Universitat d'Alacant
Universidad de Alicante

Búsqueda de Respuestas en Dominios Restringidos: aplicación sobre el dominio agrícola

Katia Vila Rodríguez

Tesis Doctorales

www.eltallerdigital.com

UNIVERSIDAD de ALICANTE

Universitat d'Alacant
Universidad de Alicante
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante

Búsqueda de Respuestas en Dominios Restringidos: aplicación sobre el dominio agrícola

Katia Vila Rodríguez

Memoria para optar al grado de Doctor en Informática bajo la dirección de

Dr. Antonio Ferrández Rodríguez

Alicante, 8 de octubre de 2010

Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante

Búsqueda de Respuestas en Dominios Restringidos: aplicación sobre el dominio agrícola

Katia Vila Rodríguez

Memoria para optar al grado de Doctor en Informática bajo la dirección de

Dr. Antonio Ferrández Rodríguez

Alicante, 8 de octubre de 2010

En la actualidad, este trabajo ha sido llevado a cabo bajo los proyectos PROMETEO/2009/119 y Textmess 2.0 (TIN2009-13391-C04-01) del Gobierno Valenciano y de España, respectivamente. Los autores agradecen al gobierno español y valenciano su apoyo económico.

Erase una vez, un día lluvioso,
con un café en la mano,
un hombre que amo me dijo:
nada es en vano, todo es por algo;
nada se pierde, todo se transforma.
Este trabajo es una prueba de ello.
Por eso y más,
para TI...

Agradecimientos

Durante y después de un trabajo que conlleve un gran esfuerzo y sacrificio, como en la vida misma, hay muchas personas con las que reír o llorar, abrazar o besar, agradecer verbalmente o en silencio...Sólo me robo un pedacito de este trabajo para dejar inmortalizadas a aquellas personas más presentes en este período y que me han ayudado en este trabajo. A:

Los tres Jose que siempre han estado ahí cuando los he necesitado: Jose Norberto un beso grande por todo el apoyo emocional y científico a lo largo de este trabajo, Jose Ignacio un 1/2 beso por aguantarme siempre al otro lado del teléfono, y Jose Manuel muchas gracias por tu apoyo en la investigación, por Shani, por la Variante Disléxica...

Mi tutor Antonio por sus opiniones y consejos en los momentos precisos, por la confianza depositada en mi, por esa total independencia en mi trabajo y apoyo en las decisiones que tomaba.

Tony y Josval, de la UMCC (Cuba), por el trabajo científico conjunto. Tony muchísimas gracias por todo, porque a ti te debo parte de mi formación científica.

Javi gracias por el diseño de esta tesis, me ha encantado!!!! y siguiendo tu consejo aclaro: "que no ha sido dañado ningún animal para realizar este diseño de portada, sólo son copias de las manchas de una vaquita".

Los miembros del Departamento de Informática de la Universidad de Matanzas (Cuba): a los que hoy son colegas de trabajo y ayer fueron mis profesores o compañeros de estudio.

Los miembros del DLSI: las personas que siempre me apoyaron y garantizaron los detalles logísticos de mis estancias en Alicante (Manolo, Rafa, Jesús Peral, y Patricio), mis colegas del laboratorio, los que siempre preguntaban como lo llevaba (Ester, Elena, Rubén, Alexandra, Jesús, Sergio, Felipe, Paul, Octavio, Héctor, Paloma, Montoyo), mis compañeros y profesores de los cursos de doctorado, el equipo de la Variante Disléxica (Jose Manuel, Javi, Helena, David, Borja) por los buenos ratos, la gente de secretaría (Diego, Chento, Nieves, María), a todos.

Los amigos que siempre se han preocupado por mi y porque saliera este trabajo: Massiel, Yolanda, Eli, Pirra y familia, Firas y María, Ariel, Robe, Roly, Chuchi, Ferna, Kabir, Raisel, Oscar, Marta, Jorge, Daniela, Carlo y a todos aquellos que no menciono pero que han sido muy importantes en determinadas etapas de mi vida.

Mi Ma y mi Pa, un beso grande y un abrazo, gracias por el eterno apoyo en mis empeños y decisiones, por estar siempre a mi lado, por la educación y el amor que siempre me han dado...los amo!!!!!

Mi sobri bello Abraham, a mi herma que tanto amo Vania, y a mi cuñi Ernesto, hasta a ti te ha tocado algo de este trabajo, y vivir todos juntos el estrés del primer borrador fue muy hermoso.

Mi familión: a todos en Pueblo Nuevo (Yianella, Leyra, Leysy, Caridad, Jesús, tía Herda, tía Rosa, tío Pedrín, Miriam); a mis primos Adolfito, Rub, Camelia y Bibiana por tantas noches de tertulia; a quienes fueron y serán como mi familia (Bertha, Emilio[2], Jorge, MC, EE, a todos); a mi primi loco Armandito.

Parte I. Introducción

1. Introducción 3
1.1. Motivaciones 8
1.2. Objetivos 13
1.3. Estructura de la tesis 15

Parte II. Estado de la cuestión

2. Sistemas de Búsqueda de Respuestas 21
2.1. Origen y perspectiva histórica de los sistemas de BR 21
2.2. Sistemas de BR en Dominios Abiertos 25
2.3. Sistemas de BR en Dominios Restringidos 28
2.3.1. Importancia y actualidad 30
2.3.2. Estado actual de los sistemas de BR en dominios restringidos 31
2.3.3. Clasificación para los sistemas de BR-DR . . . 33
2.3.4. Descripción de sistemas de BR de dominio restringido 35
2.4. Conclusiones 56

3. Estado Actual de la Adaptación de la Búsqueda de Respuestas a Dominios Restringidos 57
3.1. Tratamiento del ruido textual 57
3.1.1. Definición y orígenes del ruido textual 58

3.1.2. Aplicaciones de PLN afectadas por el ruido textual 64
3.1.3. Tratamiento del ruido textual en sistemas de RI 70
3.2. Taxonomías de tipo de respuesta esperada 79
3.3. Adaptación de patrones para sistemas de Búsqueda de Respuestas 84
3.4. Conclusiones 85

Parte III. Propuesta de Adaptación de Búsqueda de Respuestas a Dominios Restringidos

4. Estrategia de tolerancia al ruido textual para sistemas de recuperación de información en dominios restringidos 89
4.1. Algoritmos de Distancia de Edición 91
4.1.1. Distancia de Levenshtein 91
4.1.2. Distancia Needleman-Wunsch 92
4.1.3. Distancia Jaro-Winkler 92
4.1.4. Distancia de Edición Extendida 93
4.2. Discusión sobre conveniencia de las Distancias de Edición 98
4.2.1. Desventajas de las Distancias de Edición . . . 99
4.2.2. Comparación entre las Distancias de Edición 100
4.2.3. Motivación del uso de DEx para el cálculo de similitud entre palabras 109
4.3. Extensión de la DEx para multipalabras 111
4.4. Descripción de nuestra estrategia de tolerancia al ruido textual en la RI 115
4.4.1. Obtención del vector terminológico de cada término indexado (Etapa 1) 116
4.4.2. Obtención del vector terminológico de cada término relevante en la pregunta (Etapa 2) . . 119
4.4.3. Ejecución de dos procesos de RI independientes (Etapa 3) 119
4.5. Conclusiones 123

5. Método de adaptación de un sistema de búsqueda de respuestas de dominio abierto a dominios restringidos 127
5.1. Desarrollo dirigido por modelos 128
5.1.1. Modelos 128
5.2. Adaptación dirigida por modelos de SBR a dominios restringidos 138
5.2.1. Obteniendo el modelo de dominio restringido a partir del corpus 139
5.2.2. Enriquecimiento del modelo de dominio restringido. 147
5.2.3. Obteniendo taxonomías de TRE a partir del Modelo de Dominio Restringido enriquecido . 152
5.2.4. Obteniendo modelos de patrones de preguntas y respuestas 159
5.2.5. Adaptación de modelos de patrones de pregunta a un dominio restringido. 163
5.2.6. Adaptación de modelos de patrones de repuesta a un dominio restringido. 170
5.2.7. Generando el código de los nuevos patrones de preguntas y respuestas. 175
5.3. Conclusiones 177

6. MARAQA: una herramienta para la adaptación dirigida por modelos de sistemas de búsqueda de respuestas a dominios restringidos 181
6.1. Visión general de Eclipse 181
6.2. Arquitectura general de MARAQA 183
6.2.1. Metamodelos 185
6.2.2. Transformaciones 185
6.3. Conclusiones 193

Parte IV. Evaluación de la propuesta: aplicación al dominio agrícola

7. Evaluación 197
7.1. Recursos empleados en los experimentos 198
7.1.1. AliQAn: sistema de BR-DA inicial para la evaluación 198
7.1.2. JIRS: sistema de RI empleado en la evaluación 205
7.1.3. Tesauro AGROVOC: SOC de dominio restringido 206
7.1.4. Revista Cubana de Ciencia Agrícola 208
7.1.5. Corpus textual RCCA 209
7.1.6. Colección de preguntas RCCA 211
7.2. Medida de evaluación 212
7.3. Marco comparativo para evaluar nuestra propuesta: experimentos previos 212
7.3.1. Problemas en la aplicación de JIRS y AliQAn al dominio agrícola 213
7.3.2. Tipología de errores 216
7.3.3. Discusión 217
7.4. Evaluación de la estrategia de tolerancia al ruido textual para Sistemas de Recuperación de Información en Dominios Restringidos 218
7.4.1. Experimento para medir la efectividad de la distancia DM 219
7.4.2. Experimento para determinar los valores límites 221
7.4.3. Experimento para evaluar nuestra propuesta 223
7.4.4. Discusión de los resultados 223
7.5. Evaluación del Método de Adaptación de SBR-DA a dominios restringidos 225
7.5.1. Experimentos sobre la generación de taxonomías de TRE para dominios restringidos . . 225
7.5.2. Experimentos sobre adaptación de patrones a dominios restringidos 227
7.6. Conclusiones 228

Parte V. Conclusiones finales y trabajos futuros

8. Conclusiones 233
8.1. Principales aportaciones y conclusiones 234
8.2. Producción científica 236
8.2.1. Búsqueda de Respuestas en Dominios Abiertos 236
8.2.2. Búsqueda de Respuestas en Dominios Restringidos y Recursos de Conocimiento 237
8.2.3. Propuesta de tolerancia al ruido textual en el proceso de RI 237
8.2.4. Propuesta de adaptación de un SBR-DA a dominios restringidos 238
8.3. Trabajos futuros 240

Parte VI. Bibliografía consultada

Referencias 243

Parte VII. Anexos

A. Corpus de Preguntas RCCA 263
A.1. Preguntas sobre dominio agrícola para evaluar BR-DR español–español 263

B. Acrónimos 273
B.1. Glosario de acrónimos usados en la memoria 273

C. Taxonomía de TRE para el dominio agrícola 275
C.1. Taxonomía de tipo de respuestas esperadas generada por nuestra propuesta para el dominio agrícola 275

2.1. Propuestas de los SBR-DR según la clasificación definida 36
2.2. Comparación entre Extrans y el sistema *baseline* SMART. 38
2.3. Ejemplo de plantilla del SBR-DR para pronósticos del tiempo 42
2.4. Ejemplo de consulta SQL generada por el SBR-DR para pronósticos del tiempo 42
2.5. Resultados obtenidos por el el SBR-DR para pronósticos del tiempo 42
2.6. Resultados de los experimentos del BR-DR de *Bell Canada* 46
2.7. Ecuación empleada en la evaluaciones del SBR-DR para Bell Canada 47
2.8. Ejemplo de forma lógica del SBR-DR para entorno médico 52

3.1. Resumen de las principales propuestas de taxonomías de TRE. 82

4.1. Ejemplo paso 1 de la DEx: obtención de la matriz de Levenshtein. 94
4.2. Ejemplo paso 2 y 3 de la DEx: obtención de la LCS y de la CO. 95
4.3. Ejemplo paso 4 de la DEx: evaluación de la Ecuación 4.3 con la CO y los caracteres implicados. 98
4.4. Conjunto de 130 palabras (limpias o ruidosas) relevantes para el pivote “bacterias”. 104
4.5. Lista de las primeras 50 palabras devueltas por DEx. . . 105

4.6. Palabras ruidosas recuperadas por DEx que tienen relación con el pivote "bacterias". 107
4.7. Palabras recuperadas por las diferentes distancias a partir de un pivote con ruido "bacteriascapacesde". . . . 108
4.8. Ejemplo del cálculo de la similitud entre dos cadenas comparadas. 113
4.9. Matriz para la comparación entre el pivote "afrecho de trigo" y la palabra "afrechillo". 113
4.10. Ejemplo paso 7 de la DM: evaluación de la Ecuación 4.4. 114
4.11. Ejemplo de vector terminológico definido para la palabra "esponjasintravaginales" 118
4.12. Ejemplo de vector terminológico definido para la palabra "esponjasintravaginales" 119

7.1. Ejemplo de extracción de BS 200
7.2. Ejemplo de análisis de pregunta por el sistema AliQAn 203
7.3. Ejemplo de extracción de la respuesta por el sistema AliQAn 204
7.4. Evaluación del sistema AliQAn con preguntas del CLEF 205
7.5. Problemas generados en la aplicación de AliQAn al dominio agrícola. 217
7.6. Palabras recuperadas por las diferentes distancias con el pivote "bacteria". 219
7.7. Resumen estadístico de la evaluación de la estrategia de tolerancia al ruido para la RI en dominios restringidos. 222
7.8. Resultados de la evaluación de la estrategia de tolerancia al ruido para la RI en dominios restringidos. . . . 223
7.9. Resumen estadístico del Modelo de Dominio Restringido y la taxonomía de TRE creados (# clases semánticas). 226

B.1. Acrónimos usados en la memoria 274

4.1. DEx entre el pivote “enseñar” y un fragmento de su familia de palabras. 101
4.2. DEx entre el pivote “fabricar” y un fragmento de su familia de palabras. 102
4.3. DEx entre el pivote “cabalgar” y un fragmento de su familia de palabras. 102
4.4. Comparación entre DEx y otros algoritmos de distancia entre palabras (precisión a las n palabras recuperadas). 105
4.5. Comparación entre DEx y otros algoritmos de distancia entre palabras (cobertura a las n palabras recuperadas). 106
4.6. Comparación entre DEx y otros algoritmos sobre la cantidad de palabras con ruido recuperadas (a las n palabras recuperadas). 107
4.7. Comparación entre DEx y otros algoritmos sobre la precisión de palabras recuperadas a partir de un pivote con ruido. 108
4.8. Visión general de la aproximación para adicionar tolerancia a ruido en un sistema de RI. 117
4.9. Ejemplo de correspondencia entre un término de la consulta y otro del corpus con ruido. 122

5.1. Ejemplo de estructura jerárquica de MDD. 132
5.2. Ejemplo de transformación en MDD. 133
5.3. Ejemplo de relación QVT. 134
5.4. Ejemplo nuestra propuesta basada en MDD para SBR. 137

5.5. Propuesta dirigida por modelos para adaptar sistemas de BR a dominios restringidos. 140
5.6. Visión general de nuestro Metamodelo de Dominio Restringido. 141
5.7. Parte del modelo de dominio restringido de nuestro ejemplo. 147
5.8. Visión general de nuestro Metamodelo de Dominio Restringido Enriquecido. 148
5.9. Descripción de los pasos para obtener el modelo de Dominio Restringido y la taxonomía de TRE. 149
5.10. Ejemplo del enriquecimiento de parte del modelo de dominio restringido de nuestro ejemplo. 151
5.11. Regla de transformación que permite iniciar la obtención de un modelo para la taxonomía de TRE. 154
5.12. Regla de transformación que permite obtener los conceptos tope para la taxonomía de TRE. 155
5.13. Definición en OCL del criterio de granularidad de la taxonomía de TRE. 155
5.14. Regla de transformación que permite obtener los conceptos hipónimos de los nuevos conceptos de la taxonomía de TRE. 156
5.15. Fragmento de la taxonomía de TRE para dominio abierto y restringido. 157
5.16. Vista General del Metamodelo Patrones de Preguntas. . 159
5.17. Vista General del Metamodelo de Patrones de Respuestas. 161
5.18. Ejemplo de modelo obtenido para un patrón de pregunta de un SBR-DA. 162
5.19. Ejemplo de modelo obtenido para un patrón de respuesta de un SBR-DA. 163
5.20. Creación de un modelo de patrones de pregunta. 164
5.21. Creación de un patrón de pregunta adaptado a partir de uno existente. 166
5.22. Funciones auxiliares en OCL para la adaptación de patrones de pregunta. 166
5.23. Creación de los nuevos conceptos pertenecientes al modelo de patrón de pregunta adaptado. 167

5.24. Creación de las nuevas asociaciones pertenecientes al patrón de pregunta adaptado. 167
5.25. Creación de las nuevas expresiones pertenecientes al patrón de pregunta adaptado. 168
5.26. Creación de las nuevas expresiones pertenecientes al patrón de pregunta adaptado. 169
5.27. Ejemplo de modelo de patrón de pregunta obtenido para un patrón de pregunta de un SBR-DA. 171
5.28. Creación de un modelo de patrones de respuesta. 172
5.29. Creación de un patrón de respuesta adaptado a partir de uno existente. 173
5.30. Creación de los nuevos conceptos pertenecientes al modelo de patrón de respuesta adaptado. 174
5.31. Ejemplo de modelo de patrón de respuesta obtenido para un patrón de pregunta de un SBR-DA. 176

6.1. Arquitectura de la plataforma Eclipse. 182
6.2. Arquitectura general de MARAQA. 184
6.3. Extracto del metamodelo ECore. 186

7.1. Arquitectura general del sistema de BR monolingüe español, AliQAn 201
7.2. Comparación entre DM y otros algoritmos de distancia entre palabras (precisión a las n palabras recuperadas). 220
7.3. Comparación entre DM y otros algoritmos de distancia entre palabras (cobertura a las n palabras recuperadas). 220

Parte I

Introducción

Capítulo 1
Introducción

En la década de 1980 empezaron a expandirse por todo el mundo aquellas tecnologías que permitirían una infraestructura descentralizada de redes de computadoras, hoy conocida como Internet. En los noventa, con la finalidad de acceder e intercambiar datos, información y conocimiento a través de Internet se introduce la World Wide Web (comúnmente conocida como "la Web"). Con el paso de los años Internet y la Web lograron penetrar todos los países del mundo, independientemente de su cultura e idioma, creando un acceso mundial a información y comunicación sin precedentes. Desde entonces, la Web ha ido evolucionando de manera natural e inevitable, partiendo de la conocida Web 1.0 caracterizada por un estado más estático, donde los datos difícilmente cambiaban o se actualizaban, hacia la actual Web 2.0 con aplicaciones web más dinámicas y enfocadas al usuario final no sólo como consumidor sino también como productor de información.

El término Web 2.0 está comúnmente asociado con un fenómeno social, basado en la interacción que se logra a partir de diferentes aplicaciones Web, que facilitan el compartir información, la interoperatividad, el diseño centrado en el usuario y la colaboración en la Web. Por lo tanto, se puede afirmar que la Web 2.0 es una actitud y no precisamente una tecnología. Posteriormente otro estado evolutivo de la Web que se espera alcanzar es la Web Semántica, también llamada Web 3.0 o Web inteligente, cambiando las ac-

tuales folcsonomías [1] de la Web 2.0 por estándares de metadatos como las ontologías.

Todo este proceso de evolución natural que ha tenido y seguirá teniendo la Web nos confirma la necesidad de que toda aplicación, que actúe sobre la información volátil que Internet ofrece, cumpla una constante fundamental: "la habilidad para enfrentar el cambio". La aplicaciones no deben actuar como un "jardín cerrado", por el contrario, la información debe poderse introducir y extraer fácilmente.

Por otro lado, a partir de este desarrollo y evolución de la Web el número de aplicaciones Web y la cantidad de información que ofrecen sigue creciendo vertiginosamente. Debido a este hecho, el procesamiento, acceso y recuperación a la información requerida tiene una dificultad inherente. Sumemos al problema de la sobreabundancia de información la heterogeneidad en los formatos de acceso y gestión de la misma, para concluir que existe un problema obvio a la hora de localizar una información concreta. Para vencer estos aspectos, los usuarios necesitan de aproximaciones que le permitan acceder a información precisa desde el conjunto de los diferentes recursos de conocimiento o desde repositorios de información textual de una forma transparente y simple.

