo anterior que el componente léxico desconocido es “entre”, la regla de producción de la consulta del usuario sería la siguiente:

Consulta ::= N desconocido N Conj N

En ese caso se enviaría el componente léxico desconocido al componente de aprendizaje, con el fin de aprender, a partir de él, su información léxica, y continuar el proceso de análisis. Todo lo dicho hasta ahora forma parte de la primera tarea de la ontología de tareas.

El siguiente paso, después de realizar el proceso léxico-morfológico, es el análisis sintáctico, el cual, a partir de la salida del proceso anterior, y con el fin de detectar oraciones o frases significativas para el lenguaje (en nuestro caso español), usa una gramática. Dicha gramática del lenguaje natural representa el “núcleo” que define la naturaleza de los componentes (verbos, sustantivos, artículos, etc.), sus variantes (conjugación, tiempos, genero, número, etc.), y reglas para su interrelación (frases, enunciados, interrogaciones, negaciones, etc.).

Independientemente de la gramática, el proceso de “traducción” compara las reglas que se encuentra en la ontología lingüística contra las palabras del texto de entrada, cada regla que empata agrega un elemento a la estructura, o la termina de generar. La estructura que se produce es el “árbol de traducción”, en donde aparecen las reglas y el resultado del empalme.

Por ejemplo, para el texto de entrada “*Juan es bueno*”, después de haber pasado por el análisis léxico-morfológico, y realizado el proceso de traducción utilizando la ontología lingüística, la cual tiene definidas las siguientes reglas de producción:

1. O ::= FN FV

1.1 FN ::= S

1.2 S ::= N_P

1.3 FV ::= V Adver

Genera el árbol de producción siguiente (ver figura 3).

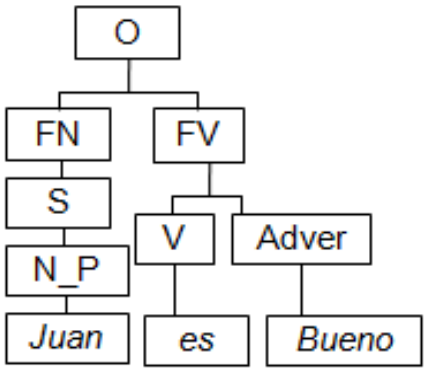

Figura 3. Árbol de traducción generado después del análisis sintáctico a la sentencia "*Juan es bueno*".

Después del análisis sintáctico, con la ontología lingüística se realiza el análisis semántico, el cual es el encargado de establecer qué combinaciones de significados de palabras individuales son posibles a la hora de crear un significado coherente en una oración, lo que puede reducir el número de posibles significados para cada palabra en una oración determinada. En este análisis se verifica nuevamente la oración, para identificar las palabras claves con las cuales el MODS podrá interpretar la consulta.

El análisis semántico, como parte del MODS, representa el núcleo de su "conocimiento", y en función a su variedad y detalle será la riqueza de vocabulario, expresión, entendimiento, respuesta y utilidad que el propio análisis ofrezca. Por ejemplo, para la consulta "*relación entre libro y revista*" se obtiene el siguiente árbol semántico (Figura 4).

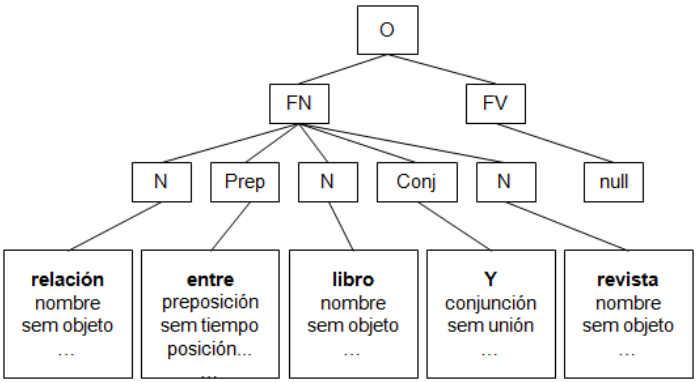

Figura 4. Árbol semántico generado después del análisis semántico

La última tarea de la ontología de tareas es el análisis pragmático o del contexto. Dado que la semántica se ocupa del significado literal de las expresiones lingüísticas, la pragmática trata el significado adicional que adquieren las consultas del contexto del usuario.

Por lo tanto, el analizador pragmático utiliza la estructura semántica obtenida en la tarea anterior para desarrollar la interpretación final de la consulta, en función de las circunstancias del contexto, usando para ello la meta-ontología interpretativa. En esta fase se cubren aspectos tales como la identificación de objetos referenciados por determinados constituyentes de la frase (sintagmas nominales[32], pronombres, elementos elididos, etc.), el análisis de aspectos temporales, la identificación de la intención del usuario (temas y focos), así como el proceso inferencial requerido para interpretar apropiadamente la consulta dentro del dominio de aplicación.

Por ejemplo, la consulta “relación entre libro y revista” tiene varias interpretaciones, las cuales se menciona algunas a continuación

1. ¿Cuál es la relación de libro y revista?
2. ¿Un libro y una revista son material bibliohemerografico?
3. ¿una revista y un libro son consultados?

4 ¿una revista, referente a la supervisión de algo?

Después del análisis pragmático, la interpretación final de la consulta, en función del contexto y del perfil del usuario, da el siguiente resultado:

1. ¿Cuál es la relación de libro y revista?

Terminando el proceso de la ontología de tareas.

3. ARQUITECTURA PARA EL APRENDIZAJE AUTOMÁTICO DE ONTOLOGÍAS

Antes de establecer la arquitectura del componente de aprendizaje automático de ontologías, primero se describen los pasos o sub-tareas que constituyen el desarrollo de una ontología, ya sea de manera manual o semi-automática, con el fin de ofrecer una mejor panorámica de cómo se usará el sistema de aprendizaje de ontologías propuesto. Una vez descrito los pasos del proceso de desarrollo de una ontología, se pasa a describir la arquitectura propuesta del componente de aprendizaje automático de ontologías enfocados en el MODS.

3.1 PROCESO DE DESARROLLO DE ONTOLOGÍAS.

El desarrollo de una ontología tiene que ver principalmente con la definición de conceptos y las relaciones entre ellos, pero además, está el conocimiento acerca de los términos[33]/palabras que son usados para referirse a ellos. Esto implica la adquisición de conocimiento lingüístico sobre los términos; su información léxica, posibles relaciones con otros términos, usos, etc. Una ontología consiste también de una estructura taxonómica (relaciones del tipo "*es_un*") y relaciones no-taxonómicas o de dominio. Finalmente, con el objeto de derivar (deducir) nuevos hechos a partir de los ya existentes, se definen algunas reglas y/o axiomas. Todos estos aspectos, en el desarrollo de una ontología lingüística, se pueden ver de manera grafica en la figura 5.

[33] Un *término* hace referencia a una palabra simple o compuesta; tal como Libro, Revista, Pararrayos, Universidad de los Andes, etc.

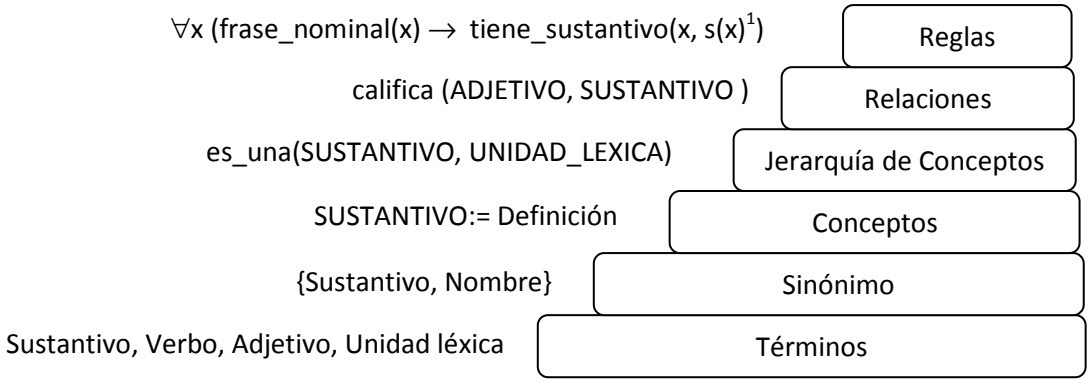

Figura 5. Pasos para el desarrollo de una ontología lingüística por niveles, adaptada de [39]

La figura 5 muestra los pasos para el desarrollo de una ontología lingüística, al frente de cada nivel se muestra un ejemplo. En orden a identificar los conceptos del dominio, es necesario, en primer lugar, identificar los términos en lenguaje natural que se refieren a ellos. La identificación de sinónimos ayuda a evitar la redundancia de conceptos, a clarificarlos, puesto que dos o más términos en lenguaje natural pueden referirse al mismo concepto. Los términos son la fuente que sirven para identificar los conceptos, los cuales formarán parte de la ontología. La definición de un concepto normalmente se plantea en términos de su definición intencional[34], definición extensional[35] y su referente lingüístico, que viene siendo el término. El siguiente paso es identificar las relaciones taxonómicas (generalización y especialización) entre los conceptos. Luego las relaciones no-taxonómicas, cuyo objetivo es definir relaciones entre conceptos conocidos (por ejemplo, entre adjetivo y adverbio, o entre verbo y sustantivo, etc.) por medio del análisis de información sobre el dominio particular (textos, diccionarios, páginas web, etc.). Y por ultimo esta la definición de reglas, para derivar hechos que no están explícitamente expresados en la ontología, como por ejemplo: "*toda frase nominal tiene un sustantivo*".

[34] Una definición intencional abstrae las características de los objetos de estudio, o plantea las restricciones que deben cumplir los individuos para ser parte de un concepto.
[35] Una definición extensional hace explicito los objetos de estudio, los individuos que cumplen ciertas propiedades o restricciones

3.2 VISIÓN GENERAL DE LA ARQUITECTURA DE APRENDIZAJE AUTOMÁTICO DE ONTOLOGÍAS.

Tal como se ha mencionado en el resumen y en la introducción, el objetivo del componente de aprendizaje automático de ontologías es enriquecer las estructuras definidas del MODS, para potenciar su capacidad de interpretación de una consulta en lenguaje natural en español. De su capacidad de aprendizaje y adaptabilidad al perfil del usuario y a la dinámica de la web semántica, depende en gran parte la evolución y éxito del MODS.

Para lograr que el MODS aprenda, el componente de aprendizaje debe soportar la adquisición automática de conocimiento léxico y semántico: términos (palabras), conceptos (palabras con significado definido), de relaciones (taxonómicas, no-taxonómicas), de instancias (individuos), de reglas (sistemas deductivos), o de otras ontologías (o parte de estas); todas ellas a ser incorporadas de manera correcta en las respectivas estructuras del MODS: ontología interpretativa, ontología lingüística, lexicón, u ontología de dominio.

Una aproximación sistemática de las sub-tareas antes mencionadas, que constituyen el proceso de aprendizaje automático de ontologías creado para dar apoyo y respaldo al MODS, se ilustra en la siguiente figura.

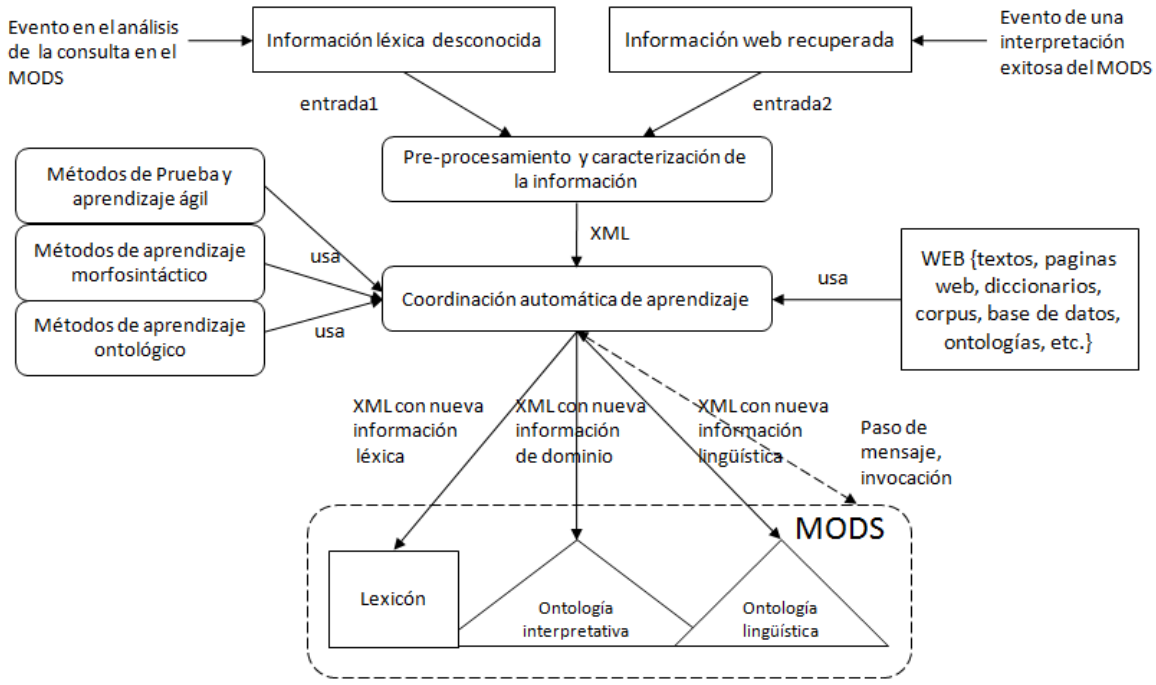

Figura 6. Arquitectura general del componente de aprendizaje automático de ontologías.

El sistema recibe dos entrada producidas por el MODS por *dos eventos diferentes*: Un primer evento ocurre en el análisis de la consulta, cuando se encuentra frente a un término desconocido por el lexicón del MODS. Cuando esto ocurre se genera la *entrada1*, que sería el término desconocido junto con la probable categoría léxica. Un segundo evento ocurre cuando el proceso de interpretación de la consulta es exitoso. Cuando esto ocurre se genera la *entrada2,* que sería la información recuperada de la Web vía la interpretación de la consulta. La información recuperada puede traer consigo información heterogénea, tal como: textos, diccionarios, bases de conocimiento, información semi-estructurada (anotaciones RDF de recursos Web, esquema XML), información relacional, ontologías (archivos OWL), entre otros.

La entrada1 y/o la entrada2 pasan por un modulo de *pre-procesamiento y caracterización de la entrada*. El preprocesado para el caso de la entrada1 consiste en analizar la estructura interna del término desconocido: raíz, afijos, forma canónica, etc. Para el caso de la entrada2 consiste en ajustar la información recuperada para poder extraer información relevante de ella. Por ejemplo, para

documentos semi-estructurados, como diccionarios, su transformación en una estructura relacional predefinida [40]; en el caso de los documentos HTML, su indexación y transformación en texto libre [41].

Una vez se tiene la información de interés de las entrada1 y entrada2, se define un archivo XML diferente para cada entrada. El archivo XML definido para la caracterización de la *entrada1* es:

```
<XML>
        <entrada1>
         <metodo id="aprendizaje_lexmor"/>
          <termino> termino desconocido</termino >
          <pre_procesado>
                <forma_canonica></forma_canonica>
                <prefijo></prefijo>
                <sufijo></sufijo>
           </pre_procesado >
          </entrada1>
</XML>
```

El archivo XML definido para la caracterización de la *entrada2* es:

```
   <XML>
        <entrada2>
                <metodo id="aprendizaje_sem"/>
                <data> información web recuperada  </data >
                <pre_procesado>
                        <texto></texto>
                        <esquema_relacional></esquema_relacional>
                        <esquema_bc>
                                <tbox></tbox>
                                <abox></abox>
                        </esquema_bc>
                        <esquema_onto>
                                <tax_conceptual></tax_conceptual>
                        </esquema_onto>
                        …
                 </ pre_procesado>
        </entrada2>
 </XML>
```

La información caracterizada entra al *modulo de coordinación automática para el aprendizaje (o motor de aprendizaje),* donde se realiza la adquisición de conocimiento, mapeando los archivos XML con los respectivos métodos/técnicas de descubrimiento de conocimiento, y produce como salida otro archivo XML con el conocimiento aprendido. El mapeo se lleva a cabo vía el valor del campo de la etiqueta <método id=" ">, a través del cual se asigna el método de aprendizaje correcto. El archivo XML definido para el aprendizaje a partir de la *entrada1* es:

```
<XML>
        <aprendizaje_lexmor>
                <categoria></categoria>
                <tipo></tipo>
                <genero> </genero>
                <numero> </numero>
                <modo> </modo>
                <tiempo> </tiempo>
                <aspecto> </aspecto>
                <voz> </voz>
                <persona> </persona>
        </aprendizaje_lexmor>
</XML>
```

El archivo XML definido para el aprendizaje a partir de la *entrada2* es:

```
<XML>
        <aprendizaje_sem>
                <concepto></concepto>
                <relacion_tax><relacion_tax>
                <relacion_ntax><relacion_ntax>
                <regla></regla>
                <individuo></individuo>
        </aprendizaje_sem>
</XML>
```

El nuevo conocimiento descubierto se incorporara en las estructuras léxicas y semánticas del MODS respectivas. Los *métodos/técnicas de aprendizaje/descubrimiento de conocimiento* morfosintáctico y ontológico están definidos en función del objeto de aprendizaje y la fuente a usar para el mismo. Estos métodos de aprendizaje son el elemento fundamental para la adquisición de

conocimiento, y comprenden un conjunto de algoritmos especializados que realizan tareas de extracción, descubrimiento, importación y búsqueda de un conocimiento particular, de tal manera a aprender conceptos, relaciones, términos, ontologías (o parte de ella), etc. Más adelante serán presentados los considerados en este trabajo.

Además del aprendizaje, el componente puede notificar sobre posibles casos particulares en el proceso de aprendizaje. Por ejemplo, para el caso del aprendizaje léxico, puede ocurrir la llegada de una palabra mal escrita o no válida en el lenguaje español, para cuya salida se genera una *notificación del error*. Otro caso deviene con una palabra rara/extraña o desusada, un anglicismo, o una palabra coloquial, en cuyo caso se devuelve, si se encuentra, un documento que hable de ello. Soportar este tipo de fenómenos lingüísticos permite, entre otras cosas, una interoperabilidad y/o retroalimentación entre el usuario y el MODS.

3.2.1 Aprendizaje semántico a partir de diferentes métodos y fuentes

El método de aprendizaje de ontologías a usar o desarrollar, depende del dominio de la ontología a desarrollar o a enriquecer. Es posible aprender de diferentes fuentes un mismo elemento ontológico (conceptos, relaciones, etc.,), lo que cambia es la técnica, método o enfoque (el paradigma).