En este sentido la primera solución aportada por la comunidad científica, para tratar con el problema del acceso sencillo y rápido a la inconmensurable cantidad de información digital accesible a toda clase de usuario, fue la Recuperación de Información (RI) o *Information Retrieval* (Baeza-Yates & Ribeiro-Neto, 1999). La RI es la tarea de seleccionar de entre un repositorio de documentos aquellos que tengan mayor relevancia para una consulta realizada por un usuario. A veces, el documento puede ser reemplazado por el título, una lista de palabras claves y/o resumen u otros metadatos, aunque actualmente lo más común es utilizar la totalidad de los textos, muchas veces subdivididos en pasajes, cada uno de los cuales sirve como documento independiente a los efectos de

[1] Folcsonomía o folksonomía, de acuerdo con su formación etimológica, folcsonomía (folc+taxo+nomía) significa literalmente "clasificación democrática o gestionada por el pueblo". Es una indexación social, es decir, la clasificación colaborativa por medio de etiquetas simples en un espacio de nombres llano, sin jerarquías ni relaciones de parentesco predeterminadas.

recuperación. En estos momentos, los sistemas de RI más conocidos son los que actúan sobre Internet y localizan información en la Web, por ejemplo algunos motores de búsqueda como Google[2], Yahoo[3], AltaVista[4] o Ask[5]. Destacar que la entrada de estos sistemas es una consulta que por lo general consiste en una lista de palabras claves, y en casos muy excepcionales, en la pregunta completa formulada en lenguaje natural. Según (Russell & Norvig, 2003), este modelo de RI corresponde casi totalmente a un nivel de palabras, ya que acepta una cantidad mínima de sintaxis que se refiere a las palabras que deben aparecer una junto a la otra, e igualmente acepta un papel diminuto de clases semánticas en forma de listas de sinónimos. Por ello los resultados pueden ser en muchas ocasiones documentos que presentan muchos términos comunes con la consulta, pero que pueden no contener la respuesta deseada. Por otro lado, la salida es una lista de documentos ordenada en función de medidas de similitud con la pregunta, intentando así resolver las necesidades requeridas por el usuario, pero luego resta una ardua tarea ya que el usuario debe revisar y leer cada documento, lo primero para ver si en realidad está relacionado con los requerimientos solicitados y lo segundo para localizar la información puntual que se desea en su interior.

Precisamente, sin negar la utilidad y capacidad de estos sistemas para localizar información relevante en la Web, los inconvenientes mencionados impulsaron la investigación en sistemas capaces de superarlos, conocidos como sistemas de Búsqueda de Respuesta (BR) o *Question Answering*. Un sistema de BR tiene como objetivo la obtención de respuestas concretas a preguntas precisas indicadas por el usuario directamente en lenguaje natural. Por tanto, son sistemas especialmente útiles en situaciones donde el usuario necesita conocer un pedazo específico de información, y no desea consumir mucho tiempo para leer toda la información disponible relacionada con el tópico que busca, en función de resolver un problema. La investigación actual –incentivada a partir de

[2] http://www.google.com/
[3] http://www.yahoo.com/
[4] http://www.altavista.com/
[5] http://www.ask.com/

1999 por conferencias como el TREC[6] (Voorhees, 1999), y luego por el CLEF[7] y NTCIR[8]– se ha enfocado en la BR desde la perspectiva de RI, donde el objetivo es encontrar pequeños extractos de texto que contengan la respuesta dentro de documentos o pasajes a partir de tipos específicos de preguntas-respuestas (también llamadas preguntas factuales) que son fácilmente evaluables. Para ello usan métodos rápidos y poco profundos que son generalmente independientes del dominio de aplicación. De esta forma se le ha dado un impulso inmenso al desarrollo de Sistemas de Búsqueda de Respuesta en Dominios Abiertos (SBR-DA) sobre información textual.

Sin embargo, el interés por la investigación en sistemas de BR comienza mucho antes, alrededor de los años sesenta, pero desde la perspectiva de la Inteligencia Artificial (IA). Desde este punto de vista, la tarea se centraba fundamentalmente en el desarrollo de sistemas, conocidos como Interfaces en Lenguaje Natural de acceso a Bases de Datos (ILNBD), que respondieran a preguntas usando conocimiento codificado en bases de datos como recursos de información. Obviamente, estos sistemas sólo proveían respuestas concernientes sobre la información previamente codificada en la base de datos. No obstante tenían como beneficio el uso de técnicas avanzadas como demostración de teoremas y razonamiento profundo para ocuparse de necesidades complicadas de información, ya que tenían un modelo conceptual del dominio de aplicación representado en la estructura de la base de datos.

Tanto la tendencia de BR basada en conocimiento estructurado como la tendencia de BR basada en textos, desde la visión de IA y RI respectivamente, se han desarrollado en paralelo. La primera está bien adaptada para aplicaciones que operan preguntas complejas en un ambiente de información estructurado, mientras que la segunda se adecúa a aplicaciones genéricas de propósito amplio tratando con preguntas objetivas simples. Por lo que se pudiera pensar que representan fines opuestos. Pero, ¿cómo haríamos

[6] TREC, *Text Retrieval Conference*. `http://trec.nist.gov/`

[7] CLEF, *Cross-Language Evaluation Forum*. `http://www.clef-campaign.org/`

[8] NTCIR, *NII Test Collection for IR Systems*. `http://research.nii.ac.jp/ntcir/index-en.html`

para desarrollar una aplicación real que maneje preguntas complejas? ¿cómo hacer que combine información específica del dominio usualmente expresada en diferentes fuentes estructuradas, semi-estructuradas, o no estructuradas? Si se intenta responder a estas preguntas usando alguna de las dos tendencias de manera independiente nos encontraremos con serios inconvenientes. Sin embargo, la convergencia entre las dos tendencias puede ser la solución a estos problemas, a través del desarrollo de Sistemas de Búsqueda de Respuesta en Dominios Restringidos (SBR-DR). La característica principal de la Búsqueda de Respuestas en Dominios Restringidos (BR-DR) es la integración de recursos de conocimiento específicos del dominio, que han sido desarrollados para la BR o con otros propósitos, con la finalidad de obtener un sistema de BR óptimamente adaptado a un área del conocimiento concreta alcanzando así buenos resultados.

Específicamente, la investigación en la BR-DR, según (Mollá & Vicedo, 2007), está dirigida a problemas relacionados con la incorporación de información específica del dominio en el estado del arte actual de la tecnología de BR con la esperanza de lograr capacidades de razonamiento profundo y ejecuciones con una precisión confiable en aplicaciones del mundo real. Sin embargo, no es nada fácil la tarea de integrar e incorporar esta información a sistemas de BR existentes, debido a la heterogeneidad de su formato de representación. En la actualidad no existe ningún método que permita adaptar automáticamente SBR-DA existentes a nuevos dominios incorporando los recursos de conocimiento disponibles en ese dominio independientemente de su formato.

Nuestro trabajo de tesis se centra en este propósito, en el de la adaptación de SBR-DA a dominios restringidos. Vamos a presentar una serie de aproximaciones para desarrollar un método de adaptación. Entiéndase aquí "método de adaptación" como el recurso que le otorgará a todo SBR-DA la capacidad para enfrentar el cambio, para introducir y extraer información de diferentes dominios restringidos independientemente del idioma, formato y calidad en los datos (referido al ruido que puede presentar la información) de sus recursos de conocimiento. Para ello, nos basaremos en el desarrollo dirigido por modelos o *model-driven*

development (Bézivin, 2005) con el propósito de usar recursos de conocimiento y corpus textuales para adaptar automáticamente y con el menor esfuerzo posible SBR-DA haciéndolos útiles en escenarios de dominios restringidos del mundo real.

En lo que resta de capítulo, hablaremos de las motivaciones que impulsan a esta tesis y de los objetivos planteados, así como la estructura de este trabajo.

1.1. Motivaciones

Los SBR-DA han sido objeto de amplio estudio en la última década, impulsados por diversos foros internacionales en este campo como el TREC, CLEF y NTCIR. En particular estos foros han venido marcando los retos en el desarrollo de la BR desde la perspectiva de RI, además de brindar un marco de evaluación y comparación entre las diversas aproximaciones. Sin embargo, a pesar de que últimamente se le ha dado mayor importancia a los SBR-DR (Mollá & Vicedo, 2007), en la aplicación de la BR en actividades concretas del mundo real, aún no cuentan con un sistema de evaluación estándar como los SBR-DA. La principal causa son las características heterogéneas de los SBR-DR actuales, ya que la mayoría han sido concebidos desde el inicio para un dominio concreto. De esta forma quedan muy ligados a las características y los recursos de conocimiento del dominio de aplicación.

Existe otra forma de desarrollar SBR-DR y es partiendo de un sistema *baseline* de BR-DA y adaptándolo al dominio. No obstante, existen varios factores que determinan las mejores técnicas para usar en la BR-DR y qué técnicas usadas en la BR-DA pueden ser o no efectivas en la BR-DR. En la actualidad persisten dificultades en la adaptación de un SBR-DA a nuevos dominios restringidos, para así convertirlo en un SBR-DR. En los siguientes párrafos enumeraremos estas dificultades, que junto a la ya comentada importancia adquirida por los SBR-DR en aplicaciones reales de BR, han motivado el trabajo desarrollado en esta tesis.

Seguidamente vamos a plantear las dificultades que debe afrontar cualquier método que se proponga actualmente para adaptar un SBR-DA a un dominio restringido, cuya superación ha motivado este trabajo:

- **Adaptación a diferentes dominios.** La colección de documentos en los SBR-DR suele ser más pequeña y de tamaño fijo con respecto a los SBR-DA, por lo que disminuye la precisión de los sistemas que usan métodos de redundancia, ya que la respuesta no suele repetirse en varias oraciones. Pero, por otro lado, aplicar técnicas de Procesamiento del Lenguaje Natural (PLN[9]) hace que los sistemas de BR tengan problemas de portabilidad.
  Se conoce que muchos SBR-DA emplean métodos derivados de técnicas basadas en redundancia, siendo así más independientes del dominio o idioma de aplicación. Una prueba de este hecho, mostrada en (H. Doan-Nguyen and L. Keila, 2004), es que entre los sistemas participantes en el TREC-8, sólo el 27 % de los sistemas encontraron la respuesta concreta (en el caso de haberla) en una sola ocurrencia, sin embargo el 50 % produjeron una respuesta que tenía un número promedio de 7 repeticiones. Las técnicas basadas en redundancia fueron discutidas por primera vez en (Brill *et al.*, 2001). Este trabajo resaltaba que mientras más crece el tamaño del corpus textual es más probable que la respuesta a una pregunta específica se pudiese encontrar a través de métodos intensivos de datos (en inglés, *data-intensive methods*), que no requieren un modelo complejo del lenguaje.
  Sin embargo, los métodos de redundancia tienen un impacto menor en los SBR-DR, especialmente en el caso de los dominios con corpus relativamente pequeños. Estos dominios naturalmente tendrán menos probabilidad de contener la respuesta correcta en más de un documento. Por lo que adquiere importancia el

[9] PLN, o NLP del inglés *Natural Language Processing*, es una subdisciplina de la IA y la rama ingenieril de la Lingüística Computacional (LC: es un campo multidisciplinar de la lingüística y la informática que utiliza la informática para estudiar y tratar el lenguaje humano). El PLN se ocupa de la formulación e investigación de mecanismos eficaces computacionalmente para la comunicación entre personas o entre personas y máquinas por medio de lenguajes naturales.

uso de técnicas sofisticadas de PLN, incluyendo la resolución de inferencias, si es necesaria, para encontrar la respuesta. Notemos sobre este aspecto que si el tamaño del corpus es relativamente pequeño es posible la aplicación de técnicas de PLN complejas (p.e. análisis léxico, sintáctico o semántico) para el corpus completo de manera *offline*. La factibilidad computacional de aplicar estas técnicas en la actualidad queda probada ya que, por ejemplo, es posible analizar gramaticalmente y extraer las entidades nombradas de corpus grandes como los usados en el CLEF o TREC.

Por otro lado, la aplicación de técnicas de PLN en la tarea de BR liga al SBR-DA con el dominio de aplicación dificultando la portabilidad a nuevos dominios. Usualmente los SBR-DA que aplican PLN en sus aproximaciones lo hacen a través de "patrones". Entiéndase por patrones todas las posibles estrategias para detectar relaciones entre los elementos de la pregunta o de las respuestas candidatas (como pueden ser formas lógicas, expresiones regulares, relaciones sintácticas, relaciones de dependencia y otras) y para hacer cumplir restricciones léxicas o semánticas a determinados elementos. Por consiguiente, existe una necesidad de desarrollar y evaluar métodos que faciliten la adaptación de los patrones del SBR-DA a diferentes conjuntos de datos o dominios.

- **Tratamiento del ruido textual.** El ruido es otro problema que provoca consecuencias nefastas en los dominios restringidos debido al tamaño pequeño del corpus y también a la terminología propia del dominio, afectando al resultado de sistemas de RI y BR.

  Uno de los primeros pasos en la adaptación y aplicación de un SBR-DA en un entorno restringido es la necesaria elaboración del propio corpus. El corpus puede estar formado por documentos de texto obtenidos automáticamente desde fuentes heterogéneas, como sitios Web, ficheros PDF (Portable Document Format) o incluso desde herramientas OCR (Optical Character Recognition) o ASR (Automatic Speech Recognition) (Vinciarelli, 2005). Alcanzar una conversión fiel de estas fuentes de datos

a ficheros de texto plano no es una tarea fácil, dado que se puede introducir ruido a partir de errores ortográficos o tipográficos. Por otro lado, si el tamaño del corpus es lo suficientemente grande, como es el caso de los corpus de dominio abierto, la redundancia de información presente ayuda a controlar los efectos del ruido porque un mismo texto puede aparecer con o sin ruido a través del corpus. En cambio, el ruido se convierte en un problema serio en los dominios restringidos donde el corpus es usualmente pequeño y presentan poca o ninguna redundancia. Por consiguiente el ruido dificulta tareas como la BR y RI en dominios restringidos, obteniendo probablemente resultados erróneos.

El ruido también afecta los resultados de la BR cuando los términos en la pregunta pueden escribirse directamente en una manera incorrecta. En los dominios restringidos en concreto, donde la terminología es un punto clave, el tratamiento de esta situación es de gran importancia ya que los términos muy técnicos presentan mayor complejidad en la escritura y por tanto acarrean mayores errores en el momento de efectuar la pregunta.

Aminorar el impacto del ruido en el resultado de la RI y BR es uno de los retos que debe afrontar cualquier SBR-DR, mediante estrategias que no dependan de un conocimiento profundo de los tipos de errores que ocurren.

- **Obtención de taxonomías refinadas de preguntas.** Los tipos de preguntas que se hacen en un dominio restringido son naturalmente diferentes a las que se realizan en dominios abiertos. Serán, mayormente, preguntas técnicas que requieren respuestas específicas ya que los usuarios, especialmente los expertos, pueden usar terminología propia del dominio. Las preguntas realizadas por este tipo de usuarios son habitualmente mucho más complejas que las de un usuario casual de un SBR-DA.

  Tener una taxonomía de preguntas o de Tipos de Respuesta Esperada (TRE) bien definida, ajustada y refinada, para el dominio restringido en cuestión, es de gran importancia para el buen desempeño del proceso de clasificación de la pregunta. La clasificación de la pregunta o reconocimiento del tipo de respuesta

esperada es la tarea que asigna una clase semántica contenida en dicha taxonomía a la pregunta del usuario. Sea cual sea el método (basado en conocimiento o basado en corpus) usado para llevar a cabo esta tarea se necesita una taxonomía de preguntas. Además, mientras más grande, refinada y precisa sea la taxonomía de preguntas, mayor será la precisión del SBR-DR a la hora de localizar la respuesta esperada.

Por consiguiente, es crucial superar los retos que impone la adaptación de las taxonomías de TRE de SBR-DA a nuevos dominios restringidos. Otra motivación de esta tesis, es precisamente proponer una estrategia de adaptación de la taxonomía de TRE que permitan aprovechar el lado positivo de que los corpus de dominios restringidos son relativamente pequeños, haciendo más fácil delimitar los aspectos sobre los que se pueden interrogar. Además dicha estrategia debe ser lo más automática posible para evitar los esfuerzos en tiempo y costo de hacerlo manualmente.

- **Utilización de recursos de conocimiento heterogéneos.** Una diferencia importante entre la BR en Dominio Abierto (BR-DA) y la BR en Dominio Restringido (BR-DR) es la existencia de recursos de conocimiento específicos para dominios restringidos que puedan ser usados. Estos recursos de conocimiento son conocidos como Sistemas de Organización del Conocimiento (SOC), en inglés llamados *Knowledge Organization Systems* (Hodge, 2000). Los SOC incluyen una variedad de esquemas que organizan, manejan y recuperan información; por ejemplo, diccionarios, tesauros, ontologías, etc.

  Intuitivamente, un buen método de BR-DR necesita tener disponible cualquier nivel de información acerca del dominio en el orden de poder manejar las necesidades de información de los usuarios con la especialización y profundidad requeridas. El tipo de información disponible para un dominio en particular está intrísecamente relacionado con la complejidad del dominio y las necesidades particulares de los usuarios del dominio. Por lo tanto, los SOC del dominio pueden extenderse desde simples listas de entidades y términos especializados hasta ontologías de alto nivel.

Sin embargo, estos recursos tienen sus propios formatos e interfaces de acceso, con lo cual la tarea de unificación de los mismos por el sistema de BR se hace extremadamente costosa.
Existe la necesidad de crear una estrategia capaz de emplear e integrar, en la arquitectura del SBR-DA para obtener un SBR-DR, cualquier tipo de SOC disponible en el dominio de forma transparente. Entiéndase por transparente, que cualquier teoría o técnica empleada en el proceso de BR pueda funcionar independientemente del SOC de dominio seleccionado, realzando así la portabilidad del sistema entre dominios.

1.2. Objetivos

Sirviendo como referencias todas las motivaciones y dificultades expuestas en la sección anterior, podemos establecer como principal objetivo de esta investigación: el desarrollo de un método que facilite la adaptación de SBR-DA a dominios restringidos, empleando de manera transparente los recursos de conocimiento del dominio disponibles. Para llevar a cabo este objetivo seguiremos tres premisas fundamentales que respetaremos durante todo el desarrollo de nuestro trabajo:

La adaptación de SBR-DA a dominios restringidos debe superar problemas de portabilidad, reusabilidad e integración. Desde hace algún tiempo las investigaciones en ingeniería de software se han enfocado en el desarrollo de técnicas para superar problemas de portabilidad, reusabilidad e integración en el desarrollo de software. Como un SBR-DA es un sistema de software, nos planteamos un método basado en técnicas de ingeniería de software para cumplir esta premisa, concretamente en el desarrollo dirigido por modelos. Nuestro propósito es la creación de modelos que representen todos los elementos del dominio útiles para el proceso de BR, por ejemplo, los términos relevantes del corpus y sus correspondientes conceptos en los SOC disponibles, y por otro lado la adquisición automática de un modelo de los patrones del SBR-DA de partida. Luego estos modelos se utilizan para crear de manera automática nuevos modelos que reflejen los patrones

adaptados al dominio creando así un SBR-DR, reduciendo a su mínima expresión la labor manual del desarrollador del sistema.

Los SBR-DR deben ser tolerantes a la presencia de ruido en sus datos. Los SBR-DR se ven más afectados por el ruido en el corpus que los SBR-DA debido al tamaño de sus corpus, lo que hace que haya poca redundancia de información. Para cumplir esta premisa proponemos una estrategia de tolerancia a fallos provocados por el ruido textual, que se base en la utilización de una distancia de edición extendida y de un SOC que sirva como vocabulario controlado e intermedio entre las palabras con ruido y las originales. Esta propuesta es independiente del tipo de ruido presente y del tipo de sistema RI o BR que la utilice.

Los SBR-DR necesitan de taxonomías refinadas de TRE para aumentar la precisión de sus respuestas. Debemos aprovechar el tamaño pequeño de los corpus de dominio restringido para delimitar los tipos de preguntas que se pueden realizar. Además tenemos que ser capaces de utilizar los diferentes SOC disponibles, integrándolos independientemente del formato, para la creación de una taxonomía más refinada que permita responder a preguntas más técnicas, complejas y específicas al dominio.

Para alcanzar el objetivo principal anteriormente expuesto, se puede desglosar en objetivos específicos que seguidamente se redactan:

- Desarrollar un método de adaptación de SBR-DA a diferentes dominios restringidos que explote al máximo los recursos de conocimiento disponibles independientemente del esquema de representación que sigan.
- Desarrollar una estrategia de tolerancia al ruido textual, que permita la mejora del proceso de RI sobre corpus pequeños y ruidosos.
- Adaptar de forma automática la taxonomía de TRE que utiliza un SBR-DA a la terminología y características propias de cualquier dominio restringido, incrementando su nivel de granularidad.
- Obtener de forma automática los patrones que utiliza el SBR-DA para realizar la BR.

- Generar de forma automática los nuevos patrones que permitan la obtención de un SBR-DR a partir de la adaptación de un SBR-DA, necesitando sólo una mínima supervisión por parte del desarrollador del sistema.
- Evaluar todas las propuestas empleando un caso de estudio representativo de los dominios restringidos, como es el dominio agrícola.

Otros objetivos secundarios, y por ello no menos importantes, derivados de las motivaciones planteadas previamente son:

- Estudiar las características principales de los SBR-DA, reconociendo la arquitectura estándar de estos sistemas y los problemas de adaptación a otros dominios que impiden su ágil aplicación a entornos más reales y restringidos.
- Establecer el estado de la cuestión de los SBR-DR, destacando la importancia y desventajas de la utilización de recursos del conocimiento.
- Enumerar otras estrategias desarrolladas para llevar a cabo los procesos principales que conforman un sistema BR y que están directamente involucrados en los problemas de portabilidad de los mismos.
- Estudiar las principales aproximaciones para tratar con el ruido textual, dentro del marco de aplicaciones de PLN.
- Desarrollar corpus de preguntas para la evaluación de la propuesta de adaptación con el apoyo de expertos en un determinado dominio, en nuestro caso en el dominio agrícola.