Tabla 13. Clasificación de XML para el aprendizaje semántico a partir de deferentes métodos y fuentes.

Método <metodo id =" ">	Fuente <data>	<Aprendizaje>	Impacta
metodo id="aprendizaje_rel"	Esquema relacional	Concepto, relaciones, individuos, reglas	Ontologías dominio y/o lingüística
metodo id="aprendizaje_bc"	Esquema de conocimiento y aseverativo (axiomas)	Concepto, relaciones, individuos, reglas	Ontologías dominio y/o lingüística

metodo id=”aprendizaje_on”	Esquema conceptual	Concepto, relaciones, individuos, reglas	Ontologías dominio y/o lingüística
metodo id=”aprendizaje_tex”	Texto libre	Concepto, relaciones, individuos, reglas	Ontologías dominio y/o lingüística

En la primera columna de métodos el valor de id permite identificar el conjunto especializado de algoritmos para el aprendizaje de elementos ontológicos a partir de una fuente dada. Por ejemplo, para el método id = “aprendizaje_tex” es posible contar con algoritmos para el aprendizaje de conceptos a partir de textos, algoritmos para el aprendizaje de relaciones taxonómica a partir de textos, etc. Igual para los otros id. La segunda columna especifica la fuente de aprendizaje a usar por los algoritmos de aprendizaje. La columna de aprendizaje dice que elementos ontológicos se pueden aprender. Por último se indica que ontología se impactaría. Esto último está ligado con el dominio y el elemento ontológico de interés. Estos métodos son posibles de ser extendidos para otras fuentes de aprendizaje.

3.2.2 Aprendizaje morfosintáctico del MODS

En cuanto al aprendizaje morfosintáctico, el MODS espera un conjunto de conocimientos según la categoría lingüística, de acuerdo con la siguiente tabla 14.

Tabla 14. Requerimientos de aprendizaje de información morfosintáctica

Categoría	Genero	Numero	Tipo	Modo	Tiempo	Aspecto	Voz	Persona
Sustantivo	X	X	X					
Adjetivo	X	X	X					
Adverbio			X					
Verbo		X	X	X	X	X	X	X
Preposición								
Conjunción								

La estructura o interfaz definida para esta información es como sigue:

lex_mor[36](*componente léxico, categoría, tipo, genero, número, modo, tiempo, aspecto, voz, persona, instancia_ontologia_linguistica*[37]).

Por ejemplo, para un nombre o sustantivo desconocido como *María*, la entrada para el componente de aprendizaje seria:

lex_mor(***María, desconocido****, null, null, null, null, null, null, null, null)*.

Para la cual se espera que el componente de aprendizaje entregue la siguiente información

lex_mor(*María, **sustantivo, femenino, singular**, null, null, null, null, null, null)*.

Es importante aclarar que esta salida no es la misma para el aprendizaje semántico ya que se aprenden cosas distintas, puesto que ocurre bajo otro evento diferente. El aprendizaje semántico es visto más como una manera de explotar una potencial fuente de aprendizaje, como lo es la información recuperada de la Web, que un requerimiento directo del MODS.

3.3 DETALLES DEL COMPONENTE DE APRENDIZAJE

El macro-algoritmo del componente de aprendizaje se presenta a continuación:

Entrada: requerimiento léxico, recursos web
Salida: nueva información léxica, nueva información semántica, mensajes de error

Inicio
Recibir entrada

Procesar y caracterizar entrada
{Entrada: cadena
Salida: XML *con data preprocesada y método de aprendizaje identificado*}

[36] Lex_mor es el nombre definido para la estructura definida para el lexicón del MODS
[37] es el puente o interfaz a través del cual comunicar o conectar los procesos de análisis morfosintáctico y semántico.

If (entrada1)

Leer componente léxico

Select case componenten_lexico

case verbo, sustantivo, adjetivo, case adverbio: hallar forma canónica, prefijo, sufijo

crear_xml (aprendizaje_lexmor, componente léxico, categoria, forma canónica, prefijo, sufijo)

else if (entrada2)

Métodos para clasificar recursos recuperados de la web según se estructura

Mientras haya recursos

Leer recurso

Select case estructura del recurso

Case estructurado: {por ejemplo ontologías OWL}

procesar recurso y descubrir información objetivo esquema conceptual

crear_xml (aprendizaje_on, taxonomía de conceptos)

{es posible crear otro XML diferente y ello depende del recurso, la información a descubrir en el recurso y el método aplicar}

Case semi-estructurado: {por ejemplo un archivo HTML, documento}

procesar recurso y descubrir información objetivo texto

crear_xml (aprendizaje_tex, texto)

{es posible crear otro XML diferente y ello depende del recurso, la información a descubrir en el recurso y el método aplicar}

Case desestructurado: {como texto libre}

procesar recurso y descubrir información objetivo texto

crear_xml (aprendizaje_tex, texto)

{es posible crear otro XML diferente y ello depende del recurso, la información a descubrir en el recurso y el método aplicar}

Fin mientras

Fin de procesar

motor_aprendizaje

{Entrada: XML

Salida: XML de aprendizaje}

Leer etiqueta <metodo id=" "/>

Select case id

Case *aprendizaje_lexmor*:

Tomar valor de etiquetas <termino></termino>, <categoria><categoria>, <forma_canonica></forma_canonica>

Usa métodos de prueba y aprendizaje ágil y

Devuelve XML de aprendizaje lexmor o mensaje de resultados

Case "*aprendizaje_rel*":

Select case elemento_a_aprender

Case concepto:

Métodos de aprendizaje de conceptos a partir de esquema un esquema relacional

conceptos aprendidos

Case relacion_tax:

Métodos de aprendizaje de relacion_tax a partir de Esquema un esquema relacional

Relaciones taxonómicas aprendidas

Case relacion_notax:

Métodos de aprendizaje de relacion_notax a partir de Esquema un esquema relacional

Relaciones no-taxonómicas aprendidas

Case reglas:

Métodos de aprendizaje de reglas a partir de Esquema un esquema relacional

Reglas aprendidas

crear xml de aprendizaje sem (conceptos, relacion tax, relacion_ntax, reglas, null)

Case "*aprendizaje_bc*":

Select case elemento_a_aprender

Case concepto:

Métodos de aprendizaje de conceptos a partir de esquema terminológico

conceptos aprendidos

Case relacion_tax:

Métodos de aprendizaje de relaciones taxonómicas a partir de esquema terminológico

Relaciones taxonómicas aprendidas

Case relacion_notax:

Métodos de aprendizaje de relacion_notax a partir de Esquema un esquema terminológico

Relaciones no-taxonómicas aprendidas

Case reglas:
Métodos de aprendizaje de reglas a partir de
Esquema un esquema terminológico
Reglas aprendidas
Case individuos:
Métodos de población de ontologías a
partir de un esquema aseverativo
individuos

crear xml de aprendizaje sem (conceptos, relacion tax, relacion_ntax, reglas, individuos)

Case "*aprendizaje_tex*":

Select case elemento_a_aprender
Case concepto:
Métodos de aprendizaje de conceptos a partir de texto
conceptos aprendidos
Case relacion_tax:
Métodos de aprendizaje de relaciones taxonómicas a partir de texto
Relaciones taxonómicas aprendidas
Case relacion_notax:
Métodos de aprendizaje de relacion_notax a partir de texto
Relaciones no-taxonómicas aprendidas
Case reglas:
Métodos de aprendizaje de reglas a partir de
texto
Reglas aprendidas
Case individuos:
Métodos de población de ontologías a
partir de texto
individuos
crear xml de aprendizaje sem (conceptos, relacion tax, relacion_ntax, reglas, individuos)
Case "{Extensible a otros recursos y métodos de aprendizaje}".
Fin de motor de aprendizaje

A continuación la tabla 15 presenta, de manera resumida y unificada, las diversas aproximaciones para el aprendizaje de diferentes elementos ontológicos, que están relacionadas con el presente trabajo. La primera columna contiene la fuente de aprendizaje, con tres principales fuentes de información: raíces de las palabras, información recuperada de la web, y ontologías de dominio, Le sigue el objetivo del proceso aprendizaje, que puede ser aprender nuevas palabras, nuevos conceptos y relaciones, tanto taxonómicas como no taxonómicas, construir axiomas formales, o poblar las ontologías. Otros objetivos de aprendizaje serian los procesos de fusión entre las ontologías del MODS y ontologías externas. En el caso de la poda, puede realizarse en las ontologías de MODS como un proceso de refinamiento y purificación de obsoletos elementos ontológicos, tales como conceptos en desuso, o cambios de relaciones entre conceptos, y en las ontologías externas como mecanismo de partición de las partes ontológicas de interés que serán usadas por el MODS. En este trabajo no son estudiados estos objetivos. La columna de las técnicas agrupa las más usadas para cada uno de estos objetivos. Por último, se identifican las estructuras del marco ontológico del MODS que afectaría cada proceso de aprendizaje. Esta tabla es usada como la base para construir el motor de razonamiento.

Tabla 15. Métodos y técnicas para el componente de aprendizaje del MODS

Fuente de aprendizaje	Objetivo de aprendizaje	Técnica de aprendizaje usada	Estructura de conocimiento impactada
Raíz de palabra,	Nuevas palabras flexivas	Técnica de verificación de nuevas palabras	Lexicón
Texto plano de dominio	Nuevos conceptos	Identificación temática, distancia semántica, técnicas de agrupamiento enfoque estadístico	Ontología lingüística Meta-ontología interpretativa
Ontologías de dominio	Partes de ontologías u ontologías	Fusión de ontologías, Poda de ontologías,	Ontología lingüística Meta-ontología interpretativa
Texto plano de dominio	Aprendizaje de instancias o población de las ontologías	Identificación por nombres de entidades en español (ej. Nombre propios en mayúscula) +Patrones lingüísticos, o por categorías predefinidas , tesauros o basada en casos	Onomasticon
Texto de dominio y Ontologías de dominio	Descubrir nuevas relaciones	Técnicas linguisticas, Mapeo, PLN	Ontología lingüística Meta-ontología Ontología Lingüística
Texto de dominio y ontologías de dominio	Aprender conceptos y nuevas relaciones entre ellos.	Análisis lingüístico y técnicas de agrupamiento	Ontología lingüística Meta-ontología Interpretativa
Texto plano de dominio	Aprender nuevos conceptos	Concepto de hipótesis basados en la calidad lingüística y conceptual de las etiquetas	Ontología lingüística Meta-ontología Interpretativa
Texto plano de dominio y Ontologías de dominio	Axiomas lógicos	Técnicas inductivas, deductivas , abductivas	Ontología lingüística Meta-ontología Interpretativa

De todos los procesos de aprendizaje presentados en el macro-algoritmo y la tabla 15, en este trabajo se desarrollara el modulo de aprendizaje léxico para la entrada1, que se corresponde con la primera fila de la tabla 15. También se describirá un método para el aprendizaje de relaciones no taxonómicas, que tiene que ver con la entrada2.

3.3.1 Detalles del aprendizaje morfosintáctico

El componente de aprendizaje morfosintáctico tiene por objetivo enriquecer el lexicón definido para el MODS. Recibe una cadena que contiene una palabra desconocida, y proporciona una de las siguientes salidas:

1. Un archivo XML de aprendizaje lexmor con nueva información léxica descubierta
2. Un mensaje de error, que puede venir acompañado por un archivo con evidencias de uso en la web de palabras muy semejantes a la buscada.

El componente de aprendizaje morfosintáctico lo constituyen cuatro módulos (ver figura 7): un primero modulo de pre-procesamiento y caracterización, denominado aprendizaje simiente (por proporcionar la semilla para el aprendizaje), un segundo módulo llamado aprendizaje ágil (por consultar información directamente desde la web), un tercer modulo llamado aprendizaje duro (por requerir procesamiento del lenguaje natural), y un último modulo de enriquecimiento, refinamiento o actualización.

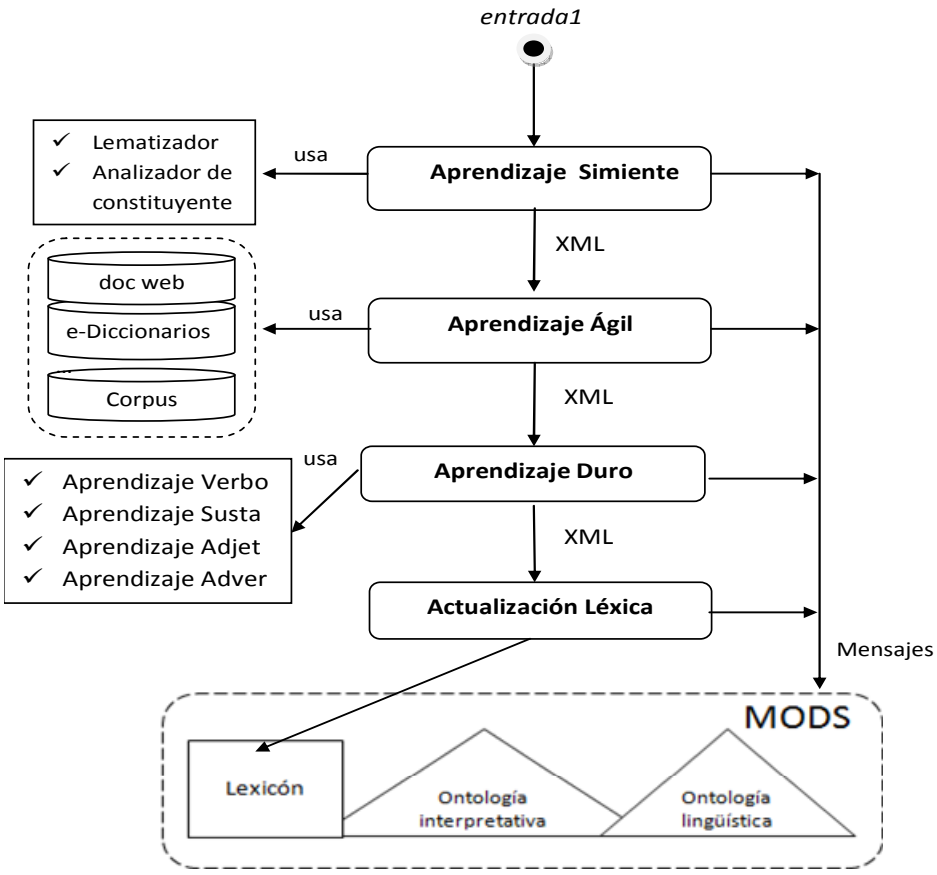

Figura 7. Arquitectura aprendizaje morfosintáctico del componente léxico

El modulo de *aprendizaje simiente* recibe la *entrada1* y halla la información morfosintáctica básica (semilla) del término, tal como su forma canoníca, afijos (prefijos y sufijos) para una palabra simple, o las sub-palabras si se tratase de un término compuesto, tal como por ejemplo “Universidad de los Andes”. Con esta información crea el archivo XML que la caracteriza. La forma canoníca de un verbo en infinitivo es él mismo, y para un verbo conjugado su forma infinitiva; para un sustantivo, adjetivo o adverbio en plural, su forma canónica es su singular. El modulo usa una librería de algoritmos diseñados para tales fines: un lematizador[38] y un analizador de constituyentes, respectivamente. En el caso de los adverbios, muchos son derivados de los adjetivos, tal como se muestra a continuación.

Tabla 16. Formación de adverbios

Adverbio	Adjetivo Básico	Singular, forma femenina
felizmente	feliz	feliz
fácilmente	fácil	fácil
alegremente	alegre	alegre
nuevamente	nuevo	nueva
rápidamente	rápido	rápida
Suavemente	Suave	suave

Es así que para el aprendizaje simiente de un adverbio se separa el sufijo del núcleo. En cuanto a los adjetivos, son palabras que nombran o indican cualidades, rasgos y propiedades, de los nombres o sustantivos a los que acompañan. Por ejemplo, en “La belleza de la flor *natural* es *insuperable*. Esta *vistosa* flor alegra nuestros sentidos”. El tratamiento de aprendizaje simiente para la palabra-adjetivo consiste en aprender su forma canónica.

Las demás unidades léxicas de orden inferior, tales como pronombres, determinantes etc., se encuentran predefinidos en MODS, y no son objeto de aprendizaje.

[38] Un lematizador permite calcular la forma canónica de una palabra

Después sigue el módulo de *aprendizaje ágil,* cuyo objetivo es validar y aprender información gramatical en función del término. Para ello recibe el archivo XML generado en el aprendizaje simiente, y devuelve alguna de las siguientes salidas (ver figura 8).

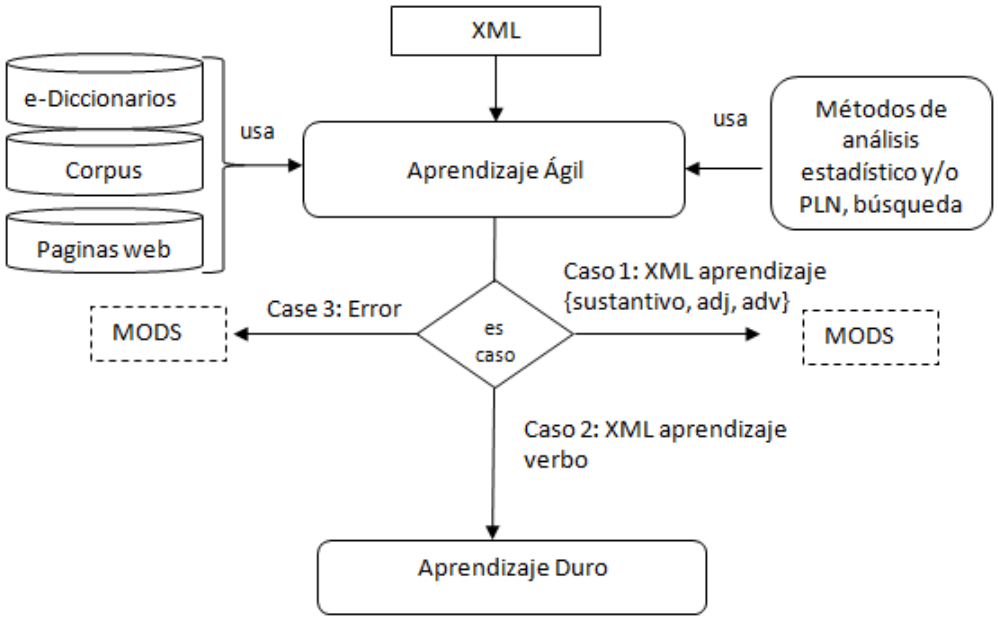

Figura 8. Módulo de aprendizaje Ágil

- Casos 1: Ocurre cuando la categoría descubierta de la palabra desconocida es sustantivo, adjetivo o adverbio. En estos tres casos, el XML aprendizaje contiene además la información gramatical de la palabra

- Caso 2: Ocurre cuando la categoría de la palabra descubierta es un verbo y el XML de aprendizaje es enriquecido con la información gramatical propia del verbo.

- Caso 3: Ocurre cuando no se ha encontrado ninguna información relacionada con la palabra desconocida (por ejemplo, cuando se tienen palabras mal escritas, no se tienen referentes de su uso en el español, etc.).

La validación puede ser sencilla o rigurosa, dependiendo de la técnica. Puede ser suficiente con encontrar información gramatical en diccionarios online a través de

búsquedas directas, usar métodos estadísticos en corpus de documentos o en la web, entre otros.

El modulo se inicia al recibir el archivo XML proveniente del aprendizaje simiente:

```
<XML>
        <entrada1>
         <metodo id="aprendizaje_lexmor"/>
          <termino> termino desconocido</termino >
          <categoria> </categoria>
          <pre_procesado>
                <forma_canonica></forma_canonica>
                <prefijo></prefijo>
                <sufijo></sufijo>
           </pre_procesado >
       </entrada1>
</XML>
```

Después lee la información que le interesa

- Los valores de las etiquetas <termino> </termino> y <categoria> </categoria>, para buscar en los diccionarios on-line con el fin de validar dicha información, o
- El valor de la etiqueta <forma_canonica></forma_canonica>, para descubrir y recoger nueva información gramatical con la cual cargar el XML de aprendizaje lexmor:

```
<XML>
        <aprendizaje_lexmor>
                <genero> </genero>
                <numero> </numero>
                <modo> </modo>
                <tiempo> </tiempo>
                <aspecto> </aspecto>
                <voz> </voz>
                <persona> </persona>
        </aprendizaje_lexmor>
</XML>
```

De manera general, el aprendizaje ágil es como sigue: una vez se ha leído el XML de entrada un primer paso consiste en usar métodos de comprobación directa y

extracción de información de diccionarios online. Si es exitosa la operación, entonces se carga el XML de aprendizaje lexmor con la información encontrada. Si no se encuentra información sobre la palabra en los diccionarios, entonces se pasa a buscar en corpus de documentos a través de métodos de comprobación y búsqueda indirecta y formal. Es indirecto en la medida que hay que realizar procedimiento de análisis más complejos que simples consultas, como se realizaron en la comprobación directa a los diccionarios. Es formal en la medida que trabaja con colecciones de documentos consensuados y representativos de un dominio, como lo son los corpus de documentos. Si hay aprendizaje se actualiza el XML lexmor con la información descubierta, sino, como última instancia se va a la web en general con el fin de encontrar alguna pagina web o documento que hable sobre el término. En caso de encontrarse algo se recupera la fuente y se envía al MODS.

El modulo que continua en el aprendizaje morfosintáctico es el *aprendizaje duro.* Este modulo está pensado para soportar el aprendizaje morfosintáctico de verbos, sustantivos, adjetivos, y adverbios. Este módulo de aprendizaje consiste en adquirir nueva información léxica, adicional a la palabra desconocida, tal como la conjugación del verbo, la flexión diminutiva o aumentativa de un sustantivo, etc.

Ahora bien, una de las tareas más complejas dentro del aprendizaje morfosintáctico es el aprendizaje de verbos, por eso nos concentraremos en él en este trabajo. El verbo es el núcleo del lenguaje español, cuenta con aproximadamente 100 modelos de conjugación para un grupo de unos 15.000 mil verbos, los cuales, al parecer, usan caprichosamente uno u otro de estos modelos.

Afortunadamente, existe una propuesta de modelización de los verbos en español, de manera tal que el sistema de verbos para el español se reduce a una serie de reglas que, combinadas describen todos los verbos españoles [42]. Estas reglas describen los patrones de comportamiento de los verbos y los rasgos comunes que estos comparten.

A partir de esta propuesta, se ha desarrollado para el aprendizaje duro un conjunto de algoritmos que pueden diferenciar entre verbos regulares e irregulares, y desplegar la conjugación de los verbos regulares del español, y parte de los irregulares, a partir de su forma en infinitivo.

En general, el verbo es una categoría léxica que, al conjugarse, puede variar en persona, número, tiempo, modo, aspecto y voz. Desde un punto de vista morfológico, el verbo consta de dos partes:

- la raíz, que es la parte que nos aporta el significado de verbo, tal y como lo podemos encontrar en el diccionario (información léxica).
- las desinencias, que son los morfemas flexivos que nos dan la información referente a la persona, número, tiempo, modo, aspecto y voz (información gramatical).

De este modo, en una forma verbal como *amo* se pueden distinguir las siguientes partes: am-, raíz cuyo significado es 'tener amor a alguien o algo', y la desinencia -o que nos revela que la forma verbal amo se corresponde con la primera persona (persona) del singular (número), del presente (tiempo) de indicativo (modo) de la voz activa.

Este modulo, por su complejidad, se describe en detalle en un capítulo aparte (capítulo IV), donde se estudian los múltiples fenómenos lingüísticos que existen y la manera computacional de afrontarlos.

Finalmente, el modulo de *Actualización del lexicón* recibe como entrada la información que describe la palabra desconocida aprendida de los módulos de aprendizaje precedentes. Esta información esta formateada en un archivo XML. La tarea de este modulo consiste en incluir la nueva información léxica en los argumentos vacios del lex_mor, o construir un nuevo lex_mor, si no existe en el lexicón, con la nueva información descubierta de la palabra desconocida. Los posibles casos de actualización del lexicón que se presentan son:

Caso 1: lex_mor con argumentos/atributos incompletos.

lex_mor(componente léxico, categoría, genero, numero, instancia_ontologia_lingüística)

Ejemplo de Lex_mor incompleto:

lex_mor(*Fernando*, categoría=sustantivo, **genero=null,** numero=singular, instancia_ontologia_lingüística=null)

La tarea del modulo de refinamiento es llenar o completar el lexico_mor con el valor aprendido.

Lex_mor actualizado: lex_mor(Fernando, categoría=sustantivo, **genero=masculino**, numero=sungular, instancia_ontologia_lingüística=null)

Caso 2: lex_mor no registrado en el lexicón.

En este caso no existe un registro de lex_mor para el componente léxico de entrada, y hay que crearlo con la información aprendida.

lex_mor(componente léxicox, categoríax, generox, numerox, instancia_ontologia_lingüísticax).

Este lex_mor puede ser construido de manera incompleta, y quedar en el caso 1, o completa con todos sus argumentos cargados.

3.3.2 Aprendizaje semántico: caso de relaciones no taxonómica

Con respecto al aprendizaje semántico, en la presente tesis se profundizara sólo en el aprendizaje de relaciones no-taxonómicas, por ser unas de las menos estudiadas en el campo del aprendizaje de ontologías, aún cuando ellas son un punto importarte en el aprendizaje de ontologías [43]. En esta sección se describe un método para la adquisición automática de relaciones no-taxonómicas, usando como base de aprendizaje la información recuperada desde la web por el MODS. Esta tarea involucra:

i) El descubrimiento de patrones semánticos usados para expresar relaciones no-taxonómicas en un dominio específico (frases verbales). Para ello se usan patrones tales como:

Patrón 1: $Termino_0$ … *tales como* … {$Termino_1$, $Termino_2$, (y/o), …, $Termino_n$}.

Lo anterior implica que para todo $Termino_i$, 1 <=i <=n, entonces hiponimo[39]($Termino_i$, $Termino_0$). Por ejemplo, en

"…merecen citarse los que están situados en el segmento basal de las primeras antenas de los decápodos, ***tales como*** *los vulgares cangrejos de rio y de mar"*[40]

hiponimo(cangrejo_de_rio, decápodo),
hiponimo(cangrejo_de_mar, decápodo).

Patrón 2: *de* T[41] *como* {T, }* {(o/y)} T

Ejemplo de la vida real es: *"obras* ***de*** *autores* ***como*** *Gabriel García Márquez, Pablo Neruda, y Ernesto Sábato"*

Patrón 3: T {,T}* {,} *u otro((a)-s)* T

Ejemplo:: *"…Contusiones, heridas, huesos rotos,* ***u otras*** *lesiones. . ."*.

ii) la recuperación desde otra fuente externa (puede ser un corpus) de relaciones candidatas, a partir del conocimiento adquirido (patrones),

iii) la selección de los conceptos y relaciones más adecuados para ser incorporados en las ontologías.

A continuación se describen, de manera general, algunas de las técnicas usadas para la adquisición de relaciones no-taxonómicas desde la web.

[39] Un hipónimo es aquella palabra que posee todos los rasgos semánticos de otra más general. Ejemplo "mañana", "tarde" y "noche" son hipónimos de día, o lunes, o martes etc.

[40] REAL ACADEMIA ESPAÑOLA: Banco de datos (CORDE) [en línea]. *Corpus diacrónico del español.* <http://www.rae.es> jueves 8 de julio 2010.

[41] T es término

- Técnicas de análisis ligero [44]: estas técnicas son usadas sólo para aquellas partes de texto que presentan el conocimiento de una manera simple, directa, clara (sin ambigüedades), reduciendo la cantidad de información a procesar.

- Técnicas de análisis estadístico: estas técnicas normalmente se aplican en las tareas de adquisición de conocimiento si se cuenta con un buen volumen de información de donde obtener medidas estadísticas relevantes. La web, como es sabido, representa una buena parte de la información producida por las personas, lo cual ofrece un buen volumen de información, además, los motores de búsqueda pueden ofrecer medidas de confianza. Todo lo anterior permite obtener estadísticas de la distribución de la información en la web.

- El uso de patrones lingüísticos: puede ser una técnica eficiente para extraer conocimiento sin la supervisión del ingeniero ontológico. Para el caso de relaciones taxonómicas, patrones lingüísticos independientes del dominio son una manera muy común para descubrir las relaciones. Ejemplo de ellos son los fenómenos semánticos tal como la meronimia[42], antonimia[43], sinonimia[44], etc. Ejemplos de patrones de meronimia para el español son *El X tiene Y y Z, Y y Z son parte de X, X consta de Y y Z, Y y Z forman parte de Y, El Y y Z de un X*.

La figura 9 presenta el macroalgoritmo del proceso de aprendizaje de relaciones no taxonómicas a partir del uso de patrones lingüísticos (para el resto de las formas de adquisición de relaciones no-taxonómicas desde la web, debería hacerse su especifico macroalgoritmo).

[42] La meronimia es una relación semántica no-simétrica entre los significados de dos palabras dentro del mismo campo semántico

[43] Son las palabras que dan nombre a realidades opuestas y, por tanto, expresan significados contrarios, como miedo y valor.

[44] Son las palabras que nombran o se refieren a una misma realidad y, por tanto, expresan un mismo significado.

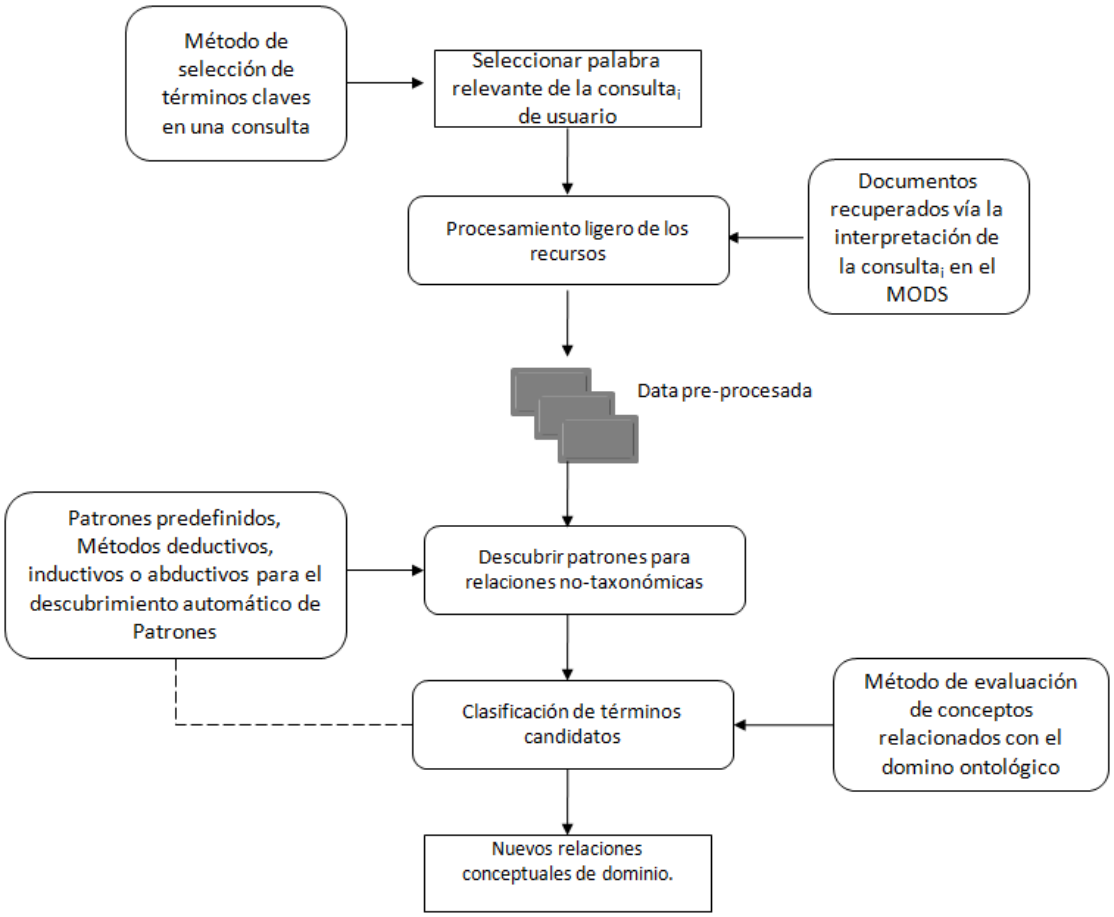

Figura 9. Proceso de aprendizaje de relaciones no taxonómicas a partir del uso de patrones lingüísticos

En específico, en la figura 9 el primer paso es descubrir patrones lingüísticos que expresen relaciones no-taxonómicas. En este caso, las relaciones son normalmente expresadas por un verbo que relaciona dos conceptos. Debido a la gran cantidad de verbos para el español, se recomienda encontrar los más relevantes para un dominio particular (por ejemplo para el dominio del "Sistema de Investigación de la Universidad de los Andes" una palabra clave inicial podría ser *investigación*). De acuerdo con esto, una consulta con la palabra clave es pasada a un motor de búsqueda, quien recupera un conjuntos de documentos que cubren al dominio específico (esto, si el motor de búsqueda usará algo parecido al MODS, puesto que los comunes aun traen mucho ruido e información irrelevante). A cada uno de estos documentos se les hace un análisis ligero, considerando

aquellos términos más próximos que acompañan a la palabra clave inicial, con el fin de encontrar frases verbales (verbos conjugados, y opcionalmente, proposiciones), formando una lista de candidatos. Estos candidatos son clasificados en función de su posición dentro de las sentencias que contienen la palabra clave inicial: *predecesores* (ejemplo, "…*trabaja con investigación"*), sucesores (ejemplo, "*investigación trata con* …"). Cada candidato es evaluado en orden a decidir si está realmente bien relacionado al dominio. Como la base del análisis ligero puede venir dado por una medida estadística, se pueden considerar medidas de co-ocurrencia entre las frases verbales y las palabras claves del dominio como una medida de relación entre ellas. Para ello es posible usar escalas de medida estadística dada por la web, que representa la distribución de un concepto en toda la web. Más concretamente, para cada frase verbal candidata, que ha sido extraída como un predecesor de la palabra clave inicial, es posible computar y clasificar) las relaciones preguntado por el número de hits[45] devueltos por un motor de búsqueda para la siguiente consulta:

Score(fraseVerbal/claveInicial) = hits("fraseVerbal claveInicial")/hits(fraseVerbal)

De igual manera, si el candidato ha sido extraído como un sucesor de la palabra clave inicial, se calcula la misma fórmula, sólo que esta vez con el orden inverso de la correspondiente pregunta *hits("claveInicial fraseVerbal").* Estas formulas son usadas para calcular el grado de relación entre dos palabras. Los valores devueltos son usados para clasificar la lista de patrones lingüísticos candidatos (frases verbales), y seleccionar aquellos que están más estrechamente relacionados.

Una vez se han obtenido los patrones lingüísticos, el siguiente paso es usarlos para descubrir conceptos que están relacionados no-taxonómicamente con la palabra clave inicial. Estos nuevos conceptos se postulan como candidatos para ser relacionados no-taxonómicamente con la palabra clave inicial, etiquetando esta relación con la frase verbal.

[45] HITS (acrónimo del inglés Hypertext Induced Topic Selection) es un algoritmo diseñado para valorar, y de paso clasificar, la importancia de una página web.

Luego se decide nuevamente cuales de los conceptos extraídos están estrechamente relacionados al dominio de búsqueda. Con el fin de realizar el proceso de selección se usa nuevamente la medida estadísticas escalables web acerca de la co-ocurrencia de estos dos nuevos términos. En este caso, el score es calculado de la siguiente manera:

Score(concepto/claveInicial) = hits("claveInicial" AND "concept") / hits("concepto")

El AND asegura que estos dos términos co-ocurran dentro del texto, pero no necesariamente en la misma sentencia u oración. Aquellos conceptos cuya relación es superior a un umbral son seleccionados e incorporados a la ontología, con una relación que es etiquetada de acuerdo a la frase verbal usada para descubrirlos.

4. CONSIDERACIONES PARA EL APRENDIZAJE MORFOLÓGICO DE VERBOS

4.1 ANÁLISIS DEL VERBO

El verbo es una categoría léxica que, al conjugarse, puede variar en persona[46], número[47], tiempo[48], modo[49], aspecto[50] y voz[51]. Desde un punto de vista morfológico, el verbo consta de dos partes:

- la raíz, que es la parte que nos aporta el significado de verbo tal y como lo podemos encontrar en el diccionario (información léxica).

- las desinencias, que son los morfemas flexivos que nos dan la información referente a la persona, número, tiempo, modo, aspecto y voz (información gramatical).

De este modo, en una forma verbal, como *amo,* se pueden distinguir las siguientes partes: *am-*, raíz cuyo significado es '*tener amor a alguien o algo*', y la desinencia[52] *-o* que nos revela que la forma verbal *amo* se corresponde con la *primera persona*

[46] La persona es la categoría gramatical cuyo significado básico consiste en referir a las entidades participantes en el acto comunicativo.
[47] El número es una categoría gramatical que se asocia con la cuantificación y abarca dos distinciones básicas: el singular y el plural.
[48] El tiempo es una categoría gramatical que sitúa el evento denotado por el verbo en un espacio temporal anterior, simultáneo o posterior al punto de referencia.
[49] El modo es la categoría gramatical cuyo contenido se ha asociado a la actitud subjetiva que se adopta ante el contenido proposicional de enunciado.
[50] Se trata de una categoría semántica que no influye en la conjugación del verbo español.
[51] El concepto de voz alude a una categoría que expresa las diferentes relaciones que se establecen entre el verbo y las funciones sintácticas.
[52] la desinencia son los sufijos o terminaciones que se agregan al núcleo de una palabra. En las desinencias se pueden apreciar dos partes: una que expresan el tiempo, aspecto y el modo y otra para el número y la persona

(persona) del *singular* (número), del *presente* (tiempo), del *indicativo* (modo) de la *voz activa*.