1.3. Estructura de la tesis

La memoria de este trabajo de tesis doctoral está organizada en siete grandes partes (introducción, estado actual del tema, adaptación de SBR-DA a dominios restringidos, evaluación aplicada al dominio agrícola, conclusiones, bibliografía y anexos) que se componen de un total de ocho capítulos y tres anexos. Vamos a describir a continuación cada uno de los capítulos restantes, describiendo brevemente su contenido:

- En el capítulo 2 se introduce el origen, la arquitectura y las principales características de los sistemas de BR. A su vez, se profundiza en la BR-DA resaltando sus características y realizando un estudio de las propuestas más interesantes en la actualidad. Para concluir resaltamos los objetivos principales y problemas actuales que presentan estos sistemas y que nuestra investigación persigue.
- En el capítulo 3 de la memoria de la tesis doctoral, presentaremos las propuestas actuales para enfrentar algunos de los problemas que existen en la adaptación de sistemas de BR-DA a nuevos dominios restringidos. Específicamente, tres problemas principales: (i) tratamiento del ruido textual en sistemas de RI para dominios restringidos, (ii) generación de taxonomías de TRE ajustadas al dominio restringido, y (iii) adaptación de los patrones de preguntas y respuestas al dominio restringido.
- En el capítulo 4 mostraremos el trabajo realizado para hacer que cualquier sistema de RI sea tolerante al ruido textual. Para ello empleamos un algoritmo de distancia de edición extendida y cualquier SOC disponible en el dominio. De esta forma evitamos la necesidad de analizar los tipos de errores que presenta el corpus, y hacemos que la propuesta sea independiente del sistema de RI y el SOC que se utilice. También explicamos la necesidad de que los sistemas de RI y BR sean tolerantes al ruido en los dominios restringidos por las características de este tipo de corpus.
- En el capítulo 5 exponemos nuestro método de adaptación de un sistema de BR-DA a dominios restringidos. El mismo tiene su base en el desarrollo dirigido por modelos, por lo que hacemos una breve descripción de esta técnica. Luego, se describen en detalle los metamodelos creados para desarrollar los modelos necesarios para llevar a cabo los pasos de nuestra aproximación. Además se detallan los pasos del método de adaptación, por ejemplo, la obtención de los modelos, de la taxonomía de TRE para el dominio, y de los patrones de preguntas y respuestas adaptados al dominio.

- En el capítulo 6 se introduce la implementación de una herramienta que da soporte a nuestra propuesta: MARAQA (***M**odel-driven **A**daptation for **R**estricted-dom**A**in **Q**uestion **A**nswering*).
- En el capítulo 7 describimos los experimentos realizados para evaluar nuestra propuesta, una vez introducidos los recursos utilizados para realizar dichos experimentos. En primer lugar, se realizan una serie de experimentos preliminares que servirán de marco de comparación y de evaluación al resto de experimentos con el fin de comprobar la efectividad de nuestra aproximación. Posteriormente, se describen los experimentos realizados para mostrar la idoneidad de nuestra aproximación para la recuperación de la información correcta desde un corpus ruidoso de dominio agrícola. Se realizan dos experimentos: (i) el primero para determinar los valores mínimos y máximos del umbral en el cual deben encontrarse los resultados de nuestra propuesta para ser válida y (ii) el segundo tiene como finalidad mostrar los resultados y utilidad de nuestra aproximación. Además se describen previamente la preparación de los experimentos y los recursos requeridos con el propósito de hacer posible su reproducción. Finalmente, se describen los experimentos realizados para mostrar la utilidad de nuestro método de adaptación de sistemas de BR a dominios restringidos, empleando como caso de estudio un dominio agrícola. Específicamente se realizaron dos experimentos en función de comprobar nuestra aproximación para: (i) generar una taxonomía de TRE ajustada al dominio restringido, y (ii) adaptar los patrones de pregunta y respuesta de un sistema de BR-DA a un dominio restringido.
- Por último, en el capítulo 8 mostraremos las conclusiones, aportaciones y trabajos futuros, así como la producción científica derivada de esta investigación.

Parte II

Estado de la cuestión

Capítulo 2

Sistemas de Búsqueda de Respuestas

En este capítulo se expone el origen y resumen de la historia de los sistemas de BR. Se presenta el problema de la BR en dominios restringidos, así como la necesidad e importancia de sistemas que resuelvan la tarea de BR en entornos restringidos. Además se presenta una descripción del estado actual de los sistemas de BR de dominio restringido con el objetivo de lograr una mejor comprensión de los mismos. Se realiza una revisión detallada de las aproximaciones más importantes desarrolladas en la actualidad, resaltando sus resultados y las técnicas que emplean.

2.1. Origen y perspectiva histórica de los sistemas de BR

Hace años está presente el interés por la investigación en sistemas de BR, desde la perspectiva de la Inteligencia Artificial[1] (IA), centrada fundamentalmente en el desarrollo de sistemas que respondieran a preguntas realizadas sobre una base de conocimiento estructurado. Obviamente, estos sistemas tenían como desventaja que sólo proveían respuestas concernientes a la información previamente codificada en la Base de Datos (BD). Por otro lado, el beneficio de esta aproximación es que tenían un modelo conceptual del dominio de aplicación representado en la estructura de la BD permitiendo el uso de técnicas avanzadas como demos-

[1] Inteligencia Artificial: es la rama de la ciencia de la computación que se ocupa de la automatización de la conducta y del pensamiento inteligente.

tración de teoremas y razonamiento profundo para ocuparse de necesidades complejas de información.

Entre los sistemas más importantes que enmarcan los comienzos, por los años sesenta, se encuentran las Interfaces en Lenguaje Natural de acceso a Bases de Datos (ILNBD). Las ILNDB se pueden describir como sistemas que tienen como fuentes de conocimiento a BD's que contienen la información relevante acerca de un tópico. En esos sistemas la pregunta del usuario en lenguaje natural se convierte en una consulta a la BD, a través de un lenguaje formal[2] o de interrogación, y la salida de la BD se toma como respuesta. A continuación se muestran algunos ejemplos:

BASEBALL (Green *et al.*, 1961): contestaba preguntas sobre jugadores del deporte béisbol en Estados Unidos en cada temporada, a partir de información almacenada en una BD. El sistema era capaz de procesar la pregunta sintáctica y semánticamente. Tuvo una efectividad notable en su dominio de aplicación.

LUNAR (Woods, 1973): permitía interrogar, en lenguaje natural, a una BD sobre los análisis geológicos realizados a muestras de materiales recogidos en misiones de exploración espacial (p.e., misión Apolo para llegar a la superficie de la Luna). Obtuvo buenos resultados, alcanzando un 90 % de aciertos a las preguntas realizadas por los usuarios del dominio en cuestión.

USL (Sopeña, 1983): constituye una interfaz interactiva con un sistema de gestión de BD relacionales, su objetivo era traducir las frases de entrada escritas en lenguaje natural a sentencias del lenguaje formal de interrogación de la BD.

Además, se puede consultar el trabajo sobre la interpretación de la comparación en consultas a una BD geográfica a través de la lógica en (Moreno *et al.*, 1993) y en (Sobrino, 2004) sobre la interrogación, en lenguaje natural, de una base de datos lógica.

Otros trabajos importantes que se deben consultar para poder comprender el desarrollo que tuvieron los sistemas que dieron paso

[2] Lenguaje formal: son los lenguajes "artificiales" que el hombre ha desarrollado para expresar las situaciones específicas que se dan en cada área del conocimiento científico (p.e. matemática, lógica e informática). Es un lenguaje cuyos símbolos primitivos (i.e. alfabeto) y reglas (i.e. gramática formal o sintaxis) para unir esos símbolos están formalmente especificados. Algunos ejemplos son: el lenguaje de la lógica matemática o los lenguajes de programación.

a la línea de investigación de BR, son los sistemas de diálogo basados fundamentalmente en plantillas o patrones. Seguidamente se citan algunos ejemplos destacados:

ELIZA (Weizenbaum, 1966): llevaba a cabo un diálogo con el usuario, en inglés, imitando la conversación con un psicólogo. El sistema era capaz de conversar sobre cualquier tema usando reglas muy simples que reconocían palabras claves en la entrada y daba respuestas apropiadas que conservaba en forma de plantillas o *templates*. Por ejemplo, si el usuario escribía, "*You are X*", ELIZA respondía, "*What makes you think I am X*", donde X es cualquier adjetivo. Era un sistema muy rudimentario para responder preguntas, pero impulsó la investigación en sistemas de diálogo (p.e., los programas que participaron en los premios anuales "Loebner Prize"[3]).

SIR (Raphael, 1964): es otro de los programas clásicos que usaron plantillas. Alcanzó un razonamiento lógico que le permitió responder preguntas como: "*Every person has two hands. Every hand has five fingers. Joe is a person. How many fingers does Joe have?*".

STUDENT (Bobrow, 1964): daba solución a problemas de matemáticas planteados en inglés para el pre-universitario. Una notable mejora que implementaba era que usaba patrones recursivos. Por ejemplo, el patrón "*if X then Y*" permitía oraciones completas en los lugares "X" y "Y", y podía aplicar patrones de oraciones completas a ellas. En (Norvig, 1992) está la re-implementación del sistema en el lenguaje Common Lisp.

SHRDLU (Winograd, 1972): simulaba la operación de un robot en un mundo virtual (i.e. un mundo pequeño de objetos, específicamente de bloques), donde se le podían realizar preguntas acerca del estado de ese mundo. SHRDLU[4] se desarrollaba, por tanto, en un dominio bien específico y simple, representado a través de reglas físicas fácilmente codificadas en el programa.

SAPLEN (López-Cózar & Rubio, 1997): es un sistema automático de pedidos en lenguaje natural, que intenta simular el

[3] `http://www.loebner.net/Prizef/loebner-prize.html`
[4] `http://hci.stanford.edu/winograd/shrdlu/`

diálogo llevado a cabo por el dependiente de un restaurante de comida rápida a la hora de atender las peticiones de los clientes.

Hasta el momento hemos analizado trabajos que contribuyeron al nacimiento de los sistemas de BR, desde la perspectiva de IA. Los sistemas analizados poseían una funcionalidad común, tenían un núcleo basado en BD's de conocimiento escritas por expertos. Aplicaban fundamentalmente herramientas de Procesamiento del Lenguaje Natural (PLN) en combinación con técnicas de IA tales como demostración de teoremas para la extracción de respuestas de la base de conocimientos y para el emparejamiento de los patrones. El trabajo de Levine muestra con detalle este tipo de aproximaciones (Levine & Fedder, 1989).

Desde finales de los años 80, las investigaciones en BR han derivado hacia el tratamiento de bases de conocimiento no estructuradas, gracias al desarrollo de las teorías de comprensión, específicamente comprensión de texto, en Lingüística Computacional (LC). Estos sistemas iniciales alcanzaron resultados bastante satisfactorios en el caso particular del tratamiento de documentos de dominios muy restringidos. Algunos ejemplos de proyectos diseñados en esta dirección fueron:

Unix Consultant (Wilensky *et al.*, 1988): implementa un sistema de ayuda en lenguaje natural sobre el sistema operativo Unix. El sistema procesa las preguntas del usuario, soporta inferencia y genera la respuesta de acuerdo a la información de Unix que mantiene y al tipo de usuario. Para ello, contaba con una base de conocimiento elaborada a mano que comprendía el dominio. El sistema como investigación ayudó a los investigadores a pesar de que no superó la fase de demostraciones simples.

LILOG (Herzog & Rollinger, 1991): es un sistema de comprensión de textos que incluye herramientas de análisis del lenguaje natural, interpretación semántica, inferencia y generación de lenguaje natural. El sistema operaba en el dominio turístico permitiendo consultas sobre información de una ciudad alemana en el idioma alemán.

En la actualidad, y desde la inclusión de la tarea de BR en el TREC–8 (Voorhees, 1999) en 1999, muchas conferencias como el propio TREC, CLEF y NTCIR han impulsado un enorme avance

en los sistemas de BR, ya que funcionan como marco de evaluación para las investigaciones que se realizan y de comparación entre los sistemas similares que se desarrollan por los investigadores. En este marco se ha desarrollado la BR desde la perspectiva de RI. Desde esta perspectiva, la BR se enfoca en encontrar extractos de texto que contengan la respuesta concreta, a preguntas formuladas por los usuarios en lenguaje natural, dentro de grandes volúmenes de documentos. Los conjuntos de prueba en las tareas de estas conferencias se han amoldado a tipos específicos de preguntas-respuestas que son fácilmente evaluados y que se enfocan en el uso de métodos rápidos y poco profundos que son generalmente independientes del dominio de aplicación. En otras palabras, la investigación actual se enfoca en la Búsqueda de Respuestas en Dominios Abiertos (BR-DA) basada en textos. En la siguiente sección profundizaremos en las principales características de los Sistemas de BR-DA (SBR-DA) a través del estudio de algunos ejemplos presentados en estas conferencias.

2.2. Sistemas de BR en Dominios Abiertos

Desde el año 1999 la investigación en SBR-DA ha ido en crecimiento, por lo que existen en la actualidad diferentes estudios en los cuales se han clasificado y agrupado las diferentes aproximaciones de SBR-DA existentes según diferentes criterios.

En (Moldovan *et al.*, 2003) se hace un análisis profundo del estado de la cuestión de los SBR-DA, con la finalidad de examinar el desempeño de cada módulo en un sistema baseline en serie y el impacto de varios recursos léxicos. Se llega a la conclusión de que el rendimiento global depende de la profundidad de los recursos de PLN y de las herramientas usadas para encontrar la respuestas. El sistema que se evalúa se colocó en los primeros puestos en las competiciones del TREC entre 1999 y 2001.

En este trabajo también se propone un taxonomía de clasificación de SBR-DA a través de 5 clases en función de: los recursos lingüísticos y de conocimiento empleados, complejidad de PLN, procesamiento de documentos, métodos de razonamiento,

si maneja que esté o no la respuesta explícitamente indicada en el documento; o si es capaz de fusionar las respuestas en caso de ser necesario. Entonces, los tipos de preguntas se dividirían según la capacidad que necesite el SBR-DA para darle respuesta de: (i) procesar preguntas factuales, (ii) permitir mecanismos simples de razonamiento, (iii) fusionar la respuesta desde diferentes documentos, (iv) interactividad y (v) razonamiento analógico.

En las conferencias TREC (p.e. TREC-8, TREC-9 y TREC 2010) la distribución promedio de las preguntas según estas categorías es del 67,5 % factuales y 27,9 % razonamientos simples, 1,7 % fusión-lista y 2,9 % interactivas-contextuales. El sistema de Moldovan se divide en 10 módulos secuenciales: 5 se corresponden al procesamiento de la pregunta, dos al procesamiento de pasajes y documentos y los últimos 3 al procesamiento de la respuesta. Para detectar el Tipo de Respuesta Esperada (TRE) utiliza una jerarquía basada en WordNet. En el procesamiento de la pregunta capturan los conceptos en la pregunta y las dependencias binarias entre los conceptos. La búsqueda de las respuestas dentro de los pasajes recuperados se restringe a aquellas candidatas que se corresponden con el TRE, usando para ello el emparejamiento de un conjunto de patrones de respuesta en los pasajes.

Moldovan realizó un análisis de los errores cometidos en cada módulo de su propuesta. De ese estudio se detectó que el 36.4 % de los errores estaban en el proceso de derivar el TRE, siendo este el módulo con mayor dificultad. El fallo en este módulo hace que sea difícil o imposible la actuación de los módulos siguientes en sus respectivas tareas. Se pueden dar dos casos para el fallo del módulo: uno que el conjunto de respuestas candidatas identificadas en los pasajes recuperados esté vacío porque el TRE es desconocido (28,2 %) o que contenga entidades erróneas cuando el TRE es incorrecto (8,2 %). Otro módulo que tuvo un error del 25,7 % fue el de la expansión de las palabras claves para la RI. En muchas ocasiones los pasajes relevantes se pueden perder si no se logran expandir las palabras claves usadas en la RI con las formas semánticamente relacionadas que aparecen en las respuestas. Entre el módulo anterior y su predecesor, correspondiente al módulo de selección de las palabras claves (incorrectamente adicionadas

o excluidas) con un 8,9 % de errores, suman los dos un 34,6 % de errores del total.

En el módulo de procesamiento de la respuesta ocurre lo siguiente: si el TRE es correctamente detectado, la identificación de las respuestas candidatas produce un 8,0 % de error. El 3,1 % de este error se debe al reconocimiento de las entidades nombradas (por el empleo de diccionarios incompletos) y el 4,9 % se debe a un falso emparejamiento con los patrones de respuesta. Aplicaron un experimento deshabilitando el uso de WordNet y la MRR[5] cayó en un 59 % con respecto a la precisión del sistema con todas las herramientas habilitadas. Los dos módulos que están más influenciados por WordNet son la obtención del TRE y la expansión de las palabras claves. Por ejemplo, para la pregunta "*What was the name of the US helicopter pilot shot down over North Korea*" la raíz ambigua de la pregunta *What* no brinda ninguna pista del TRE y si se usa WordNet se obtendría que *pilot* es una especialización de aviador que a su vez lo es de persona, entonces el TRE sería persona. Específicamente, las preguntas de tipo *What*, cuando WordNet fue deshabilitado, disminuyeron la precisión del sistema (MRR) en 37 % con respecto al 59 % del conjunto de preguntas completos. Este resultado indica que la disponibilidad de información léxico-semántica es mucho más importante para las preguntas difíciles. Al deshabilitar el reconocedor de EN el procesamiento de la respuesta carecía de la información semántica necesaria para identificar la respuesta candidata. Se valoraron las respuestas candidatas basándose estrictamente en el emparejamiento de las palabras claves, la precisión disminuyó un 32 % con respecto al sistema de partida. Se concluía en este trabajo que los sistemas de BR tienen mejor rendimiento cuando los pasajes relevantes y las respuestas candidatas son claramente definidos en las preguntas (el problema es la ausencia de esquemas o algoritmos potentes para modelar preguntas complejas en función de derivar la mayor cantidad de información posible y mejorar así los criterios de búsqueda a través de la colección de documentos).

[5] Mean Reciprocal Rank (MRR) es definido con $MRR = \frac{1}{Q}\sum_{i=1}^{Q}\frac{1}{rank_i}$, donde Q es el número de preguntas y $rank_i$ es la posición del primer pasaje relevante recuperado para la pregunta i.

Los sistemas de BR-DA actualmente en operación, afrontan la tarea de BR desde la perspectiva del usuario casual: un usuario que realiza preguntas simples que requieren un hecho, situación o dato concreto como respuesta. Los sistemas utilizan un único tipo de fuente de información en la que se realiza la búsqueda de respuestas: una base de datos textual compuesta por documentos escritos en un único lenguaje. En algunos casos se ha avanzado un poco más, mediante el uso de bases de datos léxico–semánticas (principalmente WordNet) y la integración de algún tipo particular de ontología como SENSUS (Hovy, 2000). Desde esta perspectiva, los sistemas existentes pueden contestar a preguntas simples cuya respuesta aparece en un único documento y además, los conceptos expresados en la pregunta están localizados en zonas del texto cercanas a dicha respuesta.

Entre los trabajos de BR-DA más destacados en el idioma español se encuentra el sistema de la Universidad de Alicante (Vicedo, 2004; Roger *et al.*, 2005a; Ferrández *et al.*, 2006a; Roger *et al.*, 2008) que participó en la tarea monolingüe en español del CLEF[6]. Así mismo, el sistema MIRACLE (De Pablo, 2004) presentado por algunas Universidades y empresas españolas de Madrid, explora el uso de modelos ocultos de Markov para la extracción de la respuesta, coleccionando los datos empleados en la fase de entrenamiento mediante Google. También destacan otros trabajos que realizan análisis semántico (Ferrés, 2004), o simplemente tienen en cuenta las palabras claves (*keywords*) (Méndez-Díaz, 2004), o no realizan prácticamente ningún PLN sino que se basan en el Aprendizaje Automático y las probabilidades (Pérez-Coutiño, 2004).

2.3. Sistemas de BR en Dominios Restringidos

Un sistema de BR de dominio restringido es un sistema de BR diseñado para adaptarse de forma óptima a un área concreta. Dentro de estos sistemas pueden tratarse diferentes tipos de preguntas, siendo el ámbito de aplicabilidad el que determine qué tipo

[6] CLEF: Cross-Language Evaluation Forum, http://clef-campaign.org/

de preguntas resultan más interesantes. En la sección 2.3.2 se realiza un estudio de las principales aproximaciones en la actualidad, resaltando sus características, ventajas y desventajas.