Existen formas del verbo que tienen como rasgo común el no presentar formas flexivas correspondientes a la categoría persona. Estas formas se denominan formas no personales (frente al resto que son formas personales), y son el infinitivo, el participio y el gerundio.

Tabla 17. Formas no personales de un verbo

Conjugación	Infinitivo	Participio	Gerundio
Primera Conjugación.	Cantar	Cantado	Cantando
Segunda Conjugación.	Comer	Comido	Comiendo
Tercera Conjugación.	Vivir	Vivido	Viviendo

Diferentes a la categoría voz, que expresa las diferentes relaciones que se establecen entre el verbo y las funciones sintácticas

Tabla 18. Flexiones de voz

Modo	Sentidos	Ejemplo
Indicativo	Aserción y realidad: asume como real el estado de cosas denotado por la predicación.	Voy a la ULA
Subjuntivo	No aserción e irrealidad: indica duda, deseo, situaciones hipotéticas o posibilidad. En estado descripto en la oración no se afirma como un estado conforme a la realidad.	Quizás esté en Mérida
Imperativo	Indica órdenes, prohibiciones o consejos.	Vuelva usted mañana

4.1.1 Conjugación del verbo

La conjugación de un verbo está constituida por las formas flexivas que adopta el verbo para expresar las distinciones asociadas a las categorías de persona, número, tiempo, modo, aspecto y voz. Las distintas terminaciones correspondientes a la forma del infinitivo (*am-ar, com-er y viv-ir*) han servido para clasificar a los verbos en tres grupos o conjugaciones (primera, segunda y tercera conjugación, respectivamente, como se puede ver en la tabla 17.

Para conjugar un verbo hay que conocer qué tipo de verbo es: si el verbo es regular; se aplican los pasos para conjugar un verbo regular. Si el verbo es irregular; en este caso la conjugación se hace en dos tiempos: determinar el patrón de irregularidad, y aplicar las reglas de irregularidad para ese patrón. Para verbos especiales como un auxiliar, verbos copulativos y verbos monosílabos, su conjugación es predefinida.

Conjugación regular: El proceso que sigue un verbo regular para la conjugación de las formas flexivas consiste en:

Paso 1: Extraer la raíz del verbo	*Paso 2*: Seleccionar las desinencias

Paso 1: Obtenemos la raíz del verbo mediante la eliminación de la terminación del infinitivo (-ar, -er o -ir), excepto para los tiempos de futuro simple y condicional simple de indicativo.

Paso 2: Una vez que hemos extraído la raíz del verbo y hemos identificado la conjugación a la que pertenece (-ar, -er, -ir), seleccionamos las desinencias correspondientes a cada forma verbal y las añadiremos a la raíz.

Tabla 19. Pasos generales para la formación de cualquier verbo regular

Paso 1: raíz
Paso 2: desinencias de modo y tiempo
Paso 3: desinencias de persona y número
Paso 4: tildes
= Verbo regular conjugado

4.2 PASOS PARA AVERIGUAR SI UN VERBO CUALQUIERA ES REGULAR O IRREGULAR

Paso 1: En primer lugar, es necesario comprobar si el verbo que entra pertenece a la lista de los siguientes 13 verbos particulares[53] {traer, valer, salir, tener, venir, poner, haber, decir, poder, querer, saber, andar, concluir}, o si se trata de un verbo auxiliar[54] {haber, entre otros }, copulativo[55] {*ser y estar*} o monosílabo[56] {*dar, ver, ir*}. Es necesario también tener en cuenta que cualquiera de estos verbos puede ir precedidos de algún prefijo, ya que los verbos prefijados mantienen las irregularidades propias de su base de prefijación. Si el verbo pertenece a algunos de estos grupos es irregular.

Paso 2: En segundo lugar, es necesario comprobar si el verbo termina en -quirir (adquirir), si se encuentra en esta lista (dormir, errar, morir, oler, erguir o desosar), o si se trata de algún prefijado de éstos, en cuyo caso presentará una regla de

[53] Son un grupo de verbos muy usados en español, tradicionalmente considerados como irregulares por sus diferencias con el resto de los verbos.

[54] Los verbos auxiliares son aquellos que sirven para formar los tiempos compuestos de la conjugación (verbo haber). Estos verbos contiene las desinencias verbales de estas construcciones y por lo tanto aportan la información gramatical.

[55] *Ser* y *estar* son los verbos copulativos del español. Son irregulares y su función es aportar la información gramatical a la oración mientras que casi no añaden información léxica.

[56] Son verbos cuyo infinitivo es una palabra monosílaba. En español, con la excepción de ser, que está considerado en el apartado de los copulativos, hay tres: dar, ver e ir.

irregularidad que hace que la vocal de la raíz diptongue (adquiero de adquirir, duermo de dormir, etc.).

Paso 3. En tercer lugar, es necesario confirmar si la raíz del verbo de la segunda o de la tercera conjugación termina vocal (leer, oír), ya que en ese caso también sería irregular y sufriría los cambios propios de los verbos cuya raíz termina en vocal (leyó de leer, oigo de oír). En el caso en el que el verbo pertenezca a la primera conjugación, tan solo sería irregular si su raíz termina en u o i (criar, actuar). Los verbos de la primera conjugación, con esa terminación en la raíz, se ven afectados por los cambios ortográficos de tilde (crío de criar, actúo de actuar).

Paso 4: En cuarto lugar, para saber si el verbo en cuestión se ve afectado por una regla ortográfica de cambio de letra, hay que comprobar que la raíz no termina en c, g, gu, qu, ñ o ll, en el caso de la segunda y de la tercera conjugación (corregir, seguir), o en c, z, g o gu, en el caso de la primera conjugación (sacar, trazar). Cuando la raíz del verbo termina en alguna de las consonantes indicadas, el verbo sufre un cambio ortográfico de letra (corrijo de corregir, sigo de seguir, sequé de secar, tracé de trazar). Estos verbos pueden acumular varios cambios de irregularidad a la vez, por lo que habrán de ser considerados también en el paso 5º y en el paso 6º.

Paso 5: En quinto lugar, hay que preguntar si la vocal de la raíz es *e* o *o* para los verbos de la primera conjugación (pensar, contar), o solamente *e* para los verbos de la tercera conjugación (servir, herir). En caso afirmativo, habrá que comprobar el resto de las condiciones de cambios de diptongos y alternancias vocálicas, para confirmar si el verbo puede sufrir alguna de estas irregularidades (pienso de pensar, cuento de contar, sirvo de servir, hiero de herir).

Paso 6: Finalmente, es necesario comprobar si el verbo presenta diptongo en la raíz de su infinitivo (reunir, europeizar). Si es así, se remite a las irregularidades de cambio ortográfico tilde, y sólo se añadirá tilde si se ajusta a las condiciones (reúno de reunir, etc.).

Cuando un verbo no cumple ninguno de los requisitos explicados en estos seis pasos (cantar, meter, aburrir), se puede decir que es regular, y, por tanto, se conjugará siguiendo los principios de la conjugación regular.

4.2.1 Conjugación irregular

Tal como se dijo antes, la conjugación de un verbo irregular se hace en dos tiempos, primero determinando el patrón de irregularidad y segundo aplicando las reglas de irregularidad para ese patrón.

Patrones de irregularidad: conjunto de formas verbales afectadas por una misma regla de irregularidad. Los patrones de irregularidad se corresponden con las formas verbales que cumplen una determinada condición formal o morfológica y, por ello, estas formas sufren una regla de irregularidad.

- Patrón To: si la silaba tónica cae en la raíz (*-ar, -er, -ir*)
- Patrón Te: cuando la silaba tónica cae en la raíz y la desinencia comienza por *e* (*-ir*)
- Patrón Dei: cuando la desinencia comienza por *e* o i (*-ar*)
- Patrón Dao: si la desinencia comienza por a u o (*-er, -ir*)
- Patrón Di: cuando la desinencia es tónica y comienza por *i* átona (*-er, -ir*)
- Patrón Dti: si la desinencia comienza por *i* tónica (*-er, -ir*).
- Patrón Dt-i: cuando la desinencia es tónica y empieza por vocal diferente de i (*-ir*)

Reglas de irregularidad. Una regla de irregularidad es un cambio que se realiza en la forma flexiva, hipotéticamente regular, de un verbo irregular, para dar como resultado la forma irregular del mismo. Este cambio se produce si el infinitivo del verbo satisface una o varias condiciones formales. Las reglas de la conjugación irregular se dividen en los siguientes grupos:

- Cambios ortográficos de letra: De acuerdo con la ortografía del español hay varias letras que se pronuncian de un modo distinto según la vocal que les siga. La letra c suena como z cuando la siguen las vocales e o i (cesto, circo), y como qu cuando va acompañada por a, o o u (casa, cosa, cubierto). La letras z y qu, seguidas de e o i, mantienen sus respectivas pronunciaciones. La g es pronunciada como j cuando va seguida de las vocales e o i (gesta, girasol), y como gu cuando va seguida de a, o o u (gacela, golfo, gusano). El fonema gu es la combinación de dos letras (g y u), y se utiliza para representar el sonido de la g de gacela, golfo y gusano cuando va seguida de e o i (guerra, guisante). Estas reglas ortográficas explican los cambios que experimentan en la escritura las formas verbales durante la conjugación de algunos verbos: según la vocal de la desinencia deberá sustituirse la consonante de la raíz para preservar la pronunciación del infinitivo.

- Cambios ortográficos de tilde: Cuando en una palabra entran dos vocales en contacto (siendo una de ellas a, e, o y otra i, u) hay dos posibilidades: que se pronuncien en una misma sílaba formando un diptongo (hueso), o que se pronuncien en sílabas diferentes formando un hiato (sinfonía). En un diptongo la vocal tónica siempre es a, e u o, nunca i o u (juego, abierto, guardia, descripción). Por este motivo, si se quiere deshacer un diptongo, basta con acentuar la i o la u (quería, actúo). En la conjugación verbal, este procedimiento ortográfico es muy utilizado cuando dos vocales entran en conflicto por la asignación del acento en los tiempos presentes de los diferentes modos.

- Diptongación y alternancias vocálicas en la raíz: La diptongación consiste en la alternancia en la raíz del verbo entre una vocal y un diptongo en determinadas personas y tiempos (cuento de contar). Esta alternancia está motivada por la acentuación de la vocal de la raíz: si esa sílaba es tónica habrá un diptongo, y si es átona se mantendrá la vocal del infinitivo (contaba de contar). Existe una relación establecida entre una vocal y el diptongo con el que puede alternar:

— las vocales e o i sólo pueden alternar con el diptongo ie;

— las vocales o o u sólo pueden alternar con el diptongo ue.

Las alternancias vocálicas en la raíz consisten en el cambio de una vocal a otra (sirvo de servir). No obstante, al igual que pasa con los diptongos, dicho cambio no es aleatorio, sino que se produce entre determinadas vocales:

— la vocal e sólo alterna con la vocal i y viceversa;
— la vocal o sólo alterna con la vocal u y viceversa.

- Verbos de la segunda y de la tercera conjugación cuya raíz termina en vocal: Por último, los verbos con raíz terminada en vocal, cuyas reglas de irregularidad tradicionalmente se han considerado únicas, mantienen un comportamiento sistemático y hacen uso de mecanismos para evitar la acumulación de vocales. Por ejemplo, todos los verbos de la segunda conjugación cuya raíz termina en vocal, como el verbo caer, sufren el siguiente cambio: añaden ig entre raíz y desinencia en las formas verbales afectadas (la primera persona de singular del presente de indicativo, la tercera persona de singular, la primera y tercera personas de plural del presente de subjuntivo, y todas las formas verbales del presente de imperativo).

Para el prototipo del aprendizaje morfosintáctico, en el componente de *aprendizaje duro* se implementaron sólo el patrón Dei y Dao. Los demás patrones no se implementaron porque requerían un trabajo dispendioso[57]. A continuación detallamos el comportamiento de esos dos patrones:

Patrón Dei : desinencia que empieza por las vocales (e, i). Este patrón actúa sobre la regla de irregularidad de tipo ortográfico (cambio ortográfico de letra). Se trata de ajustes ortográficos propios del español cuando, para mantener una pronunciación, ha de cambiarse la consonante según la vocal que la sigue

[57] Convertir estos patrones a formulas computacionales.

(escenifique de escenificar). El patrón Dei sólo afecta verbos de la primera conjugación. Este patrón aparece en:

— la primera conjugación (-ar).

— la regla de irregularidad de cambios ortográficos de letra (secar)

Dao[58]: desinencia que empieza por las vocales (a, o). Su funcionamiento es igual al del patrón anterior Dei, pero afecta a la segunda y a la tercera conjugación (por ejemplo, corrijo de corregir). Este patrón se encuentra en:

— la segunda y la tercera conjugación (-er e -ir).
— las reglas de irregularidad de cambios ortográficos de letra (secar), verbos cuya raíz termina en vocal (caer y sonreír)

4.2.2 Condiciones que debe cumplir un verbo para ser irregular

Tabla 20. Condiciones de irregularidad de un verbo

Si ¿es un verbo de los 13 verbos particulares anterior mente definidos, auxiliar, copulativo, o monosílabo? , ¿está formado por un prefijo + un verbo de los 13 verbos particulares anteriormente definidos / auxiliar / copulativo /monosílabo?	entonces	Regla Ir a verbos particulares ir a verbo auxiliar ir a verbos copulativos ir a verbos monosílabos
Si ¿termina en -quirir?, ¿es dormir, errar, morir, oler, erguir, desosar o es un verbo prefijado formado a partir de alguno de estos verbos?	entonces	Regla Ir a reglas de irregularidad: diptongación y alternancias vocálicas en la raíz
Si ¿su la raíz termina en vocal? (excepto: traer y terminados en -quir o -guir) y ¿pertenece a la segunda conjugación o a la tercera conjugación?	entonces	regla Ir a reglas de irregularidad: verbos cuya raíz termina en vocal
¿si la raíz termina en vocal? (excepto: traer y	entonces	regla

[58] Desinencia en a, o

terminados en -quir o -guir) y ¿pertenece a la primera conjugación y termina en -iar o -uar?		Ir a Reglas de irregularidad: Cambios ortográficos de tilde
Si ¿su raíz termina en c, g, gu, qu, ñ o ll y pertenece a la 2ª o a la 3ª conjugación? c, z, g o gu y pertenece a la 1ª conjugación?	entonces	regla
		Ir a Reglas de irregularidad: Cambios ortográficos de letra
Si ¿es la vocal de la raíz del verbo una *e* o *o* y pertenece a la primera conjugación? *e* y pertenece a la tercera conjugación?	entonces	regla
		Ir a Reglas de irregularidad: diptongación y alternancias vocálicas en la raíz
Si: ¿Tiene un diptongo en su raíz?	entonces	regla
		Ir a Reglas de irregularidad: Cambios ortográficos de tilde

Con estos patrones y reglas, junto con las condiciones definidas en el apartado anterior para identificar si un verbo es irregular, se puede averiguar el comportamiento de cualquier verbo español conociendo exclusivamente su infinitivo, y, en algunos casos, teniendo en cuenta las palabras con las que están emparentados.

Asimismo, según las reglas de irregularidad y los diferentes patrones, se pueden clasificar los verbos pertenecientes a la conjugación irregular en los siguientes grupos: verbo auxiliar (haber), verbos copulativos (ser y estar) y verbos monosílabos (dar, ver e ir). Estos verbos son los únicos cuya conjugación es tan irregular que no se pueden adscribir a ningún tipo de regla o patrón de irregularidad. Otros verbos de interés son:

Verbos de propósito general: conjunto de verbos muy frecuentes en español (hacer, tener, decir, etc.) que poseen unas irregularidades específicas (aunque sistematizadas), las cuales, en algunos casos, coinciden con las reglas y patrones de irregularidad de los verbos regularmente irregulares.

Verbos regularmente irregulares: estos verbos sufren reglas de irregularidad según cumplan una o varias condiciones.

A los verbos de propósito general y a los regularmente irregulares les afectan las reglas de irregularidad.

En general, el prototipo de prueba implementado en este trabajo soporta la conjugación de la gran mayoría de los verbos regulares, y para el caso de los irregulares, se han implementado los casos de Cambios ortográficos de letra, y diptongación y alternancias vocálicas en la raíz. Así, se desarrollo una aplicación que implementa un conjugador de verbos en español, o flexionador de formas verbales, para él aprendizaje morfosintáctico, el cual toma el verbo en infinitivo y entrega la forma conjugada solicitada, esto es, la forma verbal para el modo, tiempo y persona de los casos contemplados (ver capitulo 5). En esta aplicación no se contemplaron los verbos de propósito general (traer, valer, salir, tener, venir, poner, hacer, decir, poder, querer, saber, andar, conducir), ni el auxiliar (haber), ni los copulativos (ser, estar), ni los monosílabos (dar, ver, ir). Estos verbos, por sus particularidades (de irregularidad y número reducido), es más práctico cargarlos manualmente en el lexicón que diseñar algoritmos para su reconocimiento y conjugación.

5. DISEÑO DEL PROTOTIPO DEL APRENDIZAJE AUTOMÁTICO DE ONTOLOGÍAS

En este capítulo se elabora el análisis y el diseño del componente de aprendizaje de ontologías, y se procede a la implementación de la solución para el caso de prueba particular descrito en los capítulos 3 y 4, que tiene que ver con el aprendizaje morfosintáctico.

5.1 DIAGRAMA DE CASOS DE USO DEL COMPONENTE DE APRENDIZAJE DE ONTOLOGÍAS

La descripción de los casos de uso se presenta a continuación.