A continuación, citaremos las características más importantes de un sistema de BR basado en un dominio restringido. Según (Ferrés & Rodríguez, 2006) entre las características de un sistema de BR que trabaja en un entorno concreto se encuentran:

1. La colección de documentos para un sistema de BR de dominio restringido es más pequeña y específica. Por lo que resulta vital que la precisión en la recuperación de los pasajes sea elevada y no deben aplicarse métodos de redundancia, ya que la respuesta no suele repetirse en varias oraciones.
2. El repertorio de los patrones de preguntas suele ser pequeño, ya que estos sistemas están orientados a tareas específicas. De esta manera se favorece la precisión de los resultados del módulo de análisis de la pregunta.
3. La terminología es un punto clave para los dominios restringidos, ya que los términos muy técnicos acarrean una mayor complejidad en el analizador léxico-sintáctico por el sublenguaje del dominio. En (Rinaldi *et al.*, 2004; Spasic, 2003) se pueden consultar diferentes aproximaciones para la extracción de la terminología propia de un dominio.
4. La estructura inicial de los documentos en un entorno determinado permite, muchas veces, identificar una estructura simbólica y regular, lo que permitiría aislar determinadas partes del documento dándole un determinado peso (mayor o menor) a cada palabra en dependencia del segmento donde se encuentren (Rinaldi, 2002).
5. Es preferible que el sistema no responda antes que brindar una respuesta incorrecta, ya que la calidad de las respuestas resulta imprescindible (Chung *et al.*, 2004a).

Existen dos aproximaciones para desarrollar un sistema de BR en un dominio restringido, una es partiendo de un sistema inicial o *baseline* –como puede ser un sistema de BR para dominio abierto– y la otra es partiendo de cero. Cuando se sigue la primera aproximación se presentan problemas en la adaptación, fundamental-

mente por las características ya mencionadas de los sistemas de BR de dominio restringido, por lo que basaremos nuestra investigación en este aspecto. Ilustramos algunos de los problemas que surgen en el proceso de creación, y más adelante en nuestra propuesta se verá cómo enfrentaremos muchos de estos problemas.

- Son sistemas que se ponen en práctica en algún área específica del mundo real, por lo que el usuario podrá realizar preguntas en formatos y estilos diferentes que exigirán una respuesta correcta y hasta una cierta aclaración en determinados casos si el usuario lo solicita. Este aspecto es contrario a lo que sucede en las grandes competiciones que tienen tareas definidas para evaluar sistemas de BR, tales como el CLEF y TREC que trabajan con una clasificación bien definida de las preguntas.
- Se hace necesaria la elaboración de los propios recursos que requiere el sistema (corpus y colección de preguntas), por el contrario el resto de los sistemas de BR adquieren estos recursos gracias a los certámenes.
- No se encuentra la respuesta correcta en muchos documentos debido al tamaño fijo y restringido de la colección de documentos. Al haber menor redundancia de las respuestas en el corpus, la precisión tiende a disminuir. Una prueba de eso es que entre los sistemas participantes en el TREC-8, sólo el 27 % de los sistemas encontraron la respuesta concreta en el caso de que hubiese una sola ocurrencia, sin embargo el 50 % produjeron una respuesta con un número promedio de 7 repeticiones (H. Doan-Nguyen and L. Keila, 2004).
- Otra dificultad es la necesidad de trabajar con porcentajes elevados de precisión que superen el obtenido por el sistema *baseline*, para que sea viable la utilización del sistema en el mundo real.

2.3.1. Importancia y actualidad

En la actualidad la efectividad de los sistemas de BR es relativamente baja, por lo que aún queda mucho trabajo por hacer. Existen dos factores que corroboran la necesidad e importancia

de mejorar la precisión de los sistemas de BR en entornos restringidos:

1. La necesidad inminente de utilizar en la vida real los sistemas de BR especializándolos para dar respuestas a consultas del contexto específico donde se utilice.
2. La diversidad de formato y estilo con que un usuario casual o experto puede realizar una consulta en un entorno restringido y conocido.
3. La complejidad de los lenguajes técnico-científicos en que están escritos muchos documentos y la terminología propia del dominio dificultan la búsqueda de información para dar respuesta a la consulta de los usuarios.

En este contexto adquieren relevancia los sistemas que sean capaces de localizar y devolver información sobre un entorno restringido adaptándose al mismo de forma óptima; tomando así niveles elevados de importancia, en la búsqueda de información, los sistemas de BR en dominios restringidos.

Por otra parte, y con la idea de reafirmar la necesidad de realizar investigaciones en esta línea, cabe destacar la ausencia de modelos para lograr la completa y correcta adaptación al entorno de un sistema de BR que alcance niveles altos de precisión en la respuesta y que permitan su uso en el mundo real.

2.3.2. Estado actual de los sistemas de BR en dominios restringidos

Después de describir lo que fue el nacimiento de los sistemas de BR y las características más significativas de la BR en dominios restringidos, analizaremos las aproximaciones más importantes obtenidas en la actualidad, agrupándolas según la tendencia empleada a la hora de elaborar un sistema de este tipo. Aclárese que de ahora en adelante se utilizará BR-DR para referirnos a la Búsqueda de Respuesta en Dominios Restringidos. Primero que todo aclarar que existen diferentes tendencias a la hora de desarrollar un sistema de BR, las cuales agruparemos a partir de una

clasificación que definiremos según los enfoques que siguen en sus propuestas.

A continuación se presenta una ligera introducción a los sistemas de BR-DR que se presentarán más adelante en esta sección, los mismos se encierran en ámbitos médico, turístico, de pronósticos del tiempo o académico, entre otros. Algunos de estos sistemas participaron en la competición del CLEF y otros en el taller[7] sobre BR-DR en el marco del congreso de 2004 de la ACL (*The Association for Computational Linguistics*).

Extrans (Hess *et al.*, 2002). Es un sistema diseñado para dar respuesta a preguntas arbitrarias sobre archivos de documentación de UNIX. Se puede decir que el sistema utiliza herramientas complejas de PLN para el análisis de preguntas y documentos, aplica técnicas de inferencia y modelos de demostración de teoremas para la extracción de las respuestas, es de dominio técnico basado en terminología, y su origen fue el sistema *baseline Aircraft Maintance Manuals*[8] (AMM) del Airbus A320. La demo[9] del sistema se encuentra *on-line*, brindando una primera impresión del funcionamiento de Extrans.

WEBCOOP (Benamara, 2004). El sistema interactúa con un dominio turístico parcialmente restringido ya que incluye determinados aspectos de historia, seguridad, salud, inmigración y ecología. Aplica procedimientos de razonamiento avanzados para generar respuestas cooperativas a preguntas realizadas por el usuario en lenguaje natural (LN). Además emplea las formas lógicas y una ontología como recursos para almacenar el conocimiento que extrae desde páginas Web.

SBR-DR para pronósticos del tiempo (Chung *et al.*, 2004a). El sistema es un robot doméstico que contesta preguntas sobre el pronóstico del tiempo. Basa su funcionamiento en el uso de plantillas para representar el conocimiento y usa reglas de inferencia para extraer dicho conocimiento de páginas webs. Para reconocer los eventos del tiempo consta de una ontología propia del dominio.

[7] `http://acl.ldc.upenn.edu/acl2004/qarestricteddomain/index.html`
[8] http://www.srtechnics.com
[9] http://www.ifi.unizh.ch/cl/extrans

SBR-DR para Bell Canada (Doan-Nguyen & K., 2005; Doan-Nguyen & Keila, 2004). El objetivo del sistema es contestar a las preguntas de los clientes sobre los servicios, tanto a particulares como a empresas, que ofrece la compañía Bell Canada. El sistema se basa en un diccionario que incorpora la terminología técnica y en una caracterización de los documentos por conceptos, a partir del conocimiento que extrae un sistema baseline de las páginas web, disponibles en el sitio web de Bell Canada desde el 2003[10]. Es un sistema asistido por el criterio humano en el chequeo de las respuestas devueltas por el sistema.

SBR-DR para Nobel Prizes&LT-World (Frank *et al.*, 2005). Es una aproximación a la BR en dominios restringidos como los Premios Nobel y las Tecnologías del Lenguaje Humano. Parte de bases de conocimiento estructuradas, construyendo un análisis semántico completo de la pregunta, con una interfaz modular entre representaciones semánticas conceptuales y ontologías de dominio específico y bases de datos. Presenta una interfaz flexible para varios tipos de dispositivos de almacenamiento del conocimiento y sus lenguajes de consulta correspondientes.

SBR-DR para entorno médico (Terol *et al.*, 2006). Es un sistema de BR creado desde el inicio para un dominio médico. Está basado en técnicas de PLN y utiliza el Metatesauro de UMLS (Unified Medical Language System) como fuente de conocimiento. La principal técnica de PLN utilizada consiste en el tratamiento computacional de la forma lógica de la pregunta y en el emparejamiento de patrones.

2.3.3. Clasificación para los sistemas de BR-DR

En esta sección agrupamos las diferentes aproximaciones de SBR-DR citadas hasta el momento según una clasificación por: los niveles de PLN que empleen, los recursos de conocimiento que utilicen y el origen del sistema (creado directamente para el dominio restringido o a partir de un SBR-DA). El objetivo que perseguíamos era resaltar las similitudes y diferencias entre las diferentes propuestas; ya que los SBR-DR, a diferencia de los

[10] `www.bell.ca`

SBR-DA, no tienen ningún marco común de evaluación. Este hecho se debe a que son muy heterogéneos, tanto en su arquitectura como en el dominio de aplicación. Primeramente, describiremos cada una de las clasificaciones en que fueron agrupados y luego se resume en el Cuadro 2.1.

Enfoques según los niveles de PLN que utilizan Como se decía anteriormente en este trabajo el Procesamiento del Lenguaje Natural (PLN) es una de las ramas principales de la Inteligencia Artificial, y estudia una propiedad importante de la inteligencia humana: su capacidad de comunicarse por medio del lenguaje. Por tanto, el término PLN hace referencia a la investigación de mecanismos eficaces computacionalmente para la comunicación entre personas y máquinas por medio de lenguajes naturales. Una de las tendencias y la más usada por los sistemas de BR actuales es la que trabaja con técnicas avanzadas de lenguaje natural, un ejemplo de ello son los trabajos de (Roger, 2005; Méndez-Díaz, 2004). Todos los sistemas de BR-DR que se mencionaron anteriormente están enmarcados dentro de esta clasificación (consultar el Cuadro 2.1). Muchas investigaciones se han decidido por esta metodología usando:

- Etiquetadores morfológicos (Acebo *et al.*, 1994a).
- Lematizadores y etiquetadores de entidades (Ferrés, 2004).
- Herramientas de nivel sintáctico (Roger, 2005).
- Técnicas complejas de análisis semántico (Neumann, 2004) y contextual (Jijkoun, 2004).

Otra tendencia dentro de esta clasificación serían los sistemas opuestos, es decir, los que no usan técnicas de PLN, sino técnicas basadas en:

- Aprendizaje automático (Solorio, 2004).
- Modelos estadísticos como los Modelos Ocultos de Markov o Máxima Entropía (De Pablo, 2004; Niu, 2004).

Vale destacar que junto a cada una de las metodologías que hemos citado, colocamos referencias a trabajos de investigación de gran significado en su utilización, la mayoría son SBR-DA.

Enfoques según los recursos que emplean La aproximación de utilizar recursos que permitan adquirir, almacenar y manipular el conocimiento específico del dominio es también muy utilizada por los sistemas actuales. Estas técnicas principalmente se emplean en las fases de análisis de las preguntas y extracción de las respuestas, y pueden estar basadas en la manipulación de fuentes de conocimiento como:

- Formas lógicas (Benamara, 2004; Terol, 2005).
- Bases de datos (Frank *et al.*, 2005).
- Diccionarios o tesauros (Doan-Nguyen & K., 2005).
- Terminología propia del dominio (Rinaldi, 2002; Mollá *et al.*, 2003; Rinaldi *et al.*, 2004).
- Taxonomías (Terol *et al.*, 2006).
- Plantillas (Chung *et al.*, 2004b).
- Ontologías de propósito general (WordNet, SUMO- *Suggested Upper Merged Ontology*) u otras construidas a propósito para el sistema. Es el recurso más utilizado, prueba de ello son los sistemas que se verán en la sección 2.3.4, aunque cada uno lo utiliza con diferentes propósitos.

Enfoques según el origen del sistema El origen de un sistema de BR-DR puede ser:

- Partiendo de un sistema de BR-DA *baseline*. Ejemplos que siguen esta alternativa son: (Mollá *et al.*, 2003; Doan-Nguyen & K., 2005; Frank *et al.*, 2005).
- Creado desde el origen aplicado a un dominio restringido. Por ejemplo en (Benamara, 2004; Chung *et al.*, 2004b; Terol *et al.*, 2006).

2.3.4. Descripción de sistemas de BR de dominio restringido

A continuación se detallan las características principales y el diseño de los SBR-DR referenciados anteriormente. Como el objetivo fundamental del análisis es ver como alcanzan la adaptación al entorno restringido, resaltaremos de cada uno como se refleja

Sistemas BR-DR	Enfoques de las Propuestas		
	PLN	Fuentes de Conocimiento	Origen
Extrans	Análisis Sintático, Análisis Semántico	Formas Lógicas Mínimas, Basado en terminología	Sistema *baseline* AMM (Airbus A320)
WEBCOOP	Técnicas de Razonamiento Avanzado	Formas Lógicas, Ontologías, Plantillas	Creado desde el origen para el dominio turístico
SBR-DR: Pronósticos del tiempo	Reglas de Inferencia Etiquetador de Entidades Nombradas	Plantillas, Ontología de Eventos Climáticos, Base de Datos	Creado desde el origen para el dominio meteorológico
SBR-DR: Bell Canada	Lexicon de sinónimos	Diccionarios, Basado en terminología, Clustering (conceptos)	Sistema de RI como *baseline*
SBR-DR: Nobel Prizes&LT-World	Análisis Semántico (pregunta)	Ontología de Dominio, Bases de Datos	Sistema QUETAL como *baseline*
SBR-DR: Entorno Médico	Técnicas de PLN (Forma Lógica de la pregunta y emparejamiento de Patrones)	Metatesauro de UMLS	Creado desde el origen para el dominio médico

Cuadro 2.1. Propuestas de los SBR-DR según la clasificación definida

su arquitectura en función de los enfoques de las diferentes tendencias que existen.

Extrans ExtrAns (Hess *et al.*, 2002; Mollá, 1999), tiene como objetivo encontrar las respuestas en el texto y devolver lo encontrado literalmente, por lo que no necesita módulo de generación de la respuesta. Este sistema se basa en la restricción de la profundidad del análisis, y efectúa un módulo de conocimiento simple para desarrollar la base de conocimientos, y un módulo de generación simple para devolver la respuesta a la pregunta tal y como fue encontrada. De una manera resumida explicaremos las dos fases más importantes de la arquitectura del sistema:

Fase previa Construcción de la Forma Lógica Mínima (MLF: *Minimal Logical Form*), que no es más que una representación semántica de los documentos que se almacena en la base de conocimiento del sistema. Pero primero se realizan otros procesos

necesarios para obtener la representación semántica de cada frase, como:

- Se identifican las palabras y se marcan los términos propios del dominio utilizando un tokenizador y un módulo de procesamiento de la terminología.
- Se obtiene la estructura sintáctica de las oraciones usando el analizador lingüístico *Link Grammar* (Sleator & T., 1993).
- Se resuelven diferentes ambigüedades empleando la aproximación de Brill y Resnik (Brill & R., 1994).

La ventaja más significativa de las MLF es que producen un conocimiento mínimo del dominio, lo que hace muy fácil la portabilidad a otros dominios. También permite, gracias a la información semántica asociada, resolver sentencias problemáticas por ser largas las frases, por tener errores ortográficos o estructuras no reconocidas por el analizador sintáctico.

Fase on-line En esta fase las preguntas del usuario son analizadas usando el mecanismo base de MLF y almacenado esa representación en el base de conocimiento. Luego se presentan las posibles soluciones en dependencia de la similitud entre las representaciones de la pregunta y de las frases originales. Cuando el sistema no ofrece respuesta, se puede hacer una consulta más general manteniendo la misma forma lógica y relajando algunos criterios (añadir hipónimos). La salida del sistema permite la navegación, ya que las MLFs contienen punteros al texto original

En el cuadro 2.2) se puede ver una comparación de Extrans con respecto a un sistema de RI *baseline* como SMART. Se puede observar del análisis de los resultados que el ordenamiento que se obtiene del sistema RI no es el óptimo, es decir, los documentos que contienen las respuestas más probables no siempre son los primeros. Por el contrario Extrans siempre que devuelve una respuesta correcta, en casi todos los casos, es la primera; pero tiene como inconveniente que a veces devuelve respuestas incorrectas. Se puede concluir entonces que Extrans es un sistema de una alta precisión con mejor posicionamiento o *ranking*. El análisis del cuadro 2.2 nos deja ver que hubo una mejora importante en un 17de Extrans con respecto al sistema base.

	Baseline	ExtrAns
MRR (%)	46	63

Cuadro 2.2. Comparación entre Extrans y el sistema *baseline* SMART.

WEBCOOP El sistema WebCoop (Benamara, 2004) fue creado desde el inicio sólo para el dominio turístico. Brinda respuestas generales descriptivas en formato Web junto con explicaciones, por lo que se dice que es un sistema cooperativo.

Explicaremos de una manera simple el funcionamiento del sistema. Primero que todo en el módulo de Extracción de Información de las páginas Web podemos destacar que se cuenta con un recuperador de pasajes que extrae los pasajes de mayor importancia de las páginas. Luego el extractor de conocimiento obtiene la representación lógica de cada párrafo relevante. Para decidir que información es realmente relevante por medio de reglas y cómo organizarlas para dar una respuesta coherente e informativa, se utiliza el motor de inferencia de WEBCOOP. Para cada tipo de inferencia que se usa se definen plantillas generales en LN que traduce el mecanismo de razonamiento de los términos.

El sistema tiene un total de 28 plantillas básicas las que fueron extraídas desde dominios públicos y sus respuestas fueron normalizadas. En WEBCOOP las respuestas que se brindan al usuario tienen un estilo Web integrando la generación de LN con enlaces para facilitar respuestas complejas y dinámicas. La estructura de sus respuestas tiene dos partes:

- 1era: contiene los elementos de explicación del lenguaje natural.
- 2da: contiene los conocimientos técnicos del sistema cooperativo, basados fundamentalmente en procedimientos inteligentes de relajación de la pregunta, haciendo un uso cuidadoso de la ontología del dominio y de conocimiento general.

El sistema tiene un componente cooperativo cuya estrategia evita el problema de tener que adivinar la intención del usuario. Después de expuestos dos ejemplos de respuestas de WEBCOOP, será más fácil la comprensión de la tipología y la forma de representar el conocimiento utilizadas en el sistema. Los tipos de conocimiento usados son:

- Información puntual y de propósito general (lugares, nombre propios, distancias, etc).
- Información descriptiva (horarios de vuelos, precios de hoteles, etc).
- Conocimiento de sentido común y restricciones (para un viaje dado puede ser, por ejemplo, que el tiempo de llegada es mayor que el tiempo de salida).
- Conocimiento jerárquico asociado con las propiedades del objeto (un restaurante está caracterizado por su categoría, tipo de comida, localización, etc).
- Procedimientos o instrucciones que describen como preparar un viaje o reservar una habitación.
- Definiciones.
- Regulaciones, advertencias.
- Criterios de clasificación de acuerdo a propiedades específicas (órdenes de hoteles en función de su categoría, etc).
- Funciones de interpretación (términos confusos como caro, lejos de la playa, etc).

Entre las formas de representación del conocimiento en WEBCOOP están:

- Conocimiento general y de domino representado por medio de hechos, reglas y restricciones.
- Conjunto grande de textos indexados por sus formas lógicas.

A continuación se describen los recursos de conocimiento desarrollados por WEBCOOP:

- Conocimiento base: implementado en Prolog (reglas, nombre de ciudades, descripciones, restricciones, etc). Tiene un total de 170 reglas y 47 restricciones.
- Ontología: presenta la representación de cada nodo por un predicado tipo: *onto-node(concept,lex,properties)*, donde el concepto es descrito usando las propiedades y las posibles representaciones léxicas del concepto están el *lex*.
  En WEBCOOP se integraron manualmente dos ontologías existentes, una francesa y otra bilingüe (francés-inglés) , formándose así una ontología con un total de 1000 conceptos.

- Lexicon: contiene nombres, verbos, y adjetivos extraídos de las dos ontologías anteriormente nombradas.
- Textos indexados: Los textos son indexados indicando la categoría, la palabra clave o la fórmula que lo identifica y el propio texto. Para ello el extractor de conocimiento, que está basado en la ontología del dominio, transforma cada fragmento de texto en una representación lógica: *text(F, http)* donde F es la fórmula que representa el conocimiento extraído de la página Web y http es la dirección de esa página.
- Representación y evaluación de la pregunta: el procesamiento de la pregunta permite identificar: el tipo de pregunta (*boolean* o *entity*), el objetivo o enfoque de la pregunta y la representación semántica asociada en lógica de primer orden.

WEBCOOP sólo se tratan preguntas booleanas y de entidades, que tiene esquemas de inferencia del tipo: Presuposición falsa (FP), Falso concepto (MIS) y Relajación de la Pregunta (RR). Elaboraron un corpus de parejas de pregunta-respuesta extraídas de diferentes sitios Web. El 60 % está dedicado al turismo, el 22 % a la salud y el resto dedicado a las compras, el deporte y la educación. Del análisis del corpus se identifican las categorías conceptuales de las preguntas, así como la categorización de las funciones cooperativas.

SBR-DR para pronósticos del tiempo El sistema es un robot doméstico (Chung *et al.*, 2004b) que alcanza una elevada precisión y para las preguntas sin respuesta, prefiere la ausencia de respuesta antes que brindar una contestación incorrecta. La arquitectura básica consta de dos motores:

- Motor de extracción de información (IE): está compuesto por un crawler que es el encargado de descargar páginas seleccionadas desde *Korea Meteorological Administration* (KMA) a cada hora y un *wrapper* que extrae información sobre el tiempo en las páginas descargadas. La información sobre el clima en las páginas es semi-estructurada, por lo que su extracción es relativamente fácil, aunque no trivial, ya que hay que eliminar determinados elementos irrelevantes como pueden ser imágenes, anuncios y etiquetas HTML.

Actualmente, el motor de IE extrae la siguiente información:

1. Observaciones diarias: resúmenes del clima, visibilidad, temperatura, vientos, humedad relativa.
2. Pronósticos por 7 días: resúmenes del clima, pronósticos de temperatura (altas/bajas).

- Motor de BR: traduce las preguntas a *Structured Query Language* (SQL), envía las instrucciones SQL a la base de datos en Oracle (DBMS) que almacena la información sobre las páginas web y finalmente la respuesta es traducida al LN

El analizador de la pregunta lo primero que hace es identificar las palabras claves que describen a la pregunta realizada por el usuario, como pueden ser palabras de eventos, fechas, tiempo, y localidad. El analizador está compuesto por un etiquetador de entidades para expresiones temporales, nombres de lugares y eventos del clima, un normalizador de datos temporales y algunas reglas de inferencia.

El etiquetador de entidades consulta una ontología dependiente del dominio que recoge eventos del clima, y una ontología independiente del dominio para los nombres de lugares. La ontología para los eventos del clima consta de conceptos de eventos, similares a los synsets en WORDNET. Por ejemplo: *rain and umbrella* están en el mismo concepto de evento de la ontología, porque las preguntas que usan *umbrella* generalmente preguntan sobre *raining* (*Will I need to bring umbrella tomorrow? and Will it be raining tomorrow?*).

El normalizador de datos temporales convierte las expresiones temporales como hoy, este fin de semana y ahora en valores absolutos que pueden ser usados al consultar la base de datos. Y las reglas de inferencia se usan en el caso de que la pregunta realizada por el usuario no exprese la fecha, el tiempo y la localidad, así se infiere esa información utilizando perfil del usuario.

Ahora explicaremos, a través de un ejemplo, el funcionamiento del clasificador de las plantillas de preguntas (query frames). El robot emplea *query frames* por lo que para cada tipo de pregunta se define una plantilla (a continuación se muestra un ejemplo).

Como se trata de un dominio restringido es factible la definición de tales plantillas. Se muestra a continuación ejemplos de plantillas:

[PRECIPITATION_TOMORROW]

Cuadro 2.3. Ejemplo de plantilla del SBR-DR para pronósticos del tiempo

Y cada query frames tiene asociada una regla SQL como la siguiente:

[PRECIPITATION_TOMORROW] SELECT date, amprecpr, pmprecpr FROM forecast_tbl WHERE $date $city

Cuadro 2.4. Ejemplo de consulta SQL generada por el SBR-DR para pronósticos del tiempo

Esa regla SQL se ejecuta sobre la base de datos disponible en la aplicación recuperando la probabilidad de la lluvia para mañana.

Los resultados obtenidos por el sistema para un total de 50 preguntas sobre el clima se pueden ver en el cuadro 2.5.

	Precisión	Cobertura
Robot (%)	90.9	75

Cuadro 2.5. Resultados obtenidos por el el SBR-DR para pronósticos del tiempo

SBR-DR para Bell Canada Es un sistema (H. Doan-Nguyen and L. Keila, 2004) asistido por el criterio humano en el chequeo de las respuestas devueltas (sistema semi-automático) a consultas sobre los servicios, tanto a particulares como a empresas, que ofrece la compañía Bell Canada. El SBR-DR emplea un esquema de dos fases:

- 1era fase: Extrae una lista de respuestas candidatas, utilizando el RI OKAPI[11] (Robertson *et al.*, 1995), junto con la relevancia de cada una y el nombre del documento que la contiene.

[11] www.soi.city.ac.uk/ andym/OKAPI-PACK/

- 2da fase: Extrae la respuesta correcta.

Cada respuesta devuelta consiste en un párrafo que para OKAPI ha resultado el más acertado en la respuesta. Las respuestas candidatas serán evaluadas por el criterio de los humanos usando una escala binaria: correctas e incorrectas. Esta clase de criterio es recomendado en el contexto de las comunicaciones entre una compañía y sus clientes, porque las condiciones y los detalles técnicos de un servicio deben ser editados tan claramente como sea posible en la respuesta al cliente. Aunque se acepta algo de tolerancia en los casos de preguntas ambiguas, como por ejemplo si el cliente pregunta por teléfonos pero no específica si es o no inalámbrico, serán aceptados todos los candidatos correctos para cualquiera de los dos casos.

El proceso explicado es muy adecuado para este tipo de sistemas que trabajan directamente con el usuario, puesto que la respuesta puede venir acompañada de información adicional, que el usuario agradecerá y que no debe suponer una menor relevancia de la respuesta.

El corpus fue creado con un total de 220 documentos, a partir de páginas HTML y ficheros PDF de la compañía, asegurando que algún documento contuviese la respuesta. La forma de las preguntas era arbitraria, presentando una media de 11.3 palabras de longitud. El total de preguntas era de 140, las que fueron divididas para entrenamiento (80) y prueba (60).

En cuanto a los resultados del sistema, se realizó primero la evaluación del sistema baseline antes de la introducción de las mejoras. El sistema base obtiene unos resultados que se muestran en la figura que sigue. Como todas las respuestas candidatas son evaluadas por un agente humano se llegó a la conclusión de que no era apropiado usar la medida típica de MRR, por lo que la nomenclatura y los pasos empleados fueron:

- Para cada pregunta se obtuvo una lista de los 10 mejores candidatos.
- C(n): es el número de candidatos que son correctamente evaluados.

- Q(n): es el número de preguntas en el conjunto de entrenamiento que tienen al menos una respuesta correcta entre las n primeras candidatas.
- El chequeo manual realizado después fue fijado con n=5, para que la revisión no resultase excesivamente tediosa.

Se puede comprobar que al menos 45 preguntas (conjunto de entrenamiento) y 37 preguntas (conjunto de prueba) tienen una respuesta correcta entre la primera y la décima posición. Para mejorar la precisión del BR-DR se proponen dos formas:

- Mejorar la precisión del sistema de RI.
- Mejorar los resultados devueltos por RI con información específica para aumentar la relevancia de la respuesta candidata correcta.

Como nuestro objetivo es ver como se alcanza la adaptabilidad del sistema al dominio, analizando la información específica del mismo, centraremos el análisis en la segunda propuesta. Para ello presentaremos los experimentos realizados, para luego realizar una comparación entre sus resultados con respecto a la evaluación obtenida por el sistema *baseline*. Los experimentos fueron:

1. Reordenación de la lista de candidatas: se realizaron dos experimentos dentro de este enfoque, teniendo en cuenta los recursos empleados para realizar la reordenación:
   a) Reordenación empleando vocabulario específico: Se realiza asignando una nueva medida que aumenta el peso de las respuestas candidatas que contengan términos del vocabulario. Se emplearon términos muy discriminatorios (nombres de los servicios, solían aparecer en mayúscula, por lo que era fácil su extracción de manera automática) para conformar el vocabulario. Luego se realizó un filtrado manual, con lo cual quedaron 450 términos especiales. Tiene como desventaja que no se tratan los documentos en los que no aparece ningún término de los contenidos en el diccionario.

b) Reordenación usando una caracterización de los documentos: Para solucionar la desventaja del experimento anterior, en éste lo que se hace es clasificar los documentos en función de conceptos. Esto es viable en los sistemas de dominio restringido, donde se puedan clasificar los documentos mediante ciertos tópicos. Se construyó una jerarquía de clustering empleando la clasificación original de Bell Canada. Entonces para la BR reconocían los conceptos de la pregunta y determinaban los documentos relevantes a tales conceptos. Para clasificar la pregunta se asocian a los distintos conceptos de la jerarquía un conjunto de términos y se comparan las palabras de la pregunta con ellos, obteniéndose el concepto más relevante (si existe más de uno toman el de mayor valor). Se creó también, de manera manual, un lexicón de sinónimos.

2. Búsqueda de candidatas en dos niveles: En estos experimentos se combina al motor de RI con la clasificación de los documentos señalando su relevancia a la pregunta a través de los conceptos que los caracterizan. En los experimentos anteriores la RI se llevaba a cabo en la colección entera de documentos, ahora se propone llevarla a cabo en dos niveles. Las aproximaciones seguidas son:
   a) Combinar OKAPI con la clasificación de los documentos 1.b: En este experimento la entrada de OKAPI, para buscar la respuesta candidata, es el subconjunto de documentos (normalmente unos 20) devueltos por la clasificación anterior y se indexaría de forma separa cada pregunta. La conclusión final del experimento es desfavorable ya que los resultados en la pruebas fueron peores que en los experimentos anteriores.
   b) Cambiar el motor de RI: Aquí se modifica el motor de RI. Se realiza un filtrado previo de los documentos para obtener un subconjunto de documentos más relevantes asociado a cada pregunta y el nuevo motor de RI realiza la búsqueda de las respuestas candidatas en esos documentos. Si no devuelve ningún documento relevante, se considera la lista propuesta por OKAPI.

3. Extensión de las respuestas candidatas: Otra prueba fue extender las respuestas candidatas para mejorar la precisión. Se desarrollaron dos técnicas diferentes:
   a) Una aproximación extendía cualquier candidato devuelto por OKAPI que viniera de un documento de menos de 2000 caracteres, considerando el documento completo.
   b) Otra aproximación extendía los candidatos devueltos por el proceso de búsqueda de dos niveles del experimento anterior.

   En ambos casos hubo mejoras pero menos importantes que en otros experimentos anteriores, por lo que la idea sería combinarlo con otros métodos.

Experimentos						
n=5	1.a	1.b	2.a	2.b	3.a	3.b
Conjunto de Entrenamiento (80 preguntas)						
Q(n)	44	44	48	61	47	60
delta Q(n)	5	5	9	22	8	21
% delta Q(n)	12.8	12.8	23.1	56	20.5	53.8
Conjunto de Prueba (60 preguntas)						
Q(n)	34	35	33	42	37	41
delta Q(n)	2	4	1	10	5	9
% delta Q(n)	6.3	12.5	3.1	31.3	15.6	23.1
Orden	5to	4to	6to	1ero	3ero	2do

Cuadro 2.6. Resultados de los experimentos del BR-DR de *Bell Canada*

En el cuadro 2.6 se muestra una comparación de todos los experimentos según los resultados alcanzados, a cada experimento se le asignará un orden con respecto a los resultados en el conjunto de prueba, donde el 1ero será el que alcanzó mejores resultados. Debe aclararse que:

En el resto de las evaluaciones para cada pregunta se obtuvo una lista con sólo 5 candidatos y se empleó:

Donde: Q (5) de OKAPI es 39 en el conjunto de entrenamiento y 32 en el conjunto de prueba.

SBR-DR para Nobel Prizes&LT-World En esta aproximación (Frank *et al.*, 2005) se crearon como recursos una ontología y

delta Q(n)=Q(n) del experimento en SBR-DR - Q(n) de Okapi
% delta Q(n)= delta Q(n) / Q(n) de Okapi 100

Cuadro 2.7. Ecuación empleada en la evaluaciones del SBR-DR para Bell Canada

una base de datos (BD) para manipular los conocimientos de los dos dominios con los que interactúa (Premios Nobel y Tecnologías del Lenguaje Humano). Usaron las siguientes ontologías como referencia para diseñar la ontología propia de su dominio: SUMO y WordNet, siendo SUMO la ontología vertebral y definieron a partir de eso subconceptos entre las dos. Los conceptos principales de la aplicación son premio, laureados, área de premio, entre otros, que incluyen conceptos generales como persona y organización.

La ontología y BD para LT-World (segundo escenario de dominio restringido extraído del portal[12] de LT-World), provee información acerca de personas, productos, recursos, proyectos y organizaciones. Es libre y divulgado por el German Research Center for AI (DFKI) para la comunidad de R&D, usuarios pontenciales de las tecnologías del lenguaje, estudiantes y otros interesados. La ontología usa RDF y RDF schema (RDFS) y ha sido recientemente portada a OWL (Ontology Web Language), es el lenguaje para web semántica que se originó de la estandarización de DAML+OIL. OWL usa estructuras RDFs y RDF, pero puede describir recursos a mayor detalle dado por una buena definición semántica del modelo teórico heredado de una descripción lógica.

La arquitectura del sistema es una extensión a las tradicionales Interfaces de BD en LN (NLIDB: Natural Language Interface Data Bases) desarrolladas en los años 70 y 80. Contrario a los enfoques de NLIDB tradicionales, la arquitectura usa una ontología como interfaz entre el análisis de pregunta, la extracción de respuesta y la ingeniería del conocimiento. Se basa en recursos más generales en el modelado lingüístico y del conocimiento, y una separación clara respecto a capas modulares: el análisis semántico lingüístico, la representación léxico-conceptual y el modelado conceptual basado en el conocimiento.

[12] www.ltword.org

El SBR-DR que se analiza utiliza como sistema baseline al sistema QUETAL[13] , cuya arquitectura global se puede decir que es hibrida en dos sentidos:

- El análisis de pregunta es híbrido ya que se combina un PLN superficial y profundo para producir una representación de la pregunta robusta y enriquecida semánticamente.
- La base de conocimiento de donde se extraerá la respuesta es híbrida ya que emplea tres tipos de fuentes de información: (i) estructurada (hechos almacenados en BD relacionales u ontologías que reflejan los conceptos, sus relaciones y hechos específicos del dominio), (ii) semi-estructurada (texto enriquecido-XML- offline con técnicas de PLN y de EI) y (iii) no estructurada (textos recuperados a través de motores de búsqueda sobre textos completos locales o vía web utilizando sistemas de RI).

En QUETAL primero se realiza un análisis lingüístico y de ello obtiene una clasificación de la pregunta y de los tipos de respuesta esperada. Luego se selecciona una fuente de información para extraer las respuestas candidatas, por último se elige una respuesta. Soporta QA multilingüe, por lo que tiene etapas de traducción intermedia. Para profundizar se puede consultar (Neumann, 2003).

A continuación se explicarán los aspectos fundamentales de las diferentes fases de la arquitectura del sistema.

1. Análisis de la pregunta: Usa una arquitectura de PLN HoG (Heart-of-Gold), la misma integra de una manera flexible componentes de PLN superficial y profundo, por ejemplo: PoS tagger para reconocimiento de Entidades Nombradas (EN) y parser HPSG. El HoG es interpretado y genera un prototipo de pregunta que puede ser visto como una interpretación independiente de alto nivel de una BD u ontología, se construye por tanto una instancia de una BD específica o una pregunta ontológica, para acceder a los recursos de información y devolver una respuesta en LN.

[13] http://quetal.dfki.de

El sistema tiene una amplia cobertura para la gramática HPSG para inglés y alemán en la BR. Ambas gramáticas están integradas con el reconocimiento de EN realizado por el sistema de EI SProUT, el cual provee de una representación estructurada para clases generales de EN y los términos del dominio.
En la arquitectura HoG el RMRS (Robust Minimal Recursion Semantic), entregado por el parser HPSG, constituye un formato de intercambio entre todos los componentes de PLN, incluyendo el reconocimiento de EN.

2. Interpretación de la pregunta: Para la interpretación de la pregunta se analiza la información semántica codificada en la representación RMRS: tipo de mensaje y marcado de frase_wh. La representación de las preguntas en el RMRS se marcan por un tipo de relación semántica, ejemplo: int_m_rel son los mensajes de tipo interrogativo y en preguntas Wh los pronombres introducen un sin número de relaciones como: person_rel (who), imp_m_rel (imperativas), time_rel (when), etc. Dicha información debe ser enriquecida para obtener preguntas concisas para la extracción de la respuesta de los recursos de conocimiento estructurado.
La información mínima que identifican es q_var (variable pregunta) en la FL RMRS. Después se determina el tipo de respuesta esperada (EAT) a partir de la información que contiene q_var, definiendo el tipo ontológico de q_var y extrayendo desde la representación RMRS las restricciones relacionales para la construcción de la pregunta. Esas restricciones relacionales, definen un supuesto prototipo de pregunta (proto query), que es traducido en una consulta- en un lenguaje formal- a la base de conocimiento.
El proceso de interpretación de la pregunta se realiza entonces en tres pasos:

   - Correlacionar la representación RMRS con una representación semántica conceptual: se obtiene de la utilización de la BD creada en el marco del proyecto FrameNet. Esa BD contiene las descripciones de los marcos semánticos (frame semantic) para los verbos, sustantivos y adjetivos, donde

un marco modela una situación conceptual con los roles específicos del concepto que identifican a los participantes en esa situación. A continuación se muestra un ejemplo de este proceso en un frase semánticamente equivalente a las expresiones típicas que se usan para pedir información sobre los Premios Nobel:

- Aplicar reglas de inferencia: usando la ontología de dominio se aplican reglas de inferencia simples para enriquecer las representaciones de los marcos semánticos, pudiendo agregar variables de argumentos no instanciados para elementos de marcos no expresados. Siguiendo con el ejemplo anterior sería:
  Donde el rol LAUREATE dentro del marco AWARD se refiere a una variable en la FL que retornó el marco LAUREATE, en el que uno de los roles semánticos es NAME, AFFILIATION, etc. Se extiende así el marco AWARD con el marco LAUREATE enlazados por la variable x1.
  Multilingüe: como los marcos están definidos como estructuras conceptuales ellos pueden ser extendidos independientemente del lenguaje utilizado.
- Construcción de la proto-query: es necesario obtener el concepto de la pregunta, y para ello es necesario extraer primero los marcos relevantes del dominio y luego las restricciones del dominio, usando la ontología definida. Se muestra un ejemplo de una pregunta conceptualizada y como se obtiene la proto-query usando además la estructura del marco semántico:

3. Construcción de la Pregunta: La proto-query identifica: (i) el concepto de tipo de respuesta, que corresponde con el valor del comando SQL SELECT, (ii) los conceptos adicionales y valores que restringen el tipo de respuesta, estos conceptos llenan la condición SQL WHERE y (iii) las dependencias entre preguntas elementales, si una pregunta es compleja y necesita ser descompuesta en sub-preguntas. Entonces las proto-queries dadas por la interpretación de la pregunta son traducidas a SQL-queries.

La tarea de traducción a la consulta SQL es primero la de identificar las tablas donde se pueden encontrar los conceptos que se solicitan y en segundo lugar los campos de la tabla relevantes a emparejar con los valores dados por la proto-query. Para ello definieron reglas de correlación entre los marcos de FrameNet y las tablas y campos de la BD.

4. Procesamiento de la respuesta: Las instancias de las relaciones de dominio son almacenadas en una base de datos relacional MySQL. Se almacenan, por ejemplo, en el dominio de los Premios Nobel los ganadores del premio en dos tablas separadas, una para personas (winner_person) y otra para organizaciones (winner_organization), desde que se asociaron con diferentes atributos los conceptos persona y organismo. El procesamiento de la respuesta es sólo devolver el valor disponible en la BD, devuelto por la consulta SQL.

SBR-DR para entorno médico El sistema de BR está creado desde el inicio para un dominio médico (Terol *et al.*, 2006). El módulo de análisis de la pregunta es el núcleo del sistema ya que condiciona considerablemente los resultados del resto de módulos y, en definitiva, de la respuesta final. Está basado en técnicas de PLN y utiliza el Metatesauro de UMLS (Unified Medical Language System) como fuente de conocimiento. La principal técnica de PLN utilizada consiste en el tratamiento computacional de la forma lógica de la pregunta y en el emparejado de patrones. Por lo que centraremos la exposición en este módulo, satisfaciendo así nuestro principal objetivo, como hasta ahora, que no es más que la descripción de las técnicas que permiten la adaptación al entorno de los SBR-DR más relevantes en la actualidad.

La estructura del sistema es modular basada en cuatro módulos principales: análisis de la pregunta, recuperación documental, filtrado de frases o pasajes, y extracción de la respuesta. Para profundizar en el resto de los módulos consultar(Terol, 2005).

Entre los recursos del conocimiento creados destacan:

- Taxonomía de preguntas concretas: se obtuvo a partir de un estudio desarrollado de las principales preguntas formuladas por una serie doctores determinados (103 médicos de familia de Io-

wa y 49 doctores de atención primaria de Oregon). La misma quedó formada por las diez preguntas más frecuentemente formuladas por estos doctores, un ejemplo de ellas es: *What is the drug of choice for condition x?*

Por tanto, el sistema BR-DR es capaz de responder preguntas en base a esa taxonomía de preguntas genérica, descartando otro tipo de preguntas. Este hecho produce por una parte, una baja cobertura pero, una elevada precisión con el objetivo de que sea muy funcional en el dominio médico.