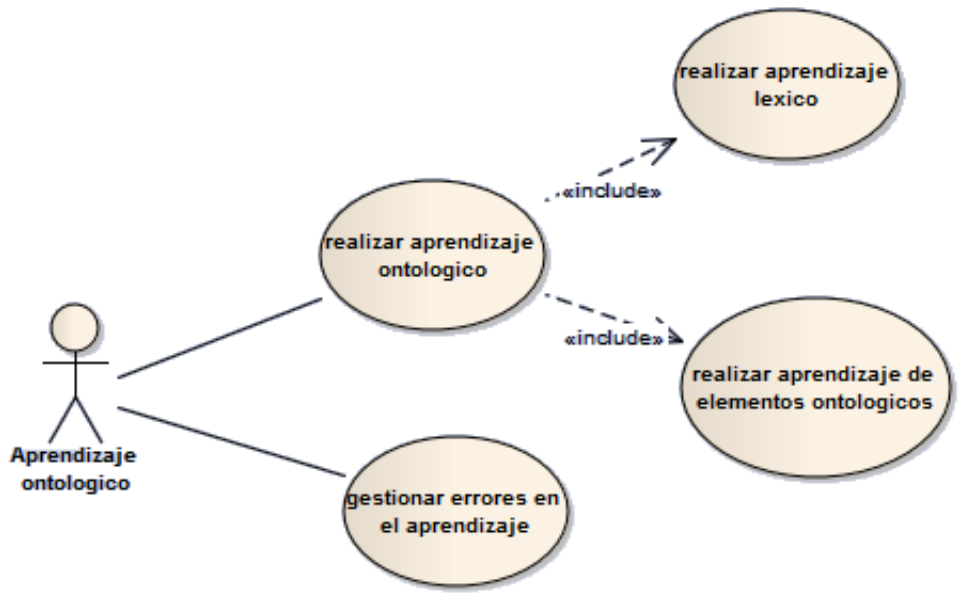

Figura 10. Diagrama de casos de uso del componente de aprendizaje de ontologías

Tabla 21. Descripción de caso de uso realizar aprendizaje ontológico

Nombre:	Realizar aprendizaje Ontológico
Actor:	Aprendizaje ontológico
Propósito : Permite invocar el sistema de aprendizaje para MODS	
Precondiciones: 1. Existe una entrada para el aprendizaje.	
Flujo Normal: 1. Se toma la fuente de aprendizaje 2. se procesa la fuente de entrada 3. se caracteriza la fuente procesada 3.1 se crea un archivo XML lexmor para mecanismos léxicos 3.2 se crea un archivo XML sem para mecanismos semánticos	
Poscondición: 1. la información caracterizada es enviada a un mecanismos de aprendizaje particular	

Tabla 22. Descripción de caso de uso realizar aprendizaje léxico

Nombre:	Realizar aprendizaje léxico
Actor:	Aprendizaje ontológico
Propósito : Permite aprender conocimiento léxico para enriquecer el lexicón del MODS	
Precondiciones: 1. Se ha invocado el mecanismos de aprendizaje morfosintáctico 2. Existe el archivo XML lexmor para el aprendizaje.	
Flujo Normal: 1. Se obtiene la información del XML lexmor 2. Se aplican los algoritmos de aprendizaje léxico 3. Se caracteriza la información aprendida 3.1 Se crea un archivo XML aprendizaje lexmor	
Poscondición: 1. El archivo XML aprendizaje lexmor (con la información aprendida) es enviado al componente de actualización	

Tabla 23. Descripción de caso de uso realizar aprendizaje semántico

Nombre:	Realizar aprendizaje semántico
Actor:	Aprendizaje ontológico
Propósito : Permite aprender conocimiento semántico para enriquecer la ontología interpretativo y o lingüística del MODS.	
Precondiciones: 1. Se ha invocado el mecanismos de aprendizaje semántico 2. Existe el archivo XML sem para el aprendizaje.	
Flujo Normal: 4. Se obtiene la información del XML sem 5. Se aplican los algoritmos de aprendizaje semántico 6. Se caracteriza la información aprendida 3.1 Se crea un archivo XML aprendizaje sem	
Poscondición: 1. El archivo XML aprendizaje sem (con la información aprendida) es enviado al componente de actualización	

Tabla 24. Descripción de caso de uso realizar gestión de errores

Nombre:	Gestionar errores de aprendizaje
Actor:	Aprendizaje ontológico
Propósito: Gestionar los errores que puedan ocurrir en el intento de aprender información léxica, y reportar estos al MODS.	
Precondiciones: 1. Se ha invocado el aprendizaje ontológico 2. Se ha caracterizado la fuente para el aprendizaje léxico 2.1 se ha creado el archivo XML lexmor	
Flujo Normal: 1. Se recibe el archivo XML lexmor 2. Se chequea la información de interés	
Poscondición 2. Enviar un mensaje con el reporte al MODS	

5.2 DIAGRAMA DE ACTIVIDADES DEL SISTEMA DE APRENDIZAJE DE ONTOLOGÍAS

Las figuras siguientes describen los diagramas de actividades para los casos de uso anteriores.

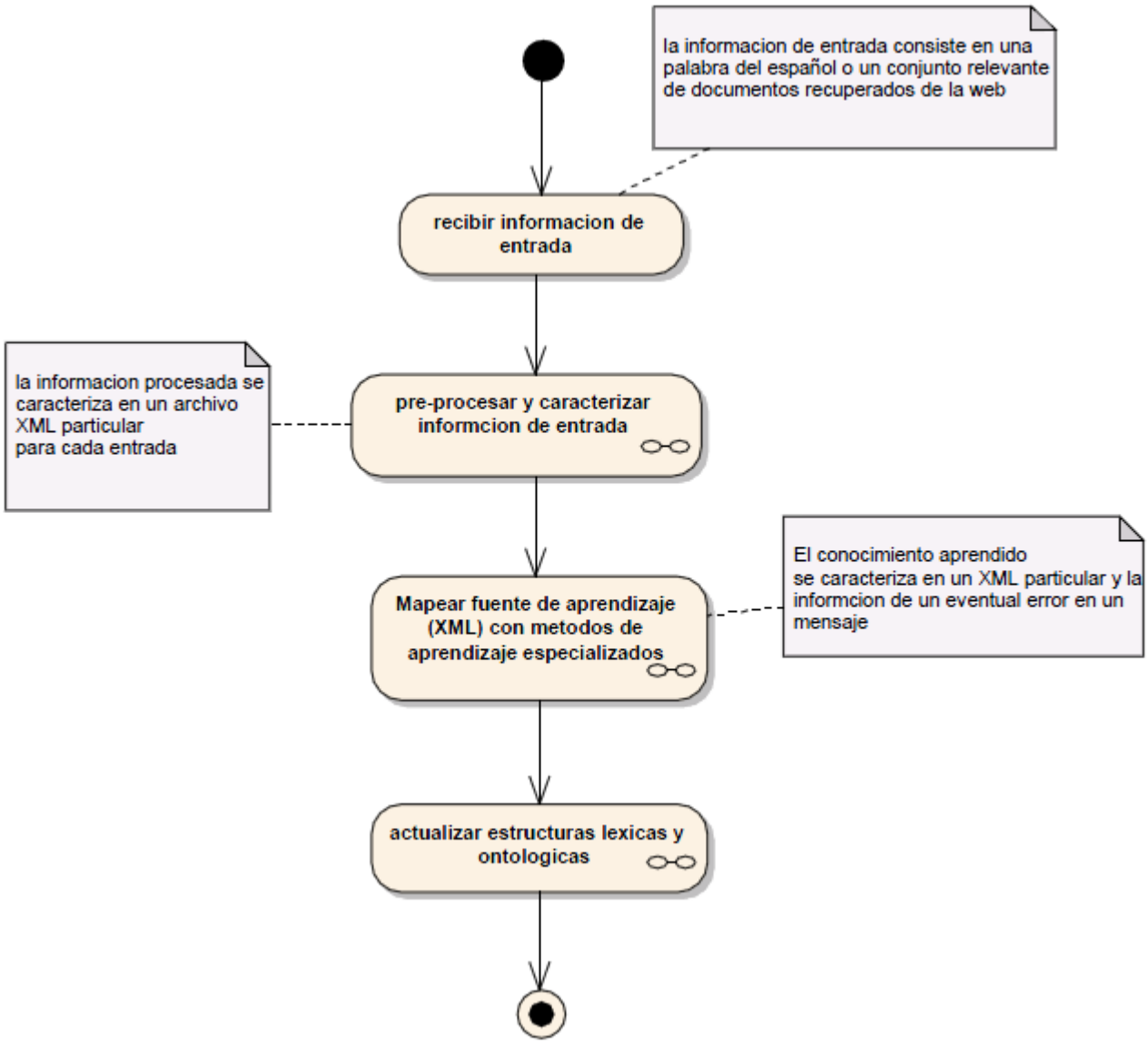

Figura 11. Diagrama de actividades: aprendizaje de ontología

5.2.1 Diagrama de actividades: pre-procesar información

En la figura 12 se muestras el conjunto de actividades que se llevan a cabo dentro del proceso general de pre-procesamiento. Existe un pre-procesamiento diferente para cada entrada.

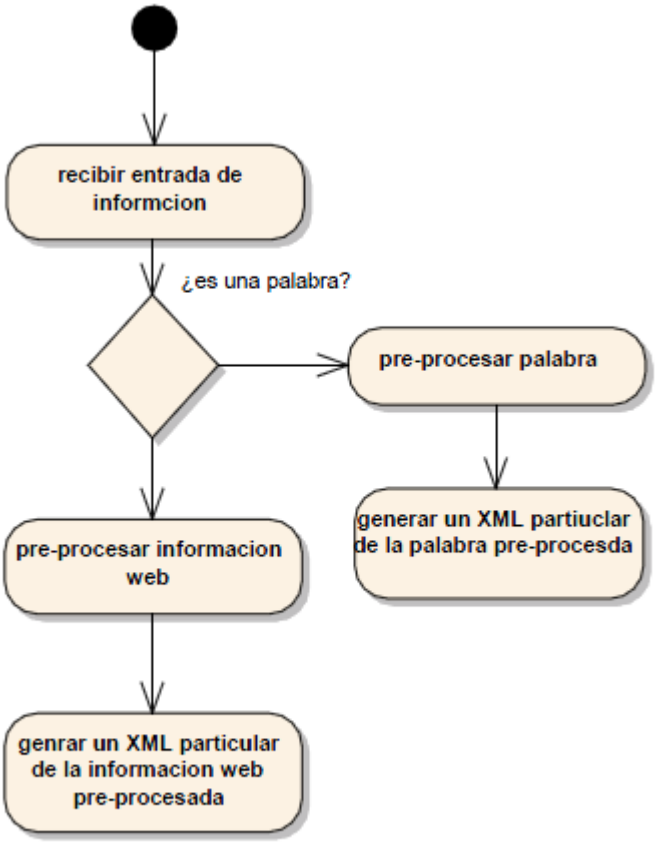

Figura 12. Diagrama de sub-actividades: pre-procesar informacion

5.2.2 Diagrama de actividades: gestionar aprendizaje

En la figura 13 se muestra el conjunto de actividades y decisiones que se llevan a cabo en el proceso general de coordinación del aprendizaje. Se tiene un conjunto particular de métodos de aprendizaje por cada tipo de entrada. Estos métodos se caracterizan por extraer conocimiento a partir de una fuente particular.

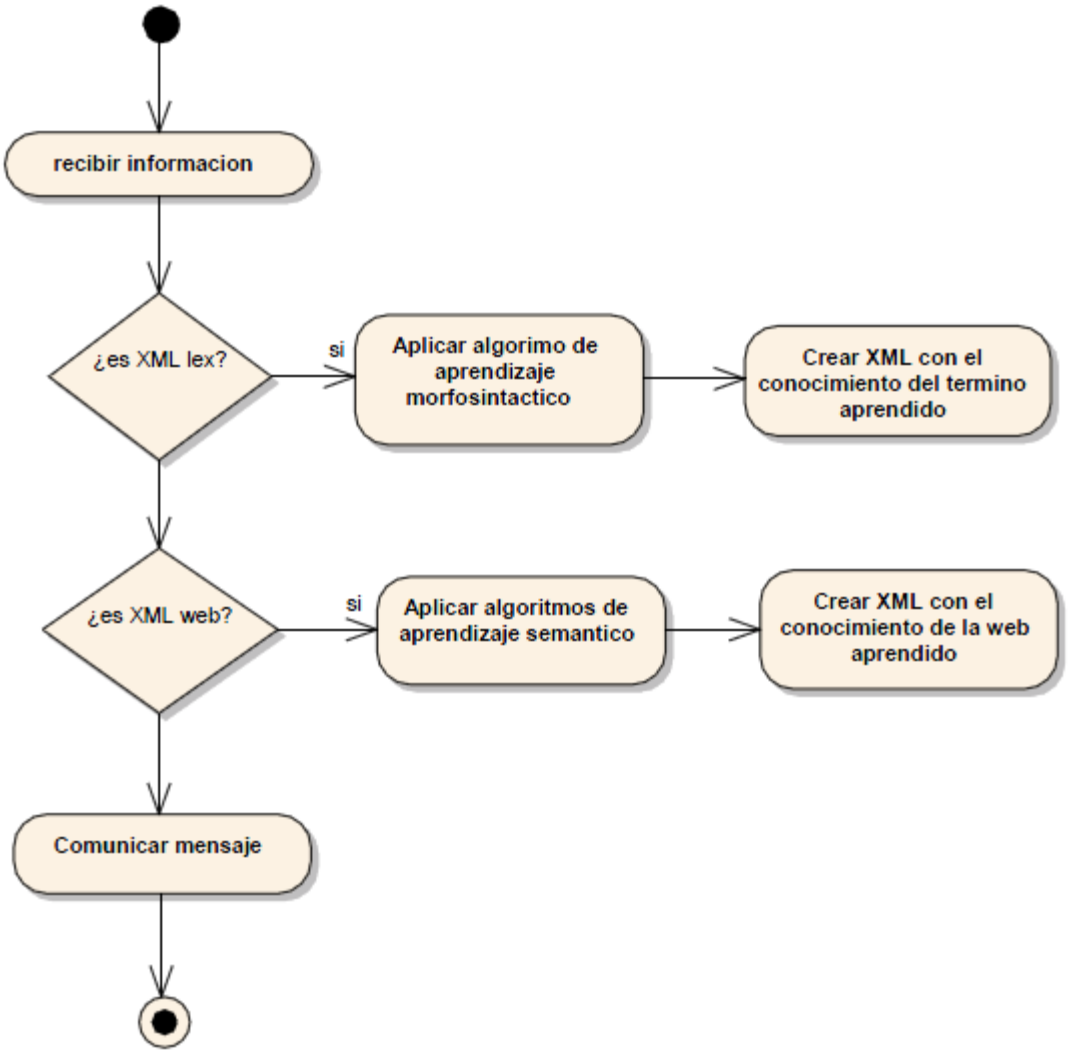

Figura 13. Diagrama de sub-actividades: gestionar aprendizaje

5.2.3 Diagrama de actividades: actualizar aprendizaje

La información aprendida va a una estructura particular dentro de las estructuras del MODS. La información léxica va al lexicón, y la información semántica a las estructuras ontológicas.

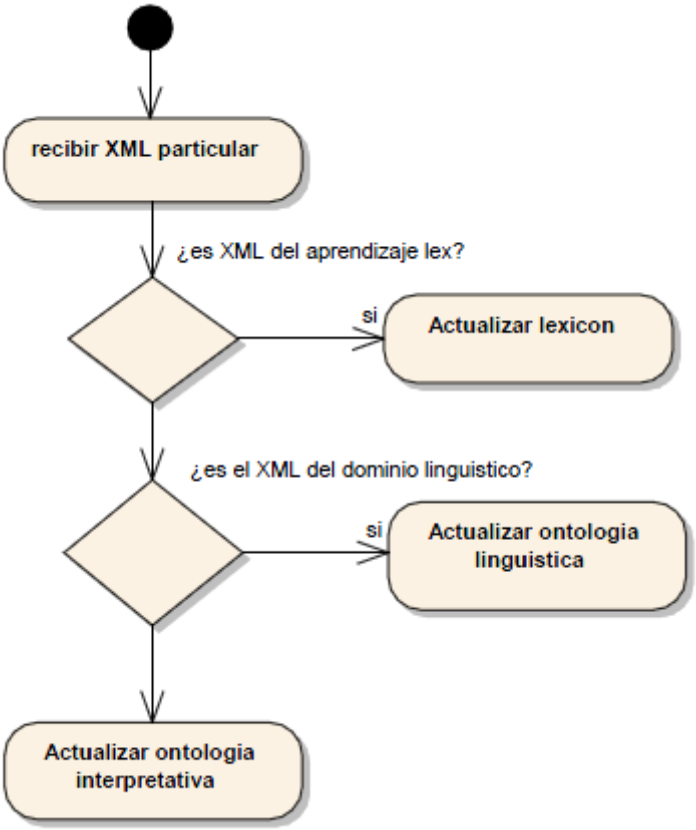

Figura 14. Diagrama de sub-actividades: actualizar aprendizaje

5.3. DISEÑO DE CLASES DEL COMPONENTE DE APRENDIZAJE DE ONTOLOGÍAS

Una clase está compuesta por atributos y métodos. Los métodos dan la funcionalidad de la clase, y tienen la misma estructura de una función en un lenguaje estructurado, con excepción de los modificadores de acceso. La figura 15

muestra el diagrama de clase que describe el componente de aprendizaje de ontologías.

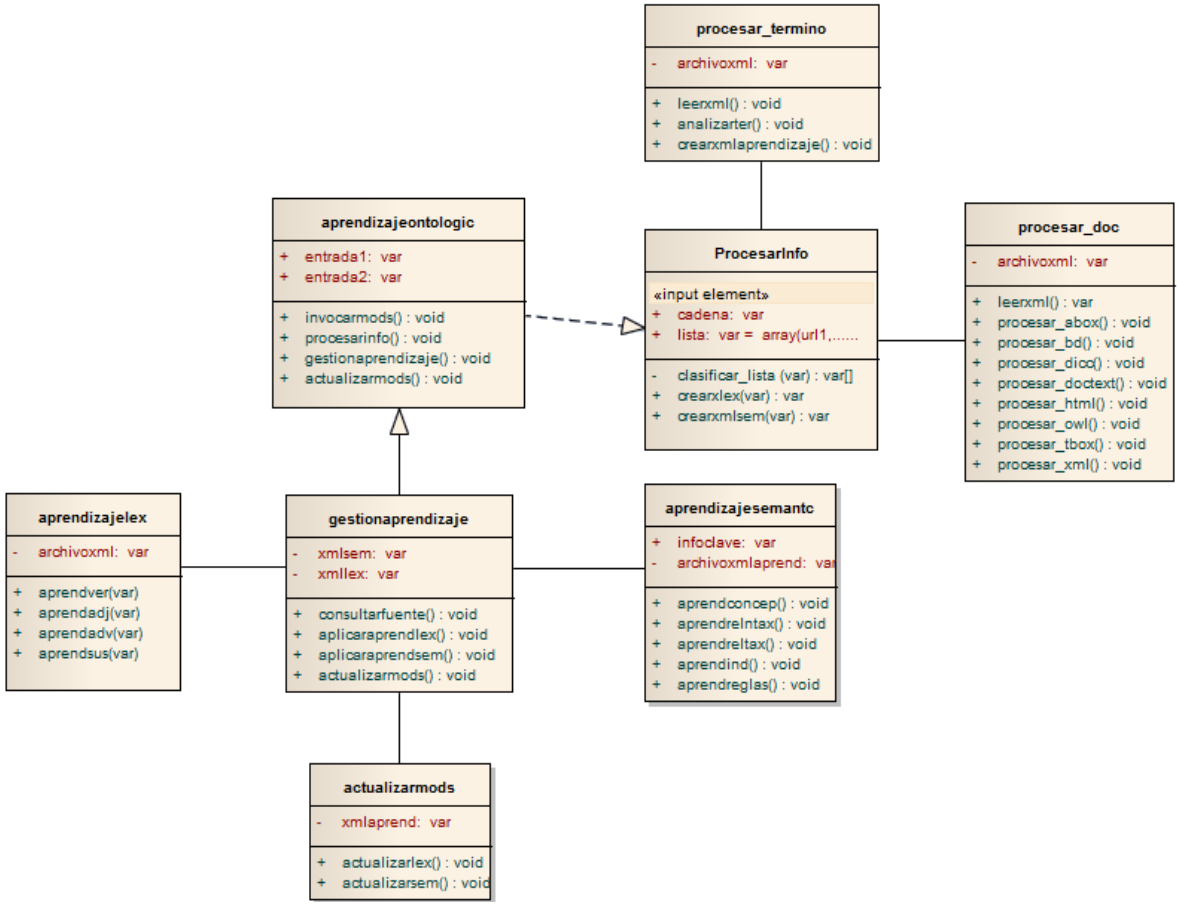

Figura 15. Diagrama de clases: componente de aprendizaje de ontologías

La clase principal es *aprendizajeontologico,* cuyas operaciones son: *invocarmods()* (cuya tarea es acceder a las estructuras del MODS para el proceso de actualización), le sigue *procesarinfo() (*el cual genera un archivo xml con la información de entrada), el método gestionaprendizaje() (recibe un archivo xml y aplica el método para esa fuente), y por último, la operación actualizar (recibe un xml con la nueva información y la incorpora en el MODS). Las otras clases, tales como *aprendizajelex, aprendizajesem, actualizarmods,* guardan los métodos necesarios para el aprendizaje y actualización, y son invocadas por

gestionaraprendizaje. Por otra parte, *procesarinfo* proporciona la entrada a *gestionaraprendizaje*.

5.4 DIAGRAMAS DE COMPONENTES DEL COMPONENTE DE APRENDIZAJE DE ONTOLOGÍAS

A través de este diagrama se muestra los diferentes componentes que constituyen al sistema de aprendizaje (ver figura 16).

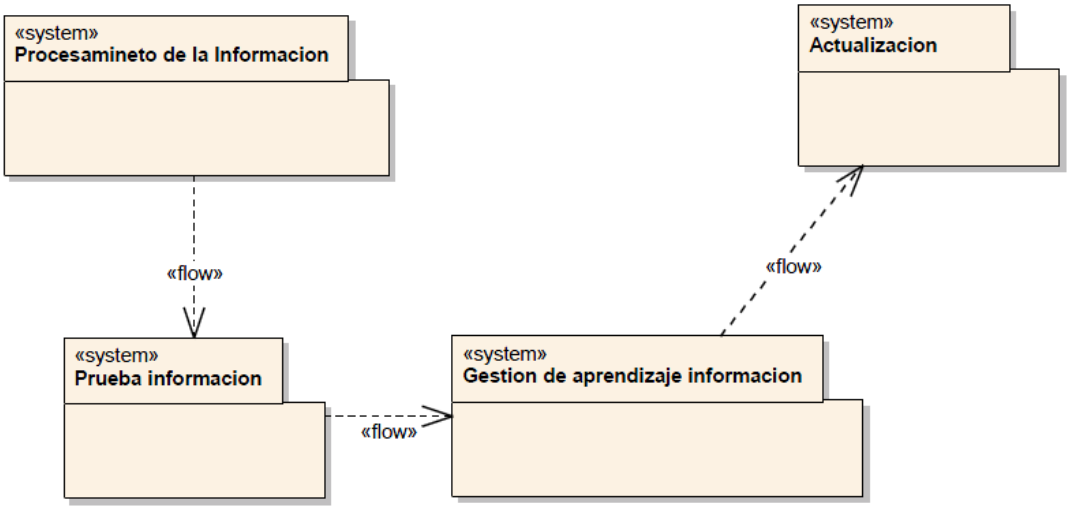

Figura 16. Diagrama de Componentes del sistema de aprendizaje.

El sistema de aprendizaje lo constituyen cuatro paquetes, cada uno de los cuales trabaja con otros sub-sistemas. A continuación detallamos cada uno:

5.4.1 Sistema de Gestión de aprendizaje de ontologías

En la figura 17 se muestran detalles de ese diagrama de componente. Los componentes más importantes son descritos a continuación: el componente *Técnicas de aprendizaje léxico* es responsable del aprendizaje de la información léxica, tal como aprender las categorías gramaticales de una palabra. Para este componente se desarrollo el módulo especial para la conjugación de verbos. Su diseño y construcción será expuesto más adelante, en la sección 5.5. El paquete

de *métodos de aprendizaje semántico* es el que contiene los métodos y técnicas para el descubrimiento de conceptos, relaciones taxonómicas, relaciones no-taxonómicas e individuos[59]. Para este componente se diseño un método para el aprendizaje de relaciones no-taxonómicas.

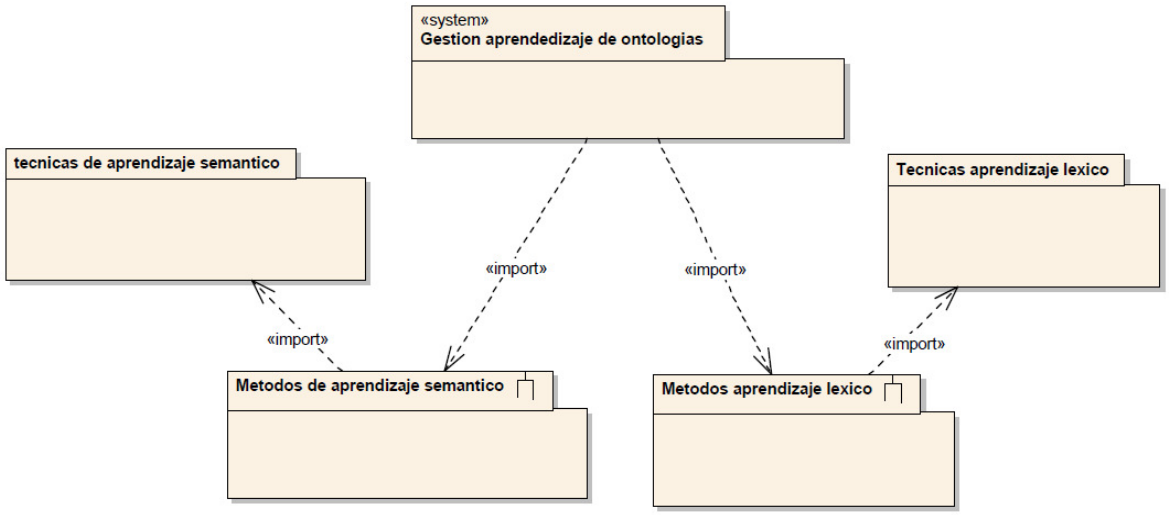

Figura 17. Diagrama de paquetes del sistema de Gestión de aprendizaje

5.4.2 Sistema de prueba de información léxica

Es importante destacar que este sistema es propio para la *entrada1*, es decir para cuando existe un petición del MODS sobre un requerimiento sobre una palabra desconocida (ver figura 18). Los componentes más importantes son descritos a continuación: el componente de *búsqueda en diccionarios online* revisa, a través de métodos de búsqueda ligera[60], la existencia de la palabra y su caracterización en diccionarios. El componente de *análisis del corpus* lleva a cabo procesos más formales, para chequear el uso dado a la palabra desconocida. Este componente puede usar técnicas de análisis lingüístico[61], apoyado en técnicas estadísticas. El componente de *búsqueda en toda la web* es un componente mucho más informal,

[59] Un individuo hace parte de un concepto que describe. Es como la instancia de la clase en el paradigma de objetos.
[60] Analizar las estructuras de un documento para extraer información de interés, tal como un párrafo de texto en una página web sin entrar en profundidad.
[61] Técnias de agrupamiento, técnicas de análisis estadístico, etc.

donde se lanza a través de un buscador la palabra desconocida, con el fin de recuperar alguna información extra que no será analiza, sino se tomara como último recurso para producir alguna respuesta.

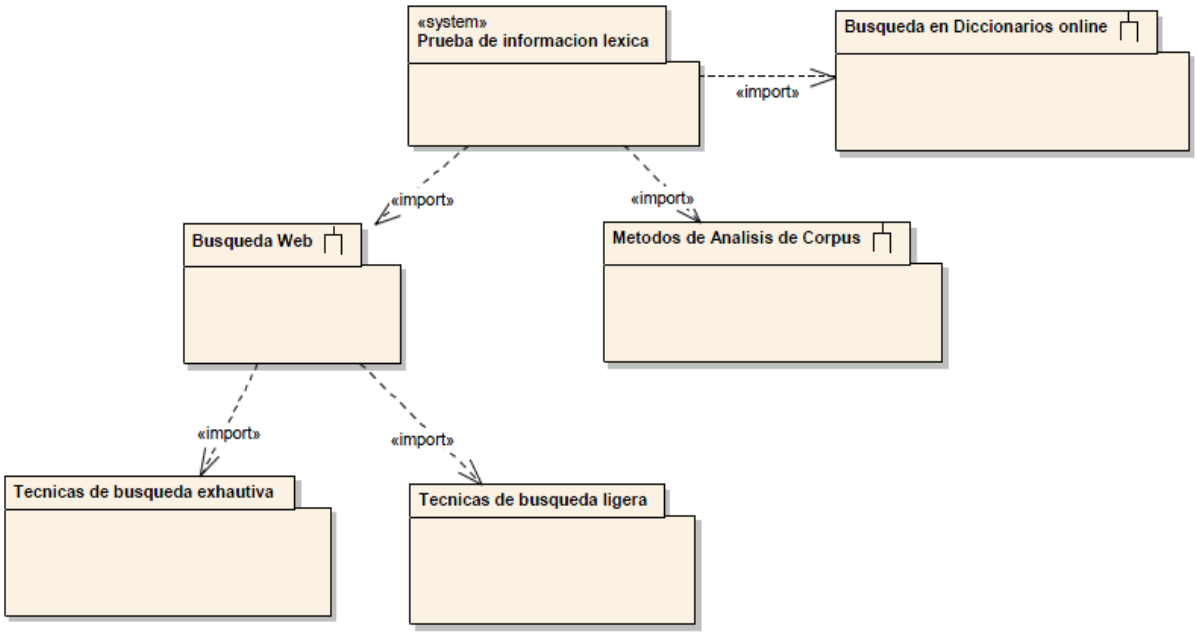

Figura 18. Diagrama de paquetes del sistema prueba de información

5.4.3 Sistema de procesamiento de la información

En la figura 19 se muestran detalles de ese diagrama de componente. Los componentes más importantes son descritos a continuación: el componente de *métodos de procesamiento de términos* tiene por objetivo analizar la estructura interna del término para extraer alguna información morfológica importante, por ejemplo la forma canónica de una palabra, su ortografía, si es compuesta; sus constituyentes, etc. Los *métodos de procesamiento web* contienen algunos programas especializados para el tratamiento de recursos web, cuya salida es material importante para el aprendizaje. Por ejemplo, extraer de una página web el texto plano.

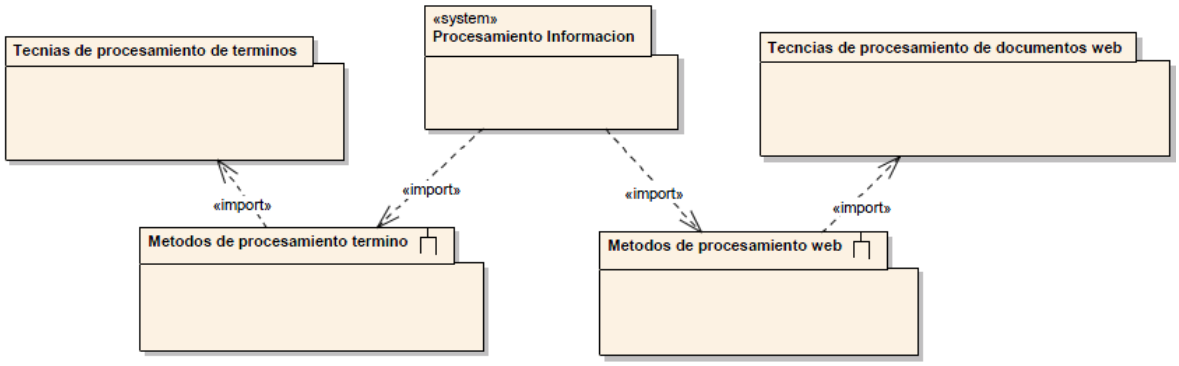

Figura 19. Diagrama de paquetes para el sistema procesamiento de información

5.4.4 Sistema de actualización del MODS

En la figura 20 se muestran detalles de ese diagrama de componente. Los componentes más importantes son descritos a continuación: Los *métodos de actualización léxica* cumplen la tarea de incorporar la nueva información léxica en el lexicón, revisar su existencia, completar algunas propiedades gramaticales, incorporar todo un nuevo registro léxico, etc. Los métodos de *actualización semántica*, aunque sus tareas se asemejan a las de actualización léxica, son mucho más complejas para algunos elementos. Por ejemplo, cómo incorporar taxonómicamente un concepto, de que conceptos es instancia un individuo, etc.

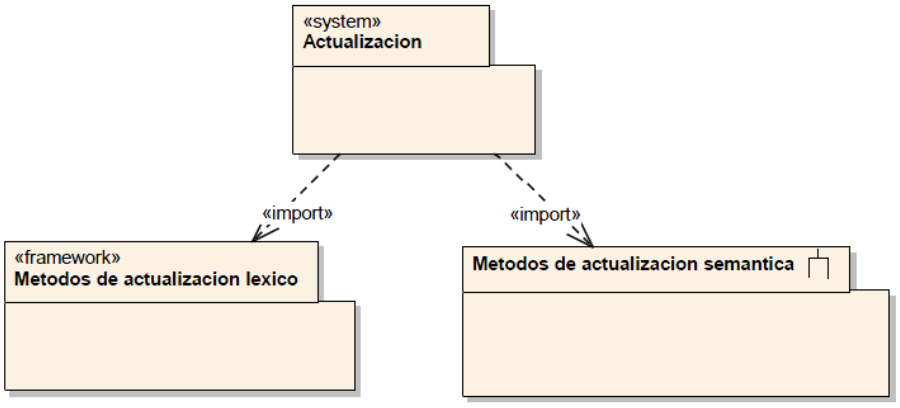

Figura 20. Diagrama de paquetes para el sistema de actualización

5.5 DISEÑO ESPECÍFICO DEL PROTOTIPO PARA EL APRENDIZAJE MORFOSINTÁCTICO: CASO CONJUGACIÓN DEL VERBO

En esta sección se detalla el diseño del prototipo del sistema de aprendizaje morfosintáctico. El prototipo se caracteriza por soportar el aprendizaje de nueva información morfosintáctica del verbo, sustantivo, etc. En el caso del verbo, es la categoría léxica que más dificultad o complejidad presenta a la hora de su aprendizaje. En parte, por la gran variedad de fenómenos lingüísticos que presenta. Por consiguiente, para el aprendizaje morfosintáctico se hará hincapié en mostrar el diseño detallado para el caso del verbo, basado en lo que se presento en los capítulos 3 y 4 (en especifico, el caso de conjugación del verbo).

5.5.1 Diagrama de caso de uso: Conjugar verbo

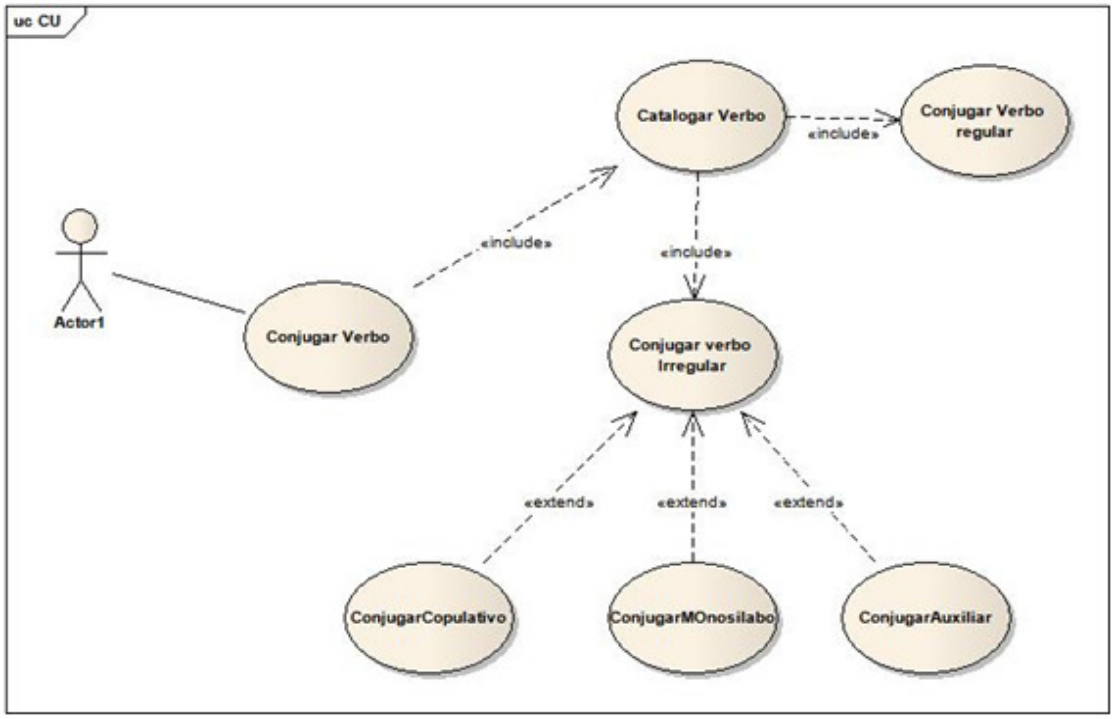

Figura 21. Diagrama de casos de uso para el modulo conjugar verbo

Descripción del caso de uso.