Los patrones de las preguntas genéricas, tratadas por el sistema acorde a la taxonomía de preguntas médicas presentada, se construyen según dos enfoques: generación manual de patrones y generación automática de patrones supervisada. Para poder desarrollar los patrones, el sistema hace uso de recursos de PLN como: Formas Lógicas, Reconocimiento de Entidades Médicas y Conocimiento Semántico. A continuación mostramos como se utilizan.

- Formas Lógicas (FL): se calculan a partir del análisis de dependencias entre las palabras de la frase. Este recurso está desarrollado a partir de una serie de reglas que infieren distintos atributos en la FL como el predicado, el tipo de predicado, el identificador del predicado y sus relaciones con otros predicados de la FL. Sigue la nomenclatura empleada por el recurso Logic Form Transformation de eXtended WordNet(Harabagiu & Moldovan, 1999). Por ejemplo, a partir de las dependencias entre las palabras de la frase *"Nerve cells consist of a large cell body and nerve fibers"* se infiere automáticamente la siguiente FL:

"nerve:NN(x3) NNC(x4,x3, x5) cell:NN(x5) consist of:VB(e1,x4,x10) large:JJ(x1) cell:NN(x2) NNC(x1,x2,x9) body:NN(x9) and:CC(x10, x1,x7) nerve:NN(x6) NNC(x7, x6, x8) fiber:NN(x8)".

Cuadro 2.8. Ejemplo de forma lógica del SBR-DR para entorno médico

Aclaremos que en el formato de la FL, cada predicado tiene al menos un argumento, pudiendo tener más de uno: el primer argumento se corresponde con el identificador del predica-

do mientras que el resto de los argumentos se asocian con los identificadores de otros predicados relacionados con el predicado actual.
- Reconocimiento de Entidades Médicas (EM): aporta al sistema toda la información referente a las entidades del dominio médico: nombres de medicamentos, síntomas, enfermedades, disfunciones, etc. La base de conocimiento de este recurso es el Metatesauro de UMLS.
- Conocimiento semántico: se extrae de fuentes de conocimiento. Por una parte, se usa WordNet (Miller, 1990) para extraer las relaciones semánticas entre términos de propósito general. Por otra parte, utilizamos el Metatesauro de UMLS para obtener las relaciones semánticas entre los términos médicos.

Un patrón está formado por una serie de entidades médicas y uno o más verbos. Seguidamente se describen los dos métodos de obtención de patrones citados anteriormente.

Generación manual de los patrones se desarrolla del siguiente modo:

- Identificación de los tipos de EM que se deben emparejar en la pregunta genérica.
- Identificación de los verbos que se deben emparejar en la pregunta genérica.
- Expansión automática de estos verbos en base a sus relaciones de similitud definidas en WordNet.
- Seleccionar el umbral de EM inferior (MELT) y superior (MEUT), que se deben emparejar entre el patrón de la pregunta genérica y la pregunta del usuario de cada patrón.
- Identificar los posibles tipos de respuesta esperados.

El verbo principal asociado a esta pregunta debe corresponder con alguno de los verbos de la lista (treat, control, take, associate with, help, prevent, manage, indicate, relieve, evaluate, help, fight and solve). Esta lista de verbos se completa automáticamente con los verbos expandidos de WordNet que tienen alguna relación de similitud con los primeros. Finalmente, los tipos de respuesta esperados para esta pregunta genérica se instancian manualmente a

los tipos semánticos de UMLS "Sustancia Farmacológica" y "Droga Clínica".

Generación automática supervisada de patrones: se realiza a través del procesamiento de las preguntas emparejadas a la taxonomía de preguntas del siguiente modo:

- Derivación de la FL asociada a cada pregunta.
- Reconocimiento de las EM en la FL. Las entidades médicas sólo pueden ser sustantivos (predicado del tipo NN) o nominales complejos (predicado del tipo NNC) incluyendo sus posibles modificadores (predicado del tipo JJ).
- Reconocimiento del verbo principal en la FL.
- Expansión automática de este verbo principal a través de las relaciones de similitud con otros verbos dadas por WordNet.
- Asignación automática del umbral MELT = número EM en la FL - 1 y MEUT = número de EM en la FL.
- Asignación manual de los tipos de respuesta esperados para cada pregunta.

El proceso es supervisado por un usuario administrador avanzado del sistema pudiendo modificar los resultados obtenidos en cada uno de estos pasos.

Una vez que el sistema tiene construidos los patrones asociados a cada una de las preguntas genéricas tratadas, se puede iniciar el proceso de búsqueda de respuestas, para lo que se hace necesario clasificar y analizar las preguntas en LN formuladas por el usuario.

Clasificación de la Pregunta: consiste en asignar uno de los patrones genéricos a la pregunta del usuario y en caso de no corresponderse con ninguna de las diez preguntas genéricas tratadas el sistema no brinda respuesta. Para poder realizar esta tarea los primeros pasos son equivalentes a los descritos para generar de manera automática los patrones y luego se debe:

- Construir la forma de la pregunta del usuario y asignar su marcador de EM MESQ = número de EM en la forma de la pregunta del usuario.
- Obtención de aquellos patrones en cuya lista de verbos esté contenido el verbo principal de la FL de la pregunta del usua-

rio y que además cumplan la restricción $MELT < MESQ < MEUT$.
- Asignación de la medida de emparejamiento de entidades EMM = número de EM que se deben emparejar entre la pregunta y el patrón.
- Selección del patrón que minimiza la diferencia entre EMM y MELT.

Análisis de la Pregunta: consiste en realizar un complejo procesamiento de la pregunta del usuario en base al patrón emparejado y su correspondiente pregunta genérica. Los dos pasos más importantes que se ejecutan en esta fase son:

- Reconocer el tipo de respuesta (enfermedades, síntomas, dosis de medicinas) esperado usando Wordnet y UMLS. Es el más importante.
- Identificar las palabras claves de la pregunta del usuario usando una serie de heurísticas a los predicados y sus relaciones con otros predicados en la FL. Las palabras claves son nominales complejos o nombres reconocidos como EM (utilizando el reconocedor de EM), adjetivos modificadores de EM, el resto de nominales complejos y nombres que no son EM incluyendo sus modificadores y el verbo principal en la FL de la pregunta del usuario. Por ejemplo, en la FL: "high:JJ(x3) blood:NN(x1) NNC(x3, x1, x2) pressure:NN(x2)", el predicado x3 es reconocido como una Enfermedad o Síndrome, entonces "high blood pressure" tiene el tratamiento de palabra clave.
- Expansión de las palabras claves aplicando una serie de heurísticas. Por ejemplo, las EM pueden ser expandidas utilizando las relaciones de similitud definidas en UMLS. De este modo, "high blood pressure" se expande a "hypertension".

Note que si son extraídas muchas palabras claves de la pregunta del usuario, solamente un número máximo de palabras clave son consideradas por el siguiente proceso de recuperación documental teniendo en cuenta la prioridad (las EM son las que tienen mayor prioridad).

Se alcanzó un 94,4 % en la precisión de la tarea de la clasificación de la pregunta (ver cuadro 2.12), lo que contribuye de seguro a que el resto de los procesos tengan un mejor resultado.

Donde:

- GQ: conjunto de 50 preguntas que cubren los 10 tipos de preguntas genéricas.
- OQ: conjunto de 200 preguntas en inglés para la tarea de evaluación de BR en el CLEF.
- GE: conjunto de preguntas genéricas definidas.
- OE: conjunto que incluye el resto de las preguntas que no se emparejan con ninguna pregunta genérica.

2.4. Conclusiones

Según los resultados oficiales de la conferencia TREC, los sistemas de búsqueda de respuestas en dominios abiertos obtienen unos resultados de evaluación que rondan entre el 30 y el 40 por ciento de precisión, lo cuál no es factible cuando se van a utilizar en determinados entornos reales. Por lo que al concluir el capítulo queda justificada la necesidad de mejorar la precisión de los sistemas de BR en determinados dominios concretos, para alcanzar una buena satisfacción por parte de los usuarios al ponerlos en práctica en el mundo real. En algunos casos, es obligatorio para la aplicación de estos sistemas que alcancen niveles elevados de precisión, ya que una respuesta incorrecta puede causar daños irreparables. Se describieron los antecedentes de los BR-DR y se citaron los diferentes enfoques para la creación de los mismos o la adaptación de un sistema de BR baseline a un dominio restringido cualquiera. Por lo que se especificó para cada una de las propuestas más relevantes de BR-DR halladas en la bibliografía consultada, el origen del sistema que puede ser creado a propósito para el dominio o a partir de un sistema baseline (como es nuestro caso). Para cualquiera de los dos casos se resaltaron los recursos empleados para la adquisición y almacenamiento del conocimiento dentro del dominio, enfocándonos siempre en cómo consiguen la adaptación al entorno restringido.

Capítulo 3

Estado Actual de la Adaptación de la Búsqueda de Respuestas a Dominios Restringidos

El capítulo anterior se realizó un estudio de los SBR-DA y los SBR-DR. Mientras estos últimos tienen en cuenta las características propias del dominio restringido correspondiente, los primeros son más generales y necesitan de un proceso de adaptación para poder emplearse en dominios restringidos. En este capítulo se presenta el estado actual de tres aspectos fundamentales a tener en cuenta en la adaptación de sistemas de BR a dominios restringidos. En primer lugar se describen las propuestas existentes para tratar con el ruido en los sistemas de RI, haciendo hincapié en su utilidad para sistemas de BR-DR. En la siguiente sección, se hace una comparación de las propuestas existentes para la definición de taxonomías de tipo de respuesta esperada y su problemática en los dominios restringidos. Finalmente, se estudia el proceso de adaptación de las diferentes fases (esto es, análisis de la pregunta y extracción de la respuesta) dentro de los sistemas de BR a nuevos dominios restringidos, así como los principales problemas existentes.

3.1. Tratamiento del ruido textual

Los seres humanos afrontan continuamente el ruido cuando escriben o leen documentos y en la mayoría de las ocasiones de manera inconsciente. Por ejemplo, durante la lectura[1] los ojos

[1] La lectura es el proceso de la recuperación y aprehensión de algún tipo de información o ideas almacenadas en un soporte y transmitidas mediante algún tipo de código, usualmente un lenguaje, que puede ser visual o táctil.

no realizan movimientos regulares sino discontinuos, es decir, no siguen las líneas impresas de una forma uniforme y continua sino que proceden a base de saltos y fijaciones. Un buen lector hace fijaciones amplias y en cada una de ellas capta con claridad cuatro o cinco letras y percibe otras no tan claras durante los saltos, pero que el cerebro sí reconoce y capta. La mayor parte de los saltos están entre 5 y 12 caracteres siendo de 7 caracteres la longitud más frecuente de los mismos (Golder & Gaonach, 2003). Precisamente por ese motivo el lector es capaz de tolerar la presencia de ruido en las palabras y es capaz de identificarlas de manera correcta. Entre los factores más importantes para identificar una palabra están sus primeras tres letras, la última letra, el empleo de minúsculas y mayúsculas y la identidad fonética. Sin embargo en la actualidad no se han logrado sistemas informáticos que imiten esta capacidad humana con el mismo nivel de eficiencia.

En esta sección se introduce la definición de ruido textual y se analiza los orígenes del mismo. Además se describe cómo afecta el ruido a determinadas aplicaciones de Procesamiento del Lenguaje Natural (PLN), por ejemplo clasificación de textos, generación de resúmenes, extracción de información, búsqueda de respuestas, etc. Luego se profundiza en algunas de las técnicas usadas para superar los efectos negativos provocados por la presencia de ruido. A continuación se exponen más detalladamente algunas aproximaciones actuales para lidiar con el ruido en los sistemas de RI, por ser uno de los objetivos de este trabajo. Finalmente se llega a conclusiones sobre los problemas aún sin resolver y las desventajas de las propuestas estudiadas desde el punto de vista de su aplicación en dominios restringidos.

3.1.1. Definición y orígenes del ruido textual

Una de las definiciones de ruido textual más conocidas es la que brindó Knoblock en (Knoblock *et al.*, 2007): *"any kind of difference between the surface form of a coded representation of the text and the intended, correct or original text"*, es decir, cualquier clase de diferencia entre la forma de representación codificada o electrónica de un texto y el texto pretendido, correcto u original.

A partir del análisis de diversos trabajos de investigación, como (Subramaniam *et al.*, 2009) y otros que citaremos más adelante, podemos afirmar que el ruido textual puede estar provocado por dos motivos principales:

Errores de herramientas automáticas de procesamiento de textos Una de las mayores fuentes de generación de ruido textual está dada por los resultados erróneos de aplicaciones que procesan textos electrónicos de manera automática; por ejemplo, aplicaciones de PLN a través de técnicas y herramientas como:

- Reconocimiento Óptico de Caracteres: en inglés *Optical Character Recognition* (OCR). Se encarga de la digitalización de información textual escrita, tipográfica o manuscrita. Identifica automáticamente símbolos o caracteres que pertenecen a un determinado alfabeto, a partir de imágenes escaneadas y empleando técnicas de reconocimiento de patrones. Los resultados se almacenan en forma de datos que se podrán editar y utilizar mediante programas de edición de texto o de PLN. Partiendo de una imagen perfecta (o sea, una imagen con sólo dos niveles de gris), el reconocimiento de los caracteres se realizará básicamente comparándolos con unos patrones o plantillas que contienen todos los posibles caracteres. Remitirse a (Rice *et al.*, 1999) para tener una excelente referencia acerca de los errores OCR.
  Sin embargo, las imágenes reales no son perfectas por tanto las salidas de herramientas OCR con frecuencia contienen errores debido al ruido introducido por el procesamiento automático de las mismas. Los errores en este caso se refieren a las diferencias entre la salida OCR y la fuente original. Específicamente, algunos de los problemas que provocan ruido en la salida de los OCR son:
  - Niveles de grises introducidos por el dispositivo que obtiene la imagen y que no pertenecen a la imagen original.
  - Ruido introducido en la imagen por la resolución del dispositivo empleado para su captura.
  - Distancia desigual de separación entre los caracteres.
  - Conexión de dos o más caracteres por píxeles comunes.

Por todo esto en los textos de OCR, el aislamiento de las palabras es mucho más difícil desde que los errores pueden incluir la sustitución e inserción de números, signos de puntuación y caracteres no alfabéticos. Además puede suceder que un caracter simple sea reconocido como múltiples caracteres (p.e. "rn" en lugar de "m"), o el caso contrario que múltiples caracteres sean reconocidos como uno solo (p.e. "d" en lugar de "cl").

- Reconocimiento Automático del Habla o de Voz: en inglés *Automatic Speech Recognition* (ASR). Es una parte de la Inteligencia Artificial que tiene como objetivo permitir la comunicación hablada entre seres humanos y computadoras electrónicas. Un sistema de ASR es una herramienta computacional capaz de procesar la señal de voz emitida por el ser humano y reconocer la información contenida en ésta (proveniente de diversas fuentes de conocimiento: acústica, fonética, fonológica, léxica, sintáctica, semántica y pragmática), convirtiéndola en texto o emitiendo órdenes que actúan sobre un proceso.
  Aunque los sistemas de ASR han evolucionado y mejorado mucho sus resultados con el transcurso del tiempo; incluso los mejores sistemas se ven afectados por la presencia de ambigüedades, incertidumbres, ruido ambiental, variaciones del hablante, distorsiones del canal, etc. Por lo que es inevitable para cualquier sistema de reconocimiento del habla cometer errores durante el proceso de identificación, resultando una transcripción ruidosa. Finalmente, los sistemas de ASR y OCR producen formas de ruido similares que dan lugar a la sustitución, borrado e inserción de palabras. Pero por otro lado, los sistemas ASR están restringidos por un lexicón y pueden dar como salidas únicamente las palabras que estén contenidas dentro ese lexicón (Vinciarelli, 2005). Por el contrario, los sistemas OCR pueden trabajar sin necesidad de un lexicón, siendo así posible la transcripción de cualquier cadena de caracteres. Este hecho tiene como aspecto negativo que las secuencias de salida de los sistemas OCR pueden estar formadas por símbolos que no se correspondan a palabras reales. Tales diferencias tienen una fuerte influencia en las propuestas de tratamiento del ruido aplicadas a uno u otro

entorno (o sea, a los textos resultantes de aplicar sistemas OCR y ASR).

- Traducción Automática: en inglés *Machine Translation* (MT). Es un área de la lingüística computacional que investiga el uso de software para traducir texto o conversaciones de un lenguaje natural a otro. En un nivel básico, la traducción por computadora realiza una sustitución simple de cada palabra de un lenguaje natural por las de otro. Por medio del uso de corpus lingüísticos se pueden realizar traducciones más complejas, lo que permite un manejo más apropiado de las diferencias en la tipología lingüística, el reconocimiento de frases, la traducción de expresiones idiomáticas y el aislamiento de anomalías. Cada sistema de MT puede generar diferentes traducciones en dependencia del tipo de modelo empleado y la calidad de los datos de entrenamiento usados. Una buena traducción será aquella que produzca oraciones que mantengan el sentido semántico y sean sintácticamente correctas.
  El ruido en la salida de los sistemas de MT depende de la calidad del corpus de entrenamiento; fundamentalmente en los corpus paralelos ya que juegan un papel importante en la construcción de este tipo de sistemas. Por tanto, el ruido en los corpus paralelos puede resultar un modelo de entrenamiento incorrecto afectando así el comportamiento del sistema. Esta forma de ruido puede ser debido a los errores de traducción como: incorrecto alineamiento entre los corpus, traducciones incorrectas, o sustituciones literales. Además, puede introducirse ruido por otros motivos como errores ortográficos, gramaticales o puntuación incorrecta. Esto último ocurre, sobre todo, cuando los corpus se construyen automáticamente, por ejemplo usando herramientas de minería Web.

Errores ortográficos en la producción de textos digitales en entornos informales Otra vía de introducir ruido en los textos es incluso cuando los mismos se producen de forma digital, particularmente en entornos informales e intrínsecamente ruidosos. Hoy en día existen numerosas formas de comunicación informal en Internet que son propensas a contener errores ortográficos,

caracteres especiales, palabras en un lenguaje no estandarizado, errores gramaticales, palabras multilingües en un mismo texto, etc. Destacar que la mayoría de los errores ortográficos no son intencionales aunque también se pueden observar abreviaturas y distorsiones intencionales. Algunos ejemplos de comunicaciones informales son:

- Correo electrónico: en inglés *e-mail*. Es un servicio de red que permite a los usuarios enviar y recibir mensajes rápidamente mediante sistemas de comunicación electrónicos. Principalmente se usa este nombre para denominar al sistema que provee este servicio en Internet, mediante el protocolo SMTP, aunque por extensión también puede verse aplicado a sistemas análogos que usen otras tecnologías. Por medio de mensajes de correo electrónico se puede enviar, no solamente texto, sino todo tipo de documentos digitales.
- *Online Chat*: término proveniente del inglés que en español equivale a charla o cibercharla, designa una comunicación escrita realizada de manera instantánea a través de Internet entre dos o más personas ya sea de manera pública o privada.
- Foros: en inglés *message boards*, son los descendientes modernos de los Sistemas de Tablón de Anuncios o *Bulletin Board System* muy populares en los años 1980 y 1990. En Internet también se conoce como foro de mensajes, opinión o discusión; y es una aplicación web que da soporte a discusiones u opiniones en línea.
- Grupos de noticias: en inglés *newsgroups*. Son un medio de comunicación dentro del sistema Usenet[2] en el cual los usuarios leen y envían mensajes textuales a distintos tablones distribuidos entre servidores con la posibilidad de enviar y contestar a los mensajes. El sistema es técnicamente distinto, pero funciona de forma similar a los grupos de discusión o foros de la *World Wide Web*.

[2] Usenet es el acrónimo de *Users Network* (Red de usuarios), consistente en un sistema global de discusión en Internet, que evoluciona de las redes UUCP (acrónimo del inglés *Unix to Unix CoPy*). Los usuarios pueden leer o enviar mensajes a distintos grupos de noticias ordenados de forma jerárquica. El medio se sostiene gracias a un gran número de servidores distribuidos y actualizados mundialmente, que guardan y transmiten los mensajes.

- *Blog*: en español bitácora, es un sitio web actualizado periódicamente que recopila cronológicamente textos o artículos de uno o varios autores, apareciendo primero el más reciente, donde el autor conserva siempre la libertad de dejar publicado lo que crea pertinente.
- Wikis: proveniente del hawaiano wiki que significa "hacer las cosas de forma sencilla y rápida". Es un sitio web cuyas páginas pueden ser editadas por múltiples voluntarios a través del navegador web. Los usuarios pueden crear, modificar o borrar un mismo texto que comparten.
- Páginas Web: conocida también como una página de Internet, es un documento electrónico adaptado para la Web. Está compuesta principalmente por información (p.e. texto o módulos multimedia) así como por hiperenlaces o hipervínculos; además puede contener o asociar datos de estilo para especificar cómo debe visualizarse, y también aplicaciones embebidas para hacerla interactiva.
- Servicio de mensajes cortos: o SMS que en inglés es acrónimo de *Short Message Service*, es un sistema de mensajes de texto para teléfonos móviles. En este caso la mayor parte del ruido que se produce está dada por el cambio intencional de la forma de las palabras: empleando abreviaturas, eliminando caracteres (p.e. vocales, caracteres repetidos, etc.) y palabras (p.e. los artículos y pronombres personales), sustituyendo expresiones fonéticamente similares y usando palabras informales o dialectos. De esta forma se superan las limitaciones de los medios de entrada de los dispositivos móviles (p.e. el tamaño pequeño del teclado) y se reducen la duración y el costo de la comunicación a través del mensaje. Otro detalle que provoca la introducción de mucho ruido en los SMS, es la utilización de las letras y signos de puntuación para expresar emociones, efectos verbales o de actitud (risa, énfasis, tristeza, etc.).

Es válido aclarar que en determinados casos un texto puede ser considerado ruidoso para su procesamiento automático por las computadoras pero no para su uso por parte de un humano y viceversa. Por ejemplo, el lenguaje usado en los SMS no es

considerado ruidoso por los humanos, sin embargo los caracteres y las palabras usadas a través de este medio de comunicación difieren mucho de un lenguaje estándar (p.e. *salu2*=saludos, *mñna*=mañana, *xq*=porque) y por tanto, sí es considerado como ruido en su tratamiento automático por las computadoras. Por otro lado, determinados textos producidos de forma automática empleando herramientas de ASR o MT, pueden considerarse perfectos para su utilización por la máquinas, ya que todas las palabras transcritas son palabras de un vocabulario válido pero pueden carecer de sentido para el humano.

De todo lo expuesto se puede apreciar una tendencia al incremento de la presencia de ruido en los textos, ya sea por el crecimiento acelerado de textos digitalizados de forma automática o de textos que tienen su origen en entornos digitales informales. Por tanto, es de gran importancia la investigación en este sentido y queda suficientemente justificada la necesidad de nuestro trabajo, sobre todo en los dominios donde mayor impacto negativo tiene la presencia del ruido, como en los dominios restringidos.

3.1.2. Aplicaciones de PLN afectadas por el ruido textual

Hasta la actualidad muchas aplicaciones y técnicas de PLN basan su arquitectura en el uso de textos libres de ruido (aclaremos que le llamaremos a partir de ahora textos limpios). Pero recientemente se ha observado un impulso en el desarrollo de estrategias que permitan la adaptación de estas técnicas y aplicaciones para que puedan manejar textos ruidosos. Este hecho se debe al continuo crecimiento de textos afectados por el ruido. A continuación hacemos un repaso de algunas de estas aplicaciones y determinadas propuestas para manejar el ruido textual.

Clasificación de textos La tarea de clasificación de textos (en inglés, *Text classification/categorization*) se encarga de asignar a un texto o documento electrónico una o más categorías basándose en su contenido. La propuesta predominante para solucionar este problema está basada en técnicas de aprendizaje automático,

donde un proceso inductivo construye automáticamente un clasificador mediante el aprendizaje de las características de las categorías, a partir de un conjunto de documentos pre-clasificados (Sebastiani, 2002). Existen dos aproximaciones fundamentales para llevar a cabo la clasificación de textos: supervisada, donde algunos mecanismos externos (como la retroalimentación o *feedback* por parte del hombre) proveen información para la correcta clasificación de los documentos; y no supervisada, donde la clasificación debe realizarse completamente sin referencia a informaciones externas. Hay también clasificación semi-supervisada, donde parte de los documentos son etiquetados por mecanismos externos. La mayoría de las aproximaciones están basadas en modelos de aprendizaje y emplean técnicas estadísticas o basadas en reglas.

El problema principal en la aplicación de herramientas de clasificación a textos ruidosos es que las aproximaciones más efectivas, desarrolladas hasta ahora, emplean algoritmos que necesitan entrenarse (p.e. máquinas de soporte vectorial o modelos de redes neuronales artificiales como perceptrón multicapa). Por tanto, si el entrenamiento se realiza sobre documentos con ruido, el rendimiento de cualquiera de esos algoritmos se verá negativamente afectado por la presencia de muchas características irrelevantes (p.e. aquellas introducidas por errores de reconocimiento en los vectores de representación de los textos) en sus entradas (Weston *et al.*, 2000). Por último, destacar la importancia de la clasificación de textos ruidosos en aplicaciones prácticas como: clasificación de hojas de reclamaciones manuscritas de clientes, asignación de ruta automática de SMS, etc. En (Agarwal *et al.*, 2007) se puede consultar un estudio sobre el efecto de diferentes tipos de ruido en la clasificación automática de textos.

Según (Vinciarelli, 2005) una posible solución para este problema es entrenar los modelos de clasificación sobre textos digitales limpios y luego aplicar esos modelos a los textos ruidosos. El desajuste entre las condiciones de entrenamiento y prueba, debido a la presencia de ruido, provocarán la disminución del rendimiento del sistema de clasificación; pero si la pérdida es aceptable, la solución planteada puede dar un resultado equilibrado entre la actuación de la clasificación y el esfuerzo experimental requerido

para llevarla a cabo. Sin embargo, esta propuesta tiene como punto negativo la necesidad de conseguir un corpus de entrenamiento textual limpio, ya que –excluyendo las aplicaciones en dominios abiertos– es difícil su adquisición en dominios más restringidos. Por tanto, la efectividad de la propuesta depende de la disponibilidad de textos libres de ruido, que cubran todas las categorías que se tomarán en cuenta en la clasificación.

Otra propuesta en el área de clasificación textual, se puede ver en (Roy & Subramaniam, 2006), donde se presenta una técnica no supervisada para generar automáticamente una taxonomía de temas (vistas como modelos del dominio) a partir de textos altamente ruidosos y redundantes. Los textos empleados son salidas de aplicar herramientas ASR a conversaciones telefónicas, entre clientes y agentes de un centro de atención al cliente, y son redundantes porque la mayoría de los clientes tienen dudas similares. Precisamente, en la redundancia tiene su base la propuesta que realizan para obtener información útil entre los textos ruidosos y así, crear la taxonomía que necesitan para realizar luego la clasificación de nuevas conversaciones telefónicas. Por tanto, la propuesta necesita de la redundancia de información para ser eficiente y este detalle no siempre se cumple (por ejemplo, en textos de dominios restringidos es poco probable encontrar redundancia de información).

Generación de resúmenes Según la definición dada en (Jones, 2007), el objetivo de la tarea de generación de resúmenes es obtener una versión reducida del documento o documentos fuente, reduciendo su contenido de tal forma que se seleccionen y queden presentes en el resumen los conceptos más importantes de dichos documentos. Por lo tanto, de la definición de esta tarea se deduce que un resumen debe contener la información más significativa de uno o varios documentos, teniendo un tamaño considerablemente inferior al del documento(s) fuente. Los sistemas de generación de resúmenes se pueden clasificar en base a múltiples factores. Una de las taxonomías más conocidas en este campo es la propuesta en (Jones, 1999), en la que se distinguen tres tipos de factores que pueden influir en los resúmenes: factores relacionados con la entrada (en inglés, *input factors*), con la salida (*output factors*),

y con la finalidad del resumen (*purpose factors*). Los factores relacionados con la entrada tratan aspectos como el género, idioma o registro. La finalidad del resumen depende de a quién vaya dirigido y del uso que se le quiera dar, ya que no es lo mismo generar un resumen para informar acerca de los últimos datos de la bolsa, que sobre los hechos históricos más importantes acaecidos en un país, por poner un ejemplo. Finalmente, la salida del resumen en sí, viene determinada por la finalidad y el objetivo que se persiga con éste.

La presencia de ruido en los textos plantea, al igual que en otras aplicaciones de PLN, retos importantes en el proceso de generación de resúmenes. En (Jing *et al.*, 2003; Lopresti *et al.*, 2009) se realizó un estudio sobre la generación de resúmenes a partir de documentos que contuviesen errores, como resultado de la aplicación de herramientas OCR. Dicho trabajo mostró que las técnicas usadas para la generación de resúmenes usando textos limpios no daban buenos resultados sobre textos ruidosos; obteniendo resultados con un importante nivel de degradación incluso con un escaso incremento del nivel de ruido en el documento. El ruido introducido provoca errores en la detección de los límites de las oraciones y por tanto, en su correcta separación (en inglés, *tokenization*); siendo estos errores los causantes del mayor daño ya que afectan al resto de los procesos (p.e. el preprocesamiento de los textos empleando etiquetadores léxicos y sintácticos). Por último, en este trabajo se proponen algunas soluciones de manera teórica como: definir una granularidad apropiada teniendo en cuenta los niveles de ruido. Si son altos estos niveles afirman que es más acertado renunciar a la extracción de las oraciones y en su lugar favorecer la extracción de palabras claves o frases nominales, o generar resúmenes a partir de los estilos de encabezados. Luego proponen la aplicación de técnicas de corrección de errores (i.e. errores ortográficos y errores en la salida de los OCR) a las palabras claves y frases extraídas.

Por último, en (Che, 1998) se propone una alternativa interesante para la generación de resúmenes basada en textos sin la necesidad de herramientas de OCR. En su lugar, ellos extraen las sentencias indicativas para el resumen usando técnicas pura-

mente basadas en imágenes y en las convenciones más comunes de maquetación de documentos. Esta propuesta es efectiva cuando se desea mostrar el resumen al usuario para su lectura; pero en el momento en que se haga necesaria su utilización, edición o recuperación por otras aplicaciones deja de ser útil.

Extracción de información La Extracción de Información (EI, en inglés, *Information Extraction*) es un tipo de recuperación de información cuyo objetivo es extraer automáticamente información estructurada o semi-estructurada desde documentos legibles por una computadora. Una aplicación típica puede ser escanear una serie de documentos escritos en una lengua natural y rellenar una base de datos con la información extraída. Las tendencias actuales en relación con la EI utilizan técnicas de PLN que se centran en áreas muy restringidas. Por ejemplo, la *Message Understanding Conference* (MUC) es una competición que se ha centrado en los siguientes aspectos durante los últimos años: mensajes para operaciones navales (1987-1989), terrorismo en países latinoamericanos (1991), microelectrónica (1993), nuevos artículos acerca de los cambios en la gerencia (1995), informes de lanzamiento de satélites (1998), etc. Entre las tareas típicas que tiene la extracción de información están:

- Reconocimiento de entidades nombradas (NER, por sus siglas en inglés *Named Entity Recognition*): intenta localizar y clasificar elementos atómicos en el texto sobre categorías predefinidas como nombres de personas, organizaciones, lugares, expresiones temporales (hora, fecha, etc.), expresiones numéricas (cantidades, valores monetarios, porcentajes), etc.
- Resolución de la correferencia (en inglés, *Coreference*): identifica distintos sintagmas nominales que se refieren al mismo objeto.
- Extracción de terminología: identifica y extrae candidatos a términos de los textos explorados.
- Extracción de relaciones: requiere la detección y clasificación de las referencias a relaciones semánticas (por ejemplo, el número de teléfono de un cliente o la dirección de un cliente).

Las investigaciones actuales sobre las consecuencias del ruido textual en la EI, están enfocados principalmente en la tarea

de NER, por ser una fase indispensable en las propuestas de EI. En (Miller *et al.*, 2000) se realizó un estudio sobre el rendimiento de la extracción de entidades nombradas bajo una variedad de escenarios que incluían tanto salidas de ASR como de OCR, aunque el principal interés de la investigación era sobre ASR. La solución que presentaron para lidiar con el ruido fue entrenar su sistema de NER sobre materiales de entrada limpios y ruidosos. Sin embargo, el rendimiento del sistema fue linealmente degradado en función del índice de errores en las palabras. Otra aproximación se puede apreciar en (Palmer & Ostendorf, 2001), donde se describe un método para mejorar la extracción de entidades nombradas modelando explícitamente los errores de reconocimiento del lenguaje hablado (provocados por herramientas ASR), a través del uso de estadísticas anotadas con medidas de confianza. Por tanto, esta aproximación requiere de un estudio previo y un conocimiento profundo de los tipos de errores que se presentan al aplicar técnicas de reconocimiento del lenguaje hablado.

Búsqueda de Respuestas En la actualidad no existen muchos trabajos en relación a la influencia ni el tratamiento del ruido textual en sistemas de BR. Podemos decir que una de las razones más lógicas puede ser que la mayoría de los sistemas de BR tienen en su arquitectura un proceso de RI, y precisamente en estos sistemas se han estudiado mucho más las consecuencias del ruido textual. Sin embargo podemos encontrar algunas aproximaciones en este sentido, aunque ninguna que integre el tratamiento del ruido textual al estándar de arquitectura de los sistemas de BR actuales.

En (Aunimo *et al.*, 2003) se presenta un sistema de BR que puede manejar datos ruidosos en lenguaje natural, específicamente el tipo de lenguaje usado en mensajes de correo y SMS (*Short Message Service*), donde las preguntas son generalmente cortas, incompletas y con un lenguaje coloquial cargado de muchos errores ortográficos. El sistema maneja esta situación empleando un método vectorial para representar las cadenas de las preguntas y una medida de similitud del coseno para la comparación entre las cadenas. Por cada pregunta que se recibe se genera automáticamente el vector que la representa y se compara con los vectores

previamente calculados de las preguntas almacenadas en la base de datos. La base de datos tiene alrededor de 24,000 pares de pregunta-respuesta existentes. Otro detalle, es que el sistema está enfocado a un área específica ya que los datos están formados por la retroalimentación de los clientes de una compañía (la mayor parte de la información son preguntas acerca de los productos y servicios) y sobre el idioma finlandés. Además, esta propuesta no realiza ningún proceso de extracción de la respuesta dentro del documento recuperado, sino que brinda como resultado una lista de pares de pregunta-respuesta de la base de datos que fueron identificados como las respuestas más adecuadas para la pregunta formulada. Luego le queda un trabajo al agente del centro de soporte técnico, quien debe seleccionar una de esas respuestas devueltas por el sistema, editar su contenido si es necesario o incluso escribir una nueva respuesta si las devueltas por el sistema no son lo suficientemente relevantes y finalmente, enviarla al cliente. Este trabajo estaba enfocado principalmente en evaluar los diferentes métodos de creación de los vectores, obteniendo los mejores resultados al usar el lema y la división de las palabras compuestas en sus constituyentes. En ningún momento el trabajo plantea una estrategia para enfrentar el problema del ruido textual dentro de la arquitectura de un sistema de BR estándar.

En este punto se puede concluir que las diferentes estrategias para resolver los problemas generados por el ruido están ceñidas al marco de aplicación. Con esto queremos decir, que las soluciones que se brindan en una determinada tarea del PLN pueden no ser útiles en otra área de investigación con objetivos y características diferentes. Otra aplicación ampliamente afectada por el ruido, es la tarea de Recuperación de Información, a la cual le dedicaremos una sección completa por constituir el eje principal de nuestra investigación en torno a la problemática del ruido textual y su influencia en la Búsqueda de Respuesta en Dominios Restringidos.

3.1.3. Tratamiento del ruido textual en sistemas de RI

La Recuperación de Información (RI) es la tarea de buscar información requerida por los usuarios en enormes repositorios de

documentos o corpus. Los sistemas de RI están basados en la comparación de cadenas de texto entre la pregunta del usuario y el corpus donde se puede encontrar la información buscada. Específicamente, a partir de una pregunta del usuario un sistema de RI retorna una lista de documentos relevantes que pueden contener la respuesta a la pregunta (Baeza-Yates & Ribeiro-Neto, 1999). Por consiguiente, el ruido puede aparecer en (i) la pregunta, ya que los términos en la pregunta pueden escribirse directamente de una manera incorrecta conteniendo errores ortográficos; o (ii) el corpus, puesto que este puede ser automáticamente creado desde la Web, ficheros PDF (*Portable Document Format*) o incluso desde herramientas automáticas como OCR o ASR (como se analizó previamente en la sección 3.1.1).

Los sistemas de RI se usan comúnmente en muchas aplicaciones de PLN (p.e. sistemas de BR, EI, etc.), ya que permiten reducir el espacio de búsqueda, es decir, el número de documentos a procesar. Por tanto, es importante que los sistemas de RI sean capaces de lidiar con el ruido. Seguidamente haremos una revisión de las principales propuestas en este sentido, divididas por el tipo de ruido que tratan (ruido en la pregunta o ruido en el corpus) y por la estrategia que emplean (corrección, filtrado o tolerante al ruido).

Ruido textual en la pregunta El ruido textual en la pregunta tiene lugar debido a los errores que puede introducir un usuario cuando interroga a un sistema de RI, algunos ejemplos son: palabras mal deletreadas, división o fusión de palabras erróneamente, pérdida de la raíz de las palabras, abreviaturas poco comunes sin su extensión, etc.

Una de las estrategias más populares para enfrentar el ruido textual en la pregunta es la corrección ortográfica. En la literatura sobre RI y PLN se pueden encontrar desde hace mucho tiempo investigaciones sobre la corrección ortográfica, por ejemplo, el trabajo presentado en (Kukich, 1992). En este sentido la mayoría de las aproximaciones plantean estrategias similares en la generación y selección del candidato para la corrección. Por el contrario emplean medidas diferentes para determinar la validez del término de búsqueda, es decir, para detectar los errores ortográficos: (i)

una medida muy tradicional es la basada en un lexicón ortográfico predefinido, donde se considera a toda cadena de caracteres que no se encuentre en el lexicón como incorrecta, (ii) otra medida es emplear los registros (en inglés, *logs*) de las preguntas realizadas en la Web para inferir la corrección ortográfica de los términos mal deletreados. A continuación, analizaremos algunos ejemplos de aproximaciones que tratan con el ruido textual presente en las preguntas, a través de técnicas de corrección ortográficas.

Corrección ortográfica Tradicionalmente, una de las técnicas más populares en la detección y corrección de errores ortográficos en documentos textuales es la distancia de Levenshtein (Levenshtein, 1966), también conocida como distancia de edición. Esta medida se utiliza desde hace mucho tiempo, en trabajos como (Kukich, 1992), para lidiar con el ruido. Sin embargo, sus resultados en la corrección de errores en las preguntas Web no han sido muy buenos, debido a que no existe ningún lexicón que cubra el vasto número de términos que se encuentran a través de la Web. Como solución se propone que el modelo de distancia de edición puede aumentarse con la utilización de un modelo del lenguaje (en inglés, LM: *Language Model*) a partir de un corpus de preguntas Web. Este modelo del lenguaje está basado en nociones de similitud en la distribución (en inglés, *distributional similarity*), para medir la posibilidad de que una palabra pueda ser reemplazada por otra, basada en las estadísticas de las palabras co-ocurrentes con ellas (Li *et al.*, 2006). Por otro lado, los registros o *logs* de preguntas pueden servir como un excelente corpus para la estimación de la similitud en la distribución; ya que no sólo constituyen una base de términos actualizados, sino además un repositorio extenso de errores ortográficos. Existen muchas aproximaciones que proponen métodos para usar algún tipo de combinación de la distancia de edición y el modelo del lenguaje construido a partir de los registros de preguntas Web.

La primera aproximación elaborada en este sentido se puede consultar en (Cucerzan & Brill, 2004), y en ella se propuso usar los registros de preguntas para inferir la corrección ortográfica de los términos con errores ortográficos. Específicamente, emplea una técnica de transformaciones iterativas en las cadenas de caracte-

res de la pregunta de entrada para convertirlas en otras cadenas, que se corresponden cada vez más a las preguntas probables de acuerdo con las estadísticas extraídas de los registros de preguntas en la Web. Este primer trabajo reportó una buena cobertura para los términos incorrectamente escritos, aunque no se detallaba la precisión de la clasificación de los términos válidos fuera del vocabulario[3] y ni de las faltas de ortografía.

Otro trabajo en este sentido es el presentado en (Li *et al.*, 2006), donde se propone el uso de una métrica de similitud en la distribución, estimada a partir de los registros de preguntas, para discriminar los errores ortográficos más frecuentes. Pero este método tiene un problema cuando hay escasez de datos, ya que para alcanzar una correcta estimación en las métricas de similitud en la distribución se requiere una cantidad suficiente de ocurrencias de cada error ortográfico posible y los términos válidos. Por tanto, esta propuesta no trabaja bien con los términos de búsqueda fuera del vocabulario y que son raramente usados, ni con los errores ortográficos poco comunes.

Finalmente, para dar solución a este problema en (Chen *et al.*, 2007) se propone un método que usa los resultados de la búsqueda en la Web para mejorar los modelos que existen de corrección ortográfica de la pregunta, basándose únicamente en los registros de las preguntas que contienen abundante información en la Web relacionada con la pregunta y que estén entre los mejores candidatos. La información de los resultados de la búsqueda en la Web que utilizan incluyen el número de páginas emparejadas con la pregunta, la distribución del término en los trozos (en inglés, *snippets*) de la página Web y las direcciones URLs[4]. En este trabajo se estudiaron dos esquemas para usar los resultados devueltos por el motor de búsqueda en la Web. El primero sólo utiliza los

[3] Términos fuera del vocabulario: en inglés, *Out-Of-Vocabulary (OOV) terms*, son los términos o palabras que son legítimas pero que no pertenecen a un lenguaje estándar y por tanto, no están en ningún diccionario. Estos términos están presentes sobre todo en las consultas realizadas en la Web.

[4] URL: de las siglas en inglés *Uniform Resource Locator*, es decir, localizador uniforme de recursos. Es una secuencia de caracteres, de acuerdo a un formato estándar, usada para nombrar recursos en Internet para su localización o identificación, como por ejemplo documentos textuales, imágenes, videos, presentaciones digitales, etc.

indicadores de los resultados retornados para la consulta de entrada, mientras que el segundo también tiene en cuenta los resultados de la búsqueda usando las correcciones potencialmente candidatas. Emplean métodos de aprendizaje automático para integrar las características estadísticas de los resultados obtenidos de la búsqueda en la Web. Vale destacar que esta propuesta es efectiva porque se aplica en un dominio abierto, como es la búsqueda en la Web, y por tanto, la existencia de ruido en el corpus no afectaría pero la situación sería bien diferente si se aplicara a entornos más restringidos donde las consecuencias de emplear información con ruido resultante de la búsqueda podrían ser muy negativas.

Ruido textual en el corpus Se puede concluir del estudio previo que existen numerosos trabajos sobre sistemas de RI y el tratamiento del ruido textual presente en los términos de la pregunta. Un detalle sorprendente es que, de forma contraria, las aproximaciones actuales de RI muestran poco interés en tratar explícitamente con el ruido en el corpus. A primera vista, esta situación parece razonable puesto que un corpus puede ser visto como una cantidad enorme de documentos redundantes en donde la respuesta esperada a una pregunta está a menudo repetida en muchos documentos, con o sin ruido. Por consiguiente, los corpus redundantes evitan que los sistemas de RI sean seriamente afectados por problemas de ruido. Desafortunadamente, esta circunstancia es sólo cierta para los corpus de dominio abierto, ya que los corpus generados a partir de dominios restringidos son usualmente más pequeños y por tanto, con poca o ninguna redundancia (Minock, 2005). Los sistemas de RI sobre corpus no redundantes pueden encontrar la respuesta en muy pocos documentos, que a su vez pueden contener ruido y provocar que la respuesta nunca sea recuperada. Consecuentemente, este escenario dificulta el uso de sistemas de RI en situaciones de la vida real donde (i) es usado un corpus de dominio restringido, y (ii) la presencia de ruido es inevitable.

También se puede decir que existe menor cantidad de trabajos en el caso del ruido introducido por herramientas OCR. Esto se debe en parte a que trabajos previos como (Taghva *et al.*, 1996) mostraron que un índice moderado de error tenía un pequeño

impacto en la efectividad de las medidas tradicionales de RI, pero esta conclusión estaba atada a ciertas suposiciones: modelo de RI basado en bolsa de palabras, bajo índice de error OCR, y documentos con longitud extensa.

En esta sección analizaremos algunas de las propuestas de la literatura consultada, que tratan el ruido textual presente en el corpus.

Corrección ortográfica Se vio con anterioridad que la investigación en técnicas algorítmicas para detectar y corregir errores ortográficos en los textos tiene una larga historia en la ciencia de la computación. En el contexto de la RI dichas técnicas han sido empleadas en la corrección de errores ortográficos, tanto en las preguntas como en el corpus de un sistema de RI. Algunos ejemplos de corrección ortográfica de grandes colecciones de documentos se pueden ver a continuación.

En (Taghva & Stofsky, 2001) se describe un sistema de corrección ortográfica diseñado específicamente para textos generados con herramientas OCR, que selecciona las palabras candidatas a través del uso de información adquirida desde múltiples fuentes de conocimiento. El sistema está basado en un dispositivo dinámico y estático de alineamiento, en el emparejamiento aproximado de cadenas de caracteres, y el análisis de $n-$gramas[5]. Se emplea un dispositivo estadístico de dos niveles generador de palabra alineadas que se usan para generar alternativas a las palabras incorrectas. El sistema ha sido diseñado para ser usado en largas colecciones de texto homogéneas. Por lo que no es aplicable a dominios restringidos.

En la tarea de Confusión del TREC-5, en inglés TREC *Confusion Track* (Kantor & Voorhees, 2000; Kantor & Voorhees, 1996), también se pueden ver algunas propuestas de corrección ortográfica. El objetivo fundamental de esta tarea era estudiar cómo se afectaba el rendimiento de la RI sobre textos ruidosos o confusos. Para ello se empleó un corpus de documentos obtenidos de

[5] Un n-grama es una subsecuencia de n elementos de una secuencia dada. Se emplean en varias áreas del procesamiento estadístico del lenguaje natural, así como en algunos métodos de predicción o descubrimiento de genes. Según el valor de n se denominan: "bigrama" con $n = 2$, "trigrama" con $n = 3$, "$n-$grama" o "modelo de Márkov de orden $(n-1)$" con $n \geq 4$.

aplicar herramientas OCR, y se definieron 49 consultas (cada consulta estaba relacionada con un único artículo conocido); por cada consulta se debía recuperar solamente un documento específico, garantizando que ese documento era el único que contenía la respuesta a la consulta. Un ejemplo de corrección ortográfica de los documentos en esta tarea fue el presentado en (Tong *et al.*, 1996), donde se aplican métodos estocásticos[6] a los documentos para corregir las palabras corrompidas oración por oración. Las correcciones son aplicadas a aquellas palabras que no emparejan de manera exacta con ninguna entrada del lexicón que emplean (a estas palabras le llamaron *c-words*). Por cada *c-word* obtienen 200 correcciones candidatas ordenadas según su probabilidad de emparejar con la palabra errónea. Luego conservan las 10 primeras candidatas de cada *c-word* para el procesamiento de las oraciones, que llevan acabo usando el algoritmo de Viterbi (Vit, n.d.) para la obtención de la secuencia de palabras más parecida para la oración. Los resultados que alcanzaron no fueron muy alentadores, ya que el rendimiento del sistema disminuyó en un 51 % debido a la presencia de ruido.

Filtrado del ruido El filtrado del ruido textual es una de las estrategias utilizadas para lidiar con corpus ruidosos, especialmente en los corpus bilingües. La presencia de ruido en estos corpus se debe a que son creados usando métodos automáticos o semiautomáticos. Vale destacar que los corpus bilingües son muy importantes para el entrenamiento de sistemas de RI bilingües o de sistemas estadísticos de Traducción Automática (TA). Por tanto, analizaremos algunas propuestas de filtrado existentes para disminuir el impacto negativo de los pares de oraciones ruidosas en la RI y la TA bilingües. Por ejemplo, en (J-Y.Nie & Cai, 2001; Shi & Nie, 2006) eliminan los documentos paralelos donde: (i) difiere el tamaño de sus respectivos ficheros, o (ii) un número relativamente

[6] Un modelo estocástico (Gar, n.d.) contiene algunos elementos aleatorios o distribuciones de probabilidad dentro del modelo, los cuales permiten introducir elementos de incertidumbre en el comportamiento del sistema; y así, no sólo predice el valor esperado de una cantidad, sino también su varianza. Un fenómeno estocástico es aquel que depende del azar. La inclusión del concepto azar o aleatorización en los modelos se puede efectuar, entre otras, mediante técnicas basadas en los métodos de Monte Carlo y mediante la teoría de las cadenas de Markov.

grande de alineamientos entre las oraciones aparecen vacíos. Este propuesta tiene en cuenta otros parámetros, evaluados en cada par de oraciones, como: similitud de sus longitudes y existencia de entradas de un diccionario bilingüe.

Otra aproximación de filtrado de ruido se puede consultar en (Imamura & Sumita, 2002), donde los autores hacen uso de un criterio de traducción literal (i.e. palabra por palabra) en cada par de oraciones para filtrar un corpus bilingüe ruidoso en los idiomas inglés-japonés. En este trabajo miden la literalidad de la traducción entre la sentencia original y la destino haciendo referencia a un diccionario bilingüe y contando el número de veces que las entradas de ese diccionario ocurren sólo en la oración original, sólo en la oración de destino, y en ambas oraciones. Por último en (Khadivi & Ney, 2005), se propone un esquema de filtrado, para eliminar los pares de oraciones ruidosas del corpus, compuesto por dos métodos: (i) uno basado en la longitud donde se usan restricciones para la longitud entre los pares de oraciones, y (ii) otro basado en la probabilidad de la traducción haciendo uso de modelos para el alineamiento de las palabras, específicamente, alineando los párrafos del documento y alineando las palabras dentro de dos párrafos alineados. Este trabajo alcanza buenos resultados en función del alineamiento de las oraciones y de la calidad final de la traducción.

A partir del análisis de un conjunto de trabajos sobre filtrado del ruido, podemos concluir que la mayoría utiliza criterios semejantes para la detección de las oraciones que presentan ruido en ambos corpus –de origen y destino– como son los métodos de alineamiento de palabras, la utilización de diccionarios bilingües y las longitudes de las oraciones. Pero sobre todo la propuesta de tratar el ruido textual en el corpus por medio de su filtrado tiene como objetivo principal detectar el ruido y removerlo del corpus. Por tanto, esta alternativa es útil en corpus grandes donde hay redundancia de información y bajos niveles de ruido, de lo contrario se podría perder información única y de gran importancia. Así que se puede concluir que no es una estrategia viable en dominios restringidos.

Tolerante al ruido Llegados a este punto podemos afirmar, al igual que Esser (Esser, 2004b), que en la actualidad es necesario desarrollar algoritmos o métodos tolerantes a los fallos provocados por el ruido textual, que sean totalmente independientes a los algoritmos actuales de búsqueda e indexación de los sistemas de RI. De este forma se haría frente a una situación cada vez más grave y común, como la presencia del ruido textual en todos los ámbitos, sin dejar de lado todo lo avanzado en el campo de la RI. Analizaremos algunas aproximaciones que siguen esta idea, destacando sus ventajas, desventajas y marco de aplicación.

En (Esser, 2004a; Esser, 2004b) se propone una interfaz tolerante a fallos para desarrollar la recuperación de documentos sobre corpus de textos muy grandes, como las enciclopedias científicas, con presencia de ruido textual. Para ello en ese trabajo se introduce una técnica de variación de patrones ponderados, en inglés *Weighted Pattern Morphing* (WPM). Este algoritmo (WPM) maneja las similitudes fonéticas, los errores tipográficos comunes como la omisión o transposición de letras, y el uso inconsistente de abreviaturas y guiones. WPM es esencialmente un reemplazamiento recursivo de las sub-cadenas del patrón original de los términos de la pregunta del usuario dirigido por matrices, es decir, la columna del algoritmo es el contenido de las llamadas matrices de pesos para penalizar. Para ello es necesario realizar un estudio previo, definir manualmente las reglas de sub-variaciones a partir de las cuales se generarán las diferentes alternativas de una palabra y crear las matrices con las penalizaciones para cada una de esas reglas. Por tanto, WPM es dependiente del idioma al que se aplica, siendo necesaria su adaptación si se desea emplear en diferentes lenguas. Además del tedioso trabajo que conlleva el estudio previo que se debe realizar para definir manualmente las reglas de sub-variaciones, tiene otra desventaja y es que el algoritmo puede generar variaciones que no pertenezcan al corpus textual que se emplee, por lo que se hace necesario llevar a cabo un filtrado de los mismos. Finalmente, la propuesta es útil para sistemas que interactuan con textos originalmente digitales y que presentan ruido debido a: la introducción de variaciones válidas de una misma palabra por diferentes autores, o la escritura de

las palabras siguiendo su pronunciación en lugar de su forma ortográficamente correcta. Por el contrario, no es una alternativa muy viable para los casos de ruido introducido por herramientas OCR, ni sobre corpus pequeños como es el caso de los dominios restringidos.

En este sentido, algunos eventos han destacado la importancia de la expansión de la pregunta para tratar con el ruido en sistemas de RI, como es la tarea de Confusión del TREC-5 (Kantor & Voorhees, 2000). Ellos defienden que las preguntas pueden expandirse con nuevos términos obtenidos por la adición de errores comunes de corrupción previamente encontrados en el corpus o a partir de listas de pares de palabras correctas e incorrectas. Por ejemplo, en (Ng *et al.*, 1996) expanden los términos que aparecen en la pregunta a través de ventanas deslizantes de 5-gramas con cada caracter reemplazado por cualquier conjunto de 0, 1 ó 2 caracteres (por ejemplo, para la palabra "globo" producen [?lobo, g?obo, gl?bo, glo?o, y glob?] como el conjunto de 5-gramas donde ? denota que no interesa el caracter que esté en esa posición). Los 5-gramas nos cruzan los límites de la palabra. El ordenamiento de la recuperación se basa en el promedio del número de emparejamientos por línea de texto. Esta propuesta alcanzó un rendimiento muy pobre. Otro ejemplo se puede analizar en (Hawking *et al.*, 1996) aquí proponen expandir las preguntas basados en errores de corrupción encontrados en un texto de ejemplo corrompido, asumiendo que los errores son similares. Por tanto, no queda garantizado que se cubran todos los tipos de errores que pueden existir en el corpus. Los resultados alcanzados tampoco fueron muy buenos.

3.2. Taxonomías de tipo de respuesta esperada

El proceso de análisis de la pregunta debe realizarse de la manera más precisa posible ya que es la primera fase de los sistemas de BR y, por tanto, el resto de fases dependen de sus resultados. Una de las tareas más importantes dentro del proceso de análisis de la pregunta es determinar el tipo semántico de la respuesta o

Tipo de Respuesta Esperado (TRE) (Pinchak & Lin, 2006) con el fin de reducir el espacio de búsqueda de posibles respuestas a la vez que se consiguen respuestas más exactas (Li & Roth, 2006; Hovy *et al.*, 2002). Para poder reconocer el TRE de la pregunta se requiere una taxonomía predefinida de TRE[7]. Cabe destacar que, en (Moldovan *et al.*, 2003), se muestra que más del 36,4 % de los errores de un sistema de BR están relacionados con una detección incorrecta del TRE (28,2 % de los errores se deben a un TRE desconocido, no obteniendo ninguna respuesta; mientras que el 8,2 % de los errores se deben a un TRE incorrecto, obteniendo una respuesta incorrecta). Por lo tanto se debe señalar que la mayoría de los errores cuando el TRE se detecta se deben a una especificación incorrecta de la taxonomía de TRE usada en el sistema de BR.

Actualmente, existen muchas propuestas de taxonomías de TRE para sistemas de BR-DA, las cuales se utilizan para un amplio espectro de preguntas (Sekine *et al.*, 2002; Hovy *et al.*, 2002; Metzler & Croft, 2005; Li & Roth, 2006). Estas propuestas tienen en cuenta la partícula interrogativa para clasificar la pregunta en un tipo determinado (por ejemplo, preguntas del tipo Qué, Cuál, Quién, Dónde, Cuándo, etc.), añadiendo posteriormente conocimiento semántico para obtener respuestas más exactas. De entre todos los tipos de preguntas, aquellas que tienen una partícula interrogativa ambigua (Qué o Cuál) son las más difíciles de analizar, ya que pueden relacionarse con cualquier tipo de respuesta (Qué objeto, Qué sustancia, Qué enzima, Qué hidrolasa, etc.); a diferencia de las partículas interrogativas Quién, Cuándo y Dónde que corresponden a los conceptos de persona, fecha y localización, respectivamente. Así pues, cuánto más ambigua sea la pregunta más conocimiento semántico se requiere para especificar una taxonomía de TRE adecuada para el dominio de aplicación del sistema de BR.

[7] Otros términos usados frecuentemente son jerarquía de preguntas (Li & Roth, 2006) u ontología de preguntas (Metzler & Croft, 2005)

Este conocimiento semántico puede provenir de recursos llamados SOC (Sistemas de Organización del Conocimiento)[8]; según las áreas de conocimiento a las que se refiera un SOC, existen dos tipos: SOC genérico (como WordNet[9], EuroWordNet[10], SUMO[11], etc.) o los más precisos SOC de dominio (tales como el tesauro Agrovoc[12] para el dominio agrícola, el metatesauro UMLS [13] para el dominio médico, etc.).

La taxonomías empleadas en sistemas de BR-DA pueden llamarse taxonomías de TRE de dominio abierto, ya que están diseñadas fuera de un dominio específico. La mayoría de estas taxonomías se crean manualmente a partir del estudio de las colecciones de preguntas de los foros TREC y CLEF; y empleando a WordNet como SOC genérico. Algunos ejemplos de las clases que suelen contener dichas taxonomías son: persona, objeto, lugar, numérico, persona, etc. Una taxonomía de este tipo tendrá un uso limitado en un dominio restringido, ya que existe una mínima probabilidad de que esté bien balanceada con relación al dominio seleccionado. Por un lado, en los dominios restringidos, como los dominios técnicos, abundan los términos que son específicos para el dominio y son muy desconocidos en otros dominios (p.e., hidrolasa y esterasa en el dominio agrícola). Y por otro lado, en el dominio abierto un término puede tener múltiples sentidos, incluyendo sentidos incorrectos para un dominio determinado, y sin embargo estos términos están usualmente desambiguados en el dominio restringido (p.e. el término "agentes" tiene en WordNet seis sentidos diferentes entre ellos "agente federal", el cual carece de sentido en un dominio agrícola donde sólo es válido con el sentido de una "sustancia que ejerce alguna fuerza o el efecto". Por tanto para adaptar un sistema de BR-DA a un dominio restringido, y así obtener un sistema de BR-DR, se hace necesario

[8] Los Sistemas de Organización del Conocimiento incluyen una variedad de esquemas que organizan, gestionan y recuperan información. Este término pretende abarcar cualquier tipo de esquema que sirva para gestionar el conocimiento (Hodge, 2000), como diccionarios, taxonomías, tesauros, ontologías, etc.

[9] http://wordnet.princeton.edu/

[10] http://www.illc.uva.nl/EuroWordNet/

[11] http://www.ontologyportal.org/

[12] http://www.fao.org/agrovoc/

[13] http://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/

ajustar la taxonomía de TRE que emplee al dominio restringido, fundamentalmente las preguntas mas difíciles de analizar por su ambigüedad como "*¿Qué esterasa incrementa la digestibilidad del fósforo orgánico en los animales?*".

Las principales características que definen una taxonomía de TRE son el tamaño, la estructura y la cobertura (Tomás & González, 2007). El tamaño se refiere al número de clases de la taxonomía. La estructura puede ser plana o jerárquica. La cobertura puede ser ideal si la taxonomía de TRE cubre la mayoría de las preguntas del dominio sin considerar si el sistema de BR será capaz de contestarlas, haciendo decrecer la precisión. Por otro lado, la cobertura puede ser realista si las clases de la taxonomía de TRE se definen según los tipos de respuesta que el sistema de BR es capaz de localizar y extraer, lo que hace que la precisión se incremente aunque la cobertura sea menor. El cuadro 3.1 resume algunas de las taxonomías de TRE existentes en la actualidad, considerando estas características.

Cuadro 3.1. Resumen de las principales propuestas de taxonomías de TRE.

	Dominio	**Tam.** (# classes)			**Estructura**		**Cobertura**	
Trabajo en:		Peq.	Med.	Gran.	Plana	Jerárquica	Ideal	Realista
(Metzler & Croft, 2005)	Abierto		31		x			x
(Li & Roth, 2006)	Abierto			50		x	x	
(Sekine *et al.*, 2002)	Abierto			150		x	x	
(Hovy *et al.*, 2002)	Abierto			180		x	x	
(Ely *et al.*, 2000)	Médico	10			x			x
(Sang *et al.*, 2005)	Médico	7			x			x
(Ferrés & Rodríguez, 2006)	Geográfico	25				x		x

Cada taxonomía de TRE resumida en la tabla 3.1 se usa en sistemas de BR, ya sea para dominio abierto o dominio restringido. Las taxonomías de TRE para dominio abierto (Metzler & Croft, 2005; Li & Roth, 2006; Sekine *et al.*, 2002; Hovy *et al.*, 2002)

se desarrollan de manera manual a partir de grandes colecciones de preguntas (obtenidas de la Web[14] o de los congresos TREC o CLEF) mediante la obtención de conocimiento de WordNet. A pesar de ser jerárquicas, no están suficientemente refinadas para ser utilizadas en dominios restringidos. Sin embargo, las taxonomías de TRE para dominio restringido también presentan problemas. Por ejemplo, las taxonomías de TRE definidas en (Ely *et al.*, 2000) y (Sang *et al.*, 2005) se obtuvieron, respectivamente, de una colección de 1001 preguntas realizadas por médicos y de 435 preguntas a partir del corpus RSI (*Repetitive Strain Injury*). Ambos trabajos, desarrollan la taxonomía de TRE de manera manual mediante el uso del metatesauro UMLS como SOC del dominio. En (Ferrés & Rodríguez, 2006) se presenta cómo una taxonomía de TRE para dominio abierto se puede adaptar a dominio restringido mediante el uso de conceptos y relaciones de una ontología de dominio