Nombre:	Conjugar Verbo
Actor:	Aprendizaje Morfosintáctico
Propósito : Hallar las formas flexivas para verbos regulares e irregulares	
Precondiciones: 3. Que el verbo sea valido 4. Que el verbo se encuentre en su forma canónica (infinitivo)	
Flujo Normal: 3. Se recibe el archivo XML lexmor donde viene la palabra verbo 4. El catalogador determina qué tipo de verbo es: si es regular o irregular 4.1 si es regular aplica reglas de conjugación regular 4.2 si es irregular 4.2.1 determina el patrón de irregularidad 4.2.2 aplicar reglas de acuerdo al patrón	
Poscondición 1. archivo XML aprendizaje lexmor con la información de la forma flexiva del verbo.	

5.5.2 Diagrama de actividades: conjugar verbo

El diagrama de actividades de la figura 22 muestra las tareas y el flujo lógico para el módulo de conjugación de verbo.

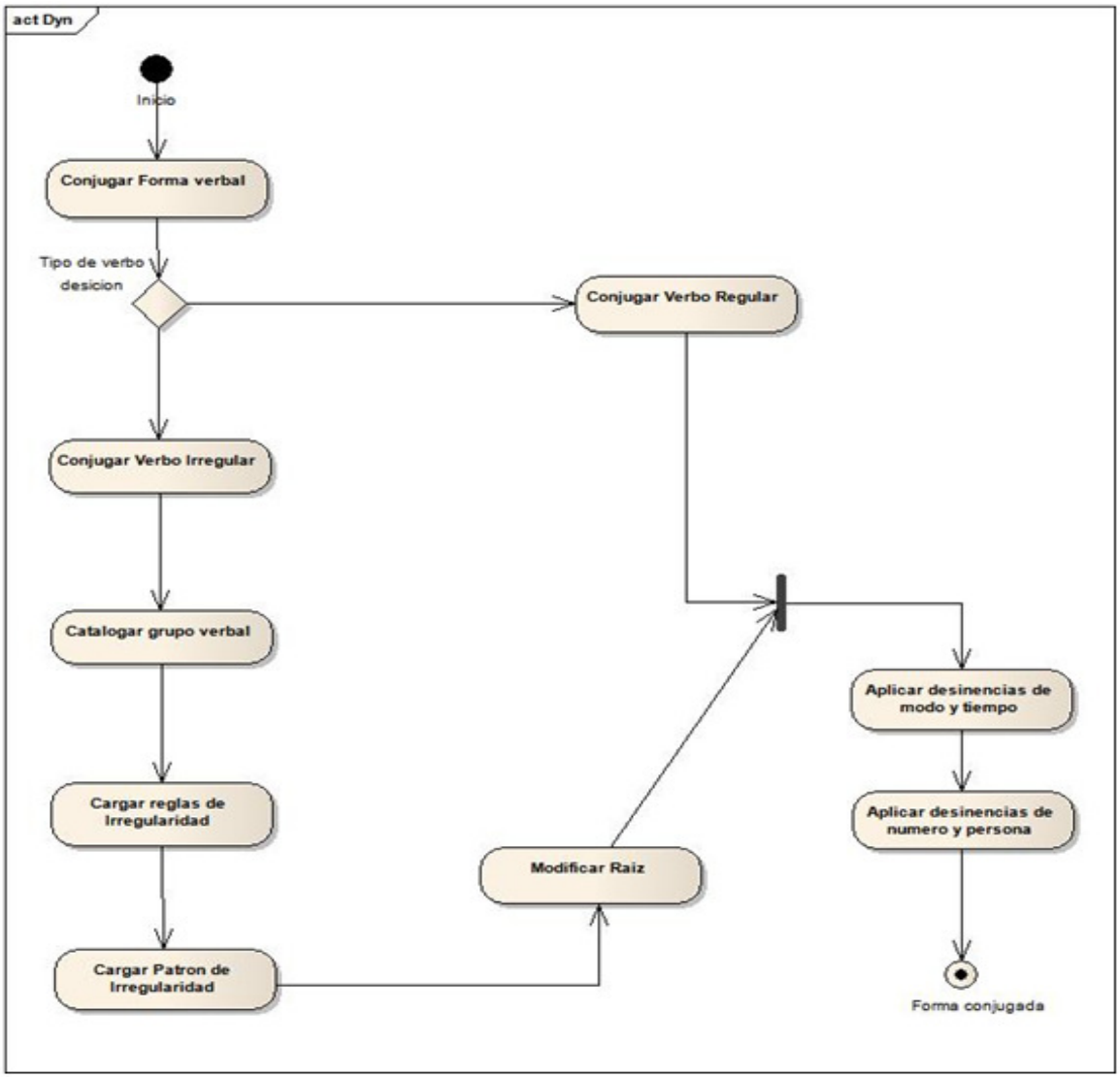

Figura 22. Diagrama de actividades para el módulo de conjugación de verbo

5.5.3 Diagrama de clase para el módulo de conjugación de verbo

flexverb es la clase principal que contiene los métodos necesarios para conjugar un verbo, sea regular o irregular. La clase *catalogador* es la que, con base a unos

métodos creados a partir de unos patrones, clasifica los verbos en regulares e irregulares; y la clase *flexvirregular* es una especialización para los verbos que se ha identificado como irregulares (contiene detalles extras, para los métodos específicos para conjugar verbos irregulares. Más adelante se habla de algunos de esos detalles).

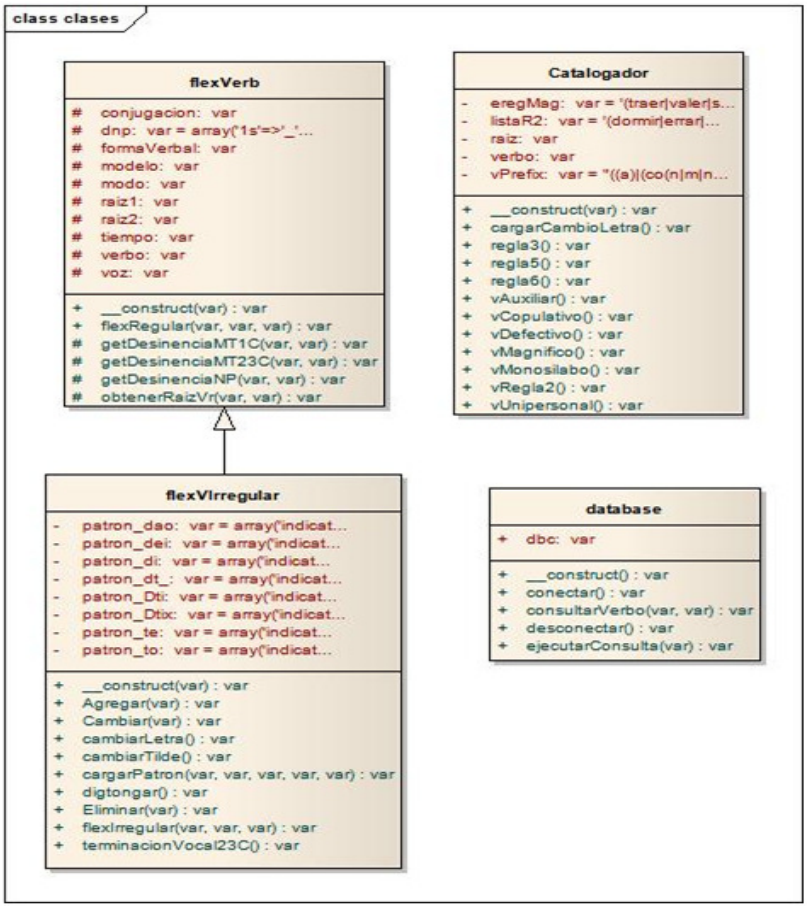

Figura 23. Diagrama de clase para el modulo de conjugación de verbo

5.5.4 Conjugación de irregular con patrón Dao y Dei

En esta última sección se muestra una pequeña parte de la programación que se desarrollo para los verbos irregulares, cuyas estructuras mapean con el patrón Dao y Dei. Estos patrones se explican en el capítulo 4, donde además se dice por

qué solo esos dos patrones se implementaron para el componente de aprendizaje duro. Su implementación (el código se encuentra en el lenguaje php) se muestra en los siguientes cuadros:

El primer cuadro muestra los modos, tiempos y personas afectados por el patrón Dei.

```
$patron_dei =  array(
'indicativo'=>array('PtoPerSim'=> array('1s')),
'subjuntivo'=>array('Pte'=>array('1s','2s','3s','1p','2p','3p')),
'imperativo'=>array('Pte'=> array('3s','1p','3p')));
```

El siguiente cuadro muestra los modos, tiempos y personas afectados por el patrón dao (implementados).

```
$patron_dao =  array(
'indicativo'=>array('Pte'=>array('1s')),
'subjuntivo'=>array('Pte'=>array('1s','2s','3s','1p','2p','3p' )),
'imperativo'=>array('Pte'=> array('3s','1p','3p' )));
```

El resto de código de los programas desarrollados se encuentra en el Anexo A. En él se puede ver la implementación de cada una de las operaciones indicadas en el diagrama de clases de la figura 23, como parte del aprendizaje morfosintáctico de verbos. Además, se encuentra el resto del código para el aprendizaje simiente, ágil, y duro.

6. CASOS DE ETUDIO DEL APRENDIZAJE MORFOSINTÁCTICO

En este capítulo se muestran varios ejemplos de aprendizaje léxico soportados por el sistema de aprendizaje morfosintáctico. Para todos los ejemplos se parte de la misma estructura de entrada definida por el MODS, a saber:

lex_mor[62](*componente léxico, categoría, tipo, genero, número, modo, tiempo, aspecto, voz, persona, instancia_ontologia_linguistica*).

A través de esta estructura el MODS envía al componente de aprendizaje sus requerimientos de información léxica, y recibe en el mismo formato lo aprendido de ella.

6.1 APRENDIZAJE A PARTIR DE UN SUSTANTIVO

Entrada1 = lex_mor(***María, Desconocido****, null, null, null, null, null, null, null, null)*.

Esta entrada1 pasa primero por el *aprendizaje simiente* (pre-procesamiento y caracterización), el cual da como resultado un archivo XML con la información estructurada siguiente:

XML entrada1
<XML> *<entrada1>* *<metodo id="aprendizaje_lexmor"/>* *<termino>**María**</termino>* *<categoria> **Desconocido** </categoria>* *<pre_procesado>* *<forma_canonica>**María**</forma_canonica>* *<prefijo>null</prefijo>*

[62] Lex_mor es la estructura de entrada proveniente del MODS

```
   <sufijo> null</sufijo>
 </pre_procesado >
</entrada1>
</XML>
```

Este esquema contempla palabras compuestas, por eso las etiquetas <forma_canonica></forma_canonica>, <prefijo></prefijo>, <sufijo></sufijo>. Una vez se han creado con satisfacción el archivo XML, se pasa al aprendizaje Ágil, que utiliza la información etiquetada de este archivo como fuente de aprendizaje, y produce la salida siguiente:

XML aprendizaje

```
<XML>
   <aprendizaje_lexmor>
        <categoria> sustantivo</categoria>
        <tipo>nombre propio</tipo>
        <genero> femenino </genero>
         <numero> singular </numero>
        <modo> null </modo>
        <tiempo> null </tiempo>
        <aspecto> null </aspecto>
        <voz> null </voz>
        <persona> null </persona>
    </aprendizaje_lexmor>
</XML>
```

Parte de esta información es extraída de diccionarios en línea (ver figura 24)

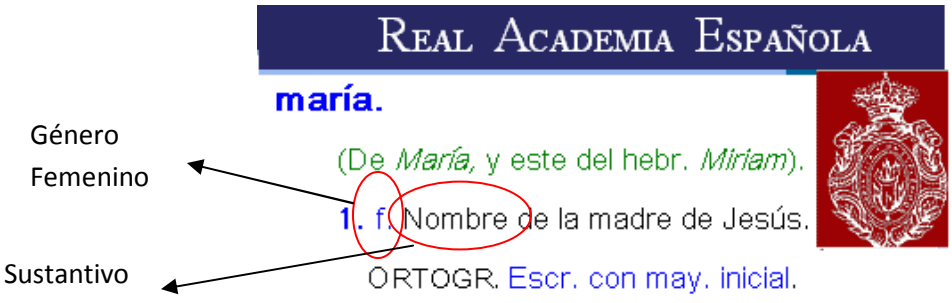

Figura 24. Respuesta de un diccionario

Y otra es deducida a través del análisis de su ortografía (para el caso de nombres propios, que por lo general inicia con mayúscula), o comprobando el morfema de género –a (o en otros casos, comprobando si la palabra requiere un artículo femenino), si presentan desinencias de plural (tal como “s”), etc.

Un sustantivo no pasa por el aprendizaje duro, con lo cual termina y devuelve al MODS la nueva estructura:

Salida (entrada1) = lex_mor(***María, sustantivo, nombre propio, femenino, singular****, null, null, null, null, null, null)*.

6.2 APRENDIZAJE A PARTIR DE UN VERBO

Para la entrada

Entrada1 = lex_mor(***Amo, Desconocido****, null, null, null, null, null, null, null, null)*.

El aprendizaje simiente genera:

XML entrada1 (fuente)
<XML> *<entrada1>* *<metodo id=”aprendizaje_lexmor”/>* *<termino>**Amo**</termino>* *<categoria> **Desconocido** </categoria>* *<pre_procesado>* *<forma_canonica>**Amo**</forma_canonica>* *<prefijo>null</prefijo>* *<sufijo> null</sufijo>* *</pre_procesado >* *</entrada1>* *</XML>*

Para este caso, el aprendizaje ágil produce dos salidas posibles: una como verbo (amar), y otra como sustantivo (persona dueña de algo).

XML aprendizaje

```
<XML>
   <aprendizaje_lexmor>
         <categoria> verbo </categoria>
        <genero> null </genero>
         <tipo>transitivo</tipo>
        <numero>singular </numero>
        <modo> null </modo>
        <tiempo> null </tiempo>
        <aspecto> null </aspecto>
        <voz> null</voz>
        <persona> null </persona>
    </aprendizaje_lexmor>
</XML>
```

XML aprendizaje

```
<XML>
   <aprendizaje_lexmor>
         <categoria> sustantivo </categoria>
        <genero> masculino </genero>
         <tipo>común</tipo>
        <numero>singular</numero>
        <modo> </modo>
        <tiempo> </tiempo>
        <aspecto> </aspecto>
        <voz> </voz>
        <persona> </persona>
    </aprendizaje_lexmor>
</XML>
```

Como sustantivo se genera la siguiente salida:

Salida (entrada1) = lex_mor(***amo, sustantivo, masculino, común, singular****, null, null, null, null, null, null)*.

Para el caso de verbo pasa al *aprendizaje duro*, donde se completa la información y se generan varias estructuras como salida del aprendizaje:

Salida (entrada1) = lex_mor(***amo, verbo,*** *null,* ***común, singular, indicativo, presente,*** *null,* ***activa,*** **[1S]** ***yo,*** *null)*.

Salida (entrada1) = lex_mor(***amas, verbo,*** *null,* ***común, singular, indicativo, presente,*** *null,* ***activa,*** **[2S] tú***, null)*.

Salida (entrada1) = lex_mor(***ama, verbo,*** *null,* ***común, singular, indicativo, presente,*** *null,* ***activa,*** **[3S] el***, null)*.

Salida (entrada1) = lex_mor(***ama, verbo,*** *null,* ***común, singular, indicativo, presente,*** *null,* ***activa,*** **[3S] ella***, null)*.

Salida (entrada1) = lex_mor(***amamos, verbo,*** *null,* ***común, plural, indicativo, presente,*** *null,* ***activa,*** **[1p] nosotros***, null)*.

Salida (entrada1) = lex_mor(***amamos, verbo,*** *null,* ***común, plural, indicativo, presente,*** *null,* ***activa,*** **[1p] nosotras***, null)*.

Salida (entrada1) = lex_mor(***amaís, verbo,*** *null,* ***común, plural, indicativo, presente,*** *null,* ***activa,*** **[2p] vosotros***, null)*.

Salida (entrada1) = lex_mor(***amaís, verbo,*** *null,* ***común, plural, indicativo, presente,*** *null,* ***activa,*** **[2p] vosotras***, null)*.

Salida (entrada1) = lex_mor(***aman, verbo,*** *null,* ***común, plural, indicativo, presente,*** *null,* ***activa,*** **[3p] ellos***, null)*.

Salida (entrada1) = lex_mor(***aman, verbo****, null,* ***común, plural, indicativo, presente****, null,* ***activa,*** **[3p] ellas***, null)*.

Estas salidas se generan también para el tiempo pretérito imperativo, pretérito perfecto simple, futuro, y condicional. Para el modo imperativo del tiempo presente para todas las personas, el modo subjuntivo para los tiempos pretérito imperativo, pretérito perfecto simple y futuro en todas las personas.

6.3 APRENDIZAJE A PARTIR DE UN ADJETIVO

Para la entrada

Entrada1 = lex_mor(**buenos*****, Desconocido****, null, null, null, null, null, null, null, null)*.

EL aprendizaje simiente genera:

XML entrada1
<XML> *<entrada1>* *<metodo id="aprendizaje_lexmor"/>* *<termino>**buenos**</termino>* *<categoria>* ***Desconocido*** *</categoria>* *<pre_procesado>* *<forma_canonica>**bueno**</forma_canonica>* *<prefijo>null</prefijo>* *<sufijo>**s**</sufijo>* *</pre_procesado >* *</entrada1>* *</XML>*

Para esta entrada, el aprendizaje Ágil produce dos salidas, como plural y singular:

XML aprendizaje
<XML> <aprendizaje_lexmor>

<categoria> ***Adjetivo****</categoria>* <tipo>**positivo**</tipo> <genero> **masculino** </genero> <numero> **plural** </numero> <modo> null </modo> <tiempo> null </tiempo> <aspecto> null </aspecto> <voz> null </voz> <persona> null </persona> </aprendizaje_lexmor> </XML>
XML aprendizaje
<XML> <aprendizaje_lexmor> *<categoria>* ***Adjetivo****</categoria>* <tipo>**positivo**</tipo> <genero> **masculino** </genero> <numero> **singular**</numero> <modo> null </modo> <tiempo> null </tiempo> <aspecto> null </aspecto> <voz> null </voz> <persona> null </persona> </aprendizaje_lexmor> </XML>

Por consiguiente, se generan dos estructuras como resultado del aprendizaje.

Salida (entrada1) = lex_mor(***buenos, adjetivo, positivo, masculino, plural,*** *null, null, null, null, null, null)*.

Salida (entrada1) = lex_mor(***bueno, adjetivo, positivo, masculino, singular,*** *null, null, null, null, null, null)*.

6.4 APRENDIZAJE A PARTIR DE UN ADVERBIO

Para la entrada:

Entrada1 = lex_mor(**abajo**, ***Desconocido***, *null, null, null, null, null, null, null, null)*.

Se genera como salida del aprendizaje simiente:

XML entrada1
<XML> *<entrada1>* *<metodo id="aprendizaje_lexmor"/>* *<termino>**abajo**</termino>* *<categoria> **Desconocido** </categoria>* *<pre_procesado>* *<forma_canonica>**abajo**</forma_canonica>* *<prefijo>null</prefijo>* *<sufijo>null</sufijo>* *</pre_procesado >* *</entrada1>* *</XML>*

Una vez se ha creado con satisfacción el archivo XML, se pasa al aprendizaje Ágil, quien produce como salida:

XML aprendizaje
<XML> <aprendizaje_lexmor> <categoria> ***Adverbio***</categoria> <tipo>**lugar**</tipo> <genero> null </genero> <numero> null </numero> <modo> null </modo> <tiempo> null </tiempo> <aspecto> null </aspecto> <voz> null </voz> <persona> null </persona> </aprendizaje_lexmor> </XML>

Salida (entrada1) = lex_mor(***abajo, adverbio, lugar,*** *null, null, null, null, null, null, null, null)*.

6.5 TRAZA DE SALIDA PARA UNA PALABRA NO VALIDA

Entrada1 = lex_mor(**nvlda*, Desconocido**, null, null, null, null, null, null, null, null)*.

XML entrada1
<XML> *<entrada1>* *<metodo id="aprendizaje_lexmor"/>* *<termino>***nvlda***</termino>* *<categoria>* ***Desconocido*** *</categoria>* *<pre_procesado>* *<forma_canonica>***null***</forma_canonica>* *<prefijo>null</prefijo>* *<sufijo>null</sufijo>* *</pre_procesado >* *</entrada1>* *</XML>*

Una vez se han creado con satisfacción el archivo XML, se pasa al aprendizaje Ágil quien utiliza éste como fuente de aprendizaje y produce como salida:

XML aprendizaje
<XML> <aprendizaje_lexmor> *<categoria>* ***Desconocido*** *</categoria>* <tipo>null</tipo> <genero> null </genero> <numero> null </numero> <modo> null </modo> <tiempo> null </tiempo> <aspecto> null </aspecto> <voz> null </voz> <persona> null </persona> </aprendizaje_lexmor> </XML>

Salida (entrada1) = lex_mor(**nvlda***, **Desconocida,** null, null, null, null, null, null, null, null, null)*.

6.6 TRAZA DE SALIDA PARA UNA PALABRA DESCONOCIDA

Para la entrada

Entrada1 = lex_mor(**abe***, **Desconocido**, null, null, null, null, null, null, null, null)*.

El aprendizaje simiente genera:

XML entrada1
<XML> *<entrada1>* *<metodo id="aprendizaje_lexmor"/>* *<termino>***abe***</termino>* *<categoria> **Desconocido** </categoria>* *<pre_procesado>* *<forma_canonica>null</forma_canonica>* *<prefijo>null</prefijo>* *<sufijo>null</sufijo>* *</pre_procesado >* *</entrada1>* *</XML>*

Una vez se han creado con satisfacción el archivo XML, se pasa al aprendizaje Ágil, quien encuentra en Internet la palabra ”ave”, y produce como salida:

XML aprendizaje
<XML> <aprendizaje_lexmor> *<categoria>* null *</categoria>* <tipo> null </tipo> <genero> null </genero> <numero> null </numero> <modo> null </modo> <tiempo> null </tiempo>

```
        <aspecto> null </aspecto>
        <voz> null </voz>
        <persona> null </persona>
    </aprendizaje_lexmor>
</XML>
```

Salida (entrada1) = lex_mor(**abe***,* null**,** null*, null, null, null, null, null, null, null, null, **url**)*.

Este último caso devuelve las url donde consiguió el uso del término “ave”.

7. CONCLUSIONES Y RECOMENDACIONES

7.1 CONCLUSIONES

En esta tesis se presento una arquitectura para el aprendizaje automático de ontologías, capaz de adquirir nuevo conocimiento en función de la información suministrada por el proceso de análisis de una consulta en lenguaje natural, y/o de la información recuperada desde la web por dicha consulta. Esta arquitectura, a partir del nuevo conocimiento adquirido, permite potenciar las estructuras semánticas y léxicas de un Marco Ontológico Dinámico para una Web Semántica.

La arquitectura de aprendizaje de ontologías se caracteriza por integrar diferentes métodos y técnicas de descubrimiento de conocimiento léxico y semántico. Además, a través de la arquitectura es posible explotar cualquier recurso de información sea una palabra, una página web, una base de datos, una ontología, etc.

La arquitectura soporta, dentro del aprendizaje léxico, la gestión de errores, reportándolos, en algunos casos, con alguna información adjunta útil para su consideración (por parte de quien planteo la consulta), tal como una página web, documento de texto, etc., con contenido extra sobre lo que encontró, que considera semejante al error detectado.

Por otro lado, en cuento a los procesos de actualización, la arquitectura está diseñada para identificar, de manera inequívoca, la estructura a impactar o actualizar dentro del MODS, lo que asegura una consistencia en el proceso de aprendizaje.

La arquitectura es capaz de ofrecer a un sistema basado en conocimiento, o un sistema léxico tal como un tesauro[63], lexicón, o incluso un diccionario, la capacidad de adaptación y evolución por medio del aprendizaje a partir de la web. También, la arquitectura del componente de aprendizaje puede funcionar de manera independiente como un pequeño buscador web de información léxica, útil para lingüista o personas interesadas en conocer un poco más sobre las palabras del español.

Como se puede ver, se ha diseñado una arquitectura novedosa con capacidad para soportar una variedad de elementos de aprendizaje. Pero lograr su implementación completa ha mostrado ser un reto de gran complejidad. Una de las razones es que estamos frente a uno de los fenómenos más agudos que ocurren en la inteligencia, el *aprendizaje lingüístico*. En términos computacionales, esto implica la construcción de varios algoritmos que emulen procesos que para los humanos son fáciles, como por ejemplo saber si una palabra es un verbo o un sustantivo (por ejemplo: *comer* y *pan*), o la construcción de un axioma (por ejemplo inductivamente, como en: "hace una semana vi por primera vez un cisne y era blanco, hace 6 días vi otro[64] cisne y era blanco, …ayer vi otro cisne y era blanco, hoy vi otro cisne y era blanco, por lo tanto puedo inferir que todos los cisnes son blancos[65]") Los anteriores dos casos de aprendizaje ocurren dentro de lo que se han denominado aprendizaje morfosintáctico y aprendizaje semántico, en los eventos que llaman a la arquitectura de aprendizaje. Computacionalmente, crear algoritmos para considerar la infinidad de casos como los anteriores no es sencillo.

Con base a lo anterior se ha profundizado, en la parte del aprendizaje léxico en la fenomenología del verbo, y para el aprendizaje semántico en el aprendizaje de relaciones no-taxonómicas. Como resultado, se diseño un sistema flexionador de verbos capaz de diferenciar entre un verbo regular e irregular, y conjugarlo en

[63] Un tesauro es un repositorio de información morfosintáctica enriquecido.
[64] Como probar que no era el mismo
[65] En la vida real existen cisnes que no son blancos.

todas las formas personales e impersonales en varios tiempos; y con respecto al aprendizaje de relaciones no-taxonómicas, se diseño una propuesta para su aprendizaje basada en un método guiado por patrones. Ambos esquemas de aprendizaje tendrán fuentes de información distintas, la primera derivada del proceso de análisis de la consulta, y la segunda derivada de la información recuperada desde la web por la consulta.

El sistema flexionador de verbos es parte del componente de aprendizaje duro, el cual, a su vez, es parte del prototipo del componente de aprendizaje morfosintáctico implementado, que impacta el lexicón. El prototipo implementado tiene otros componentes (el aprendizaje simiente y el aprendizaje ágil), que le permiten aprender otras categorías lingüísticas: sustantivos, adjetivos y advervios.

Finalmente, una de las principales teorías sobre las cuales se baso este trabajo es conocida como ingeniería ontológica. Al respecto, podemos concluir que aunque existen varios métodos, metodologías y herramientas para el aprendizaje de ontologías desde diversas fuentes (siendo el texto la mas estudiada), no existe una metodología o método que guie de manera precisa el proceso de aprendizaje, solo proporcionan delineamientos generales. La mayoría de los métodos se basan en técnicas de procesamiento del lenguaje natural, usando en algunos casos corpus. Todos estos métodos requieren de la participación del ingeniero ontológico en partes del proceso. Con respecto a las herramientas, la mayoría se basan en clustering o agrupamiento, en enfoques estadísticos, o aproximaciones semánticas. Gran parte de estos métodos y herramientas se encuentras descritos en el marco teórico.

7.2 RECOMENDACIONES

A través del diseño de la arquitectura, y del desarrollo de un primer prototipo para el caso del aprendizaje morfosintáctico, se ha podido dilucidar una variedad de futuras investigaciones, dentro de las cuales se encuentran:

- Extender el sistema flexionador de verbos, incorporándole todos los casos de los verbos irregulares.
- Diseñar un mecanismo de gestión automática de las ontologías del MODS, a partir de la información que va aprendiendo. Por ejemplo, para el caso de conceptos determinar cuando un concepto nuevo es más o menos general que otro ya existente en una de las ontologías, o chequear que el concepto aprendido no sea contradictorio con otro existente, entre otras cosas.
- Crear un prototipo del componente de aprendizaje que soporte el aprendizaje de relaciones no-taxonómicas, sobre la base del método diseñado en este trabajo.
- Diseñar e implementar las otras formas de aprendizaje semántico no consideradas en este trabajo (relaciones taxonómicas, etc.)

REFERENCIAS BIBLIOGRÁFICAS

[1] Rodriguez, T. "Marco Ontológico Dinámico Semántico para la Web Semántica". Propuesta de tesis Doctoral. Universidad de los Andes (Mérida-Venezuela). 2009.

[2] Nirenburg S. y Raskin V. "Ontological Semantics". Cambrige, MA: The MIT Press, 2004. ISBN 0-262-14086-1.

[3] Gómez A, Fernández M y Corcho O. "Ontological engineering: with examples from the areas of knowledge management, e-commerce and the Semantic Web", London, Springer, 2004. ISBN 1852335513.

[4] Steffen Staab y Studer Rudi. "Handbook on Ontologies". Springer-Verlag Berlin Heidelberg 2004 New York. ISBN 3-540-40834-7

[5] McGuinnes D, Fiker R, Rice J, y Wilder S."An environment for merging and testing large ontologies". En principios de representación y razonamiento de conocimiento (KR2000). Morgan Kaufmann Publishers, San Francisco, CA 2000.

[6] Euzenat J y Shvaiko P. "Ontology Maching",Springer-Verlang, 2007, ISBN: 978-3-540-49611-3.

[7] Kalfoglou Y, y Schorlemmer M. "Ontology Mapping: the state of the art",.Journal on Data Semantics, 2003. ISBN 978-3-540-39733-5.

[8] Reed S, y Bernstein A. "Mapping ontologies into CyC". In proceedings of the AAA'02 workshop on Ontologies and the Semantic Web, Edmonton, Canada, 2002.

[9] Noy N.F, y Musen M. "PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment". In Proceedings of the 17th National Conference on Artificial Intelligence (AAAI'00), Austin, TX, USA, July 2000.

[10] Craven M.,DiPasquo D.,et al., "Learning to construct knowledge bases from the World Wide Web". Artificial Intelligence, 2000.

[11] Mikheev A., y Finch S., "A Workbench for finding structure in text". In proceedings of the 5th conference on Applied Natural Porcessing-ANLP'97, March 1997, Washington DC, USA, Pag 372-379, 1997.

[12] Hahn U., y Romacker M. “Content management in the syndicate system”. How technical documents are automatically transformed to text knowledge bases. Data & Knowledge Engineering, 2000.

[13] Enguehard C, “ANA: Apprentissage Naturel Automatique d’un réseau sémantique”, Tesis Doctoral, UTC, 1992

[14] Evans DA et al., “A Report on CLARIT TREC-2001 Experiments”. In EM Voorhees and DK Harman (Editores), The tenth Text Retrieval Conference (TREC-2001). NIST Special Publication Office. 2002.

[15] Takeuchi k., et al.,”Construction of Grammar-based Term Extraction Model for Japanese”, In proceedings of COMPUTERM2004, pp.91-94, 2004.

[MARSID] Agrawal, R, et al.,.“Mining Associations between Sets of Items in Massive DataBases” registrado en las actas del congreso ACM SIGMOD sobre Gestión de datos, realizado en Washington, DC., May 26-28, 1993.

[16] Faure D, Poibeau T.,” First experiments of using semantic knowledge learned by ASIUM for information extraction task using INTEX”. In: S. Staab, A. Maedche, C. Nedellec, P. Wiemer-Hastings (eds.), Proceedings of the Workshop on Ontology Learning, 14th European Conference on Artificia Intelligence ECAI’00, Berlin, Germany, 2000.

[17] Yamaguchi T., “Constructing domain ontologies based on concept drift analysis”. In Proceedings of IJCAI-99 Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends, in conjunction with the Sixteenth International Joint Conference on Artificial Intelligence, August, Stockholm, Sweden. 1999.

[18] Aussenac-Gilles N. y Seguela P. “Les relations sémantiques: du linguistique au formel”. Cahiers de grammaire. N° spécial sur la linguistique de corpus. A. Condamines (Ed.) Vol 25. Toulouse : Presse de l'UTM. 2000

[19] Randall D., Howard S., “What is a Knowledge Representation”. http://groups.csail.mit.edu/medg/ftp/psz/k-rep.html (Citada el 9 de marzo 2010).

[20] Missikoff M., Navigli R., y Velardi P. “ The Usable Ontology: An Environment for Building and Assessing a Domain Ontology”. Research paper at International Semantic Web Conference (ISWC) 2002, June 9-12th, Sardinia, Italia, 2002.

[21] Grupo de Investigación AKSW (Agile Knowledge Engineering and Semantic Web). Universidad de LEIPZIG Germany. Proyecto DL-LEARNER http://aksw.org/projects/DLLearner?v=133g (Citada el 1 de abril 2010).

[22] Chaelandar G, y Grau B. “SVETLAN’- A System to Classify Words in Context”. In S. Staab, A. Maedche, C. Nedellec, P. Wiemer-Hastings (eds.) Proceedings of the Workshop on Ontology Learning, 14th European Conference on Artificial Intelligence ECAI’00, Berlin, Germany, August 20-25. 2000.

[23] Jones, S. y Paynter, G.W. “Automatic extraction of document keyphrases for use in digital libraries: evaluation and applications”. Journal of the American Society for Information Science and Technology (JASIST). 2002.

[24] Shamsfard M., y Barforoush A., “An introduction to Hasti: An Ontology learning System”. http://www.actapress.com/PaperInfo.aspx?PaperID=260428reason=500 (Citado el 8 mayo 2010).

[25] Moreno L., et al., “Introducción al procesamiento de lenguaje natural”. Servicios de publicaciones Universidad de Alicante. 1999.

[26] Valencia R., “Un entorno para la extracción incremental de conocimiento desde texto en lenguaje natural”. Tesis Doctoral. Universidad de Murcia, Departamento de ingeniería de la información y las comunicaciones., 2005.

[27] Daille, B. “Study and implementation of combined techniques for automatic extraction of terminology”. In Judith L. Klavans and Philip Resnik, editors, The Balancing Act: Combining Symbolic and Statistical Approaches to Language. MIT Press, Cambridge, MA, 49-66. 1996.

[28] Jacquin, C. y Liscouet, M. 1996. “Terminology extraction from texts corpora: application to document keeping via Internet”. In: TKE ‘96: Terminology and Knowledge Engineering, 74-83. Berlin: Indeks Verlag. http://www.limsi.fr/Individu/jacquemi/FASTR/index.html, (Citado 12 mayo 2010).

[29] Heid, U., et al.“Term extraction with standard tools for corpus exploration. Experience from German”. In: TKE ‘96: Terminology and Knowledge Engineering,, 139-150. Berlin: Indeks Verlag. 1996.

[30] Frantzi, K. y Ananiadou, S. “Statistical measures for terminological extraction”. Working paper of the Department of Computing of Manchester Metropolitan University. 1995.

[31] Bourigault, D. “LEXTER, un Logiciel d'EXtraction de TERminologie. Application à l'acquisition des connaissances à partir de texts”. Tesis Doctoral. Paris: École des Hautes Études en Sciences Sociales. 1994.

[32] Plante, P. y Dumas, L “Le Dépoulliment terminologique assisté par ordinateur”. Terminogramme, 46, 24-28. . 1998.

[33] Justeson, J. and Katz, S. “Technical terminology: some linguistic properties and an algorithm for identification in text”. Natural Language Engineering, 1, 1: 9-27. 1995.

[34] Maedche, A. y Volz, R. “The Text-To-Onto Ontology Extraction and Maintenance Environment”. To appear in Proceedings of the ICDM Workshop on integrating data mining and knowledge management, San Jose, California, USA. 2001.

[35] Alexander M., y Steffen S. “Discovering conceptual relations from text”. In W. Horn, editor, Proceedings of the 14th European Conference on Artificial Intellignece (ECAI’2000), 2000.

[36] Martinez P, García S. “A Knowledge-based Methodology applied to Linguistic Engineering”. In R. Nigel Horspool Ed., Systems implementation 2000: Languages, Methods y Tools. London. 1998.

[37] Ferrández O,. Izquierdo R. “Un sistema de búsqueda de respuesta basado en ontologías, implicación textual y entornos reales”. http://www.sepln.org/revistaSEPLN/revista/41/sec2-art1.pdf (Citado 3 junio 2010).

[38] Puerto E, et al “La expresiva SHIQ como lenguaje ontológico para la Web Semántica”. Revista Colombia de Tecnologías de Avanzada. ISSN: 1692-7257. Vol:1 fasc:7 pag 78-83, 2006.

[40] Maedche A. y Staab S. Ontology Learning. In S. Staab & R. Studer (eds) “Handbook on Ontologies in Information Systems”. Springer 2003. http://www.aifb.uni-karlsruhe.de/WBS/sst/Research/Publications/handbook-ontology-learning.pdf 2003.

[41] Maedche, A. y Volz, R. “The Text-To-Onto Ontology Extraction and Maintenance Environment”. To appear in Proceedings of the ICDM Workshop on integrating data mining and knowledge management, San Jose, California, USA. 2001.

[42] Basterrechea E. y Rello L. “el verbo en español”. Primera edición febrero del 2010. ISBN: 978-84-93770600.

[43] Jiang, T., Tan, A., Wang, K.: Mining generalized associations of semantic relations from textual web content. IEEE Transactions on Knowledge and Data Engineering. 2007

[44] Lin D. y Pantel P. Concept discovery from text. In Proceedings of the International Conference on Computational Linguistics (COLING), 2002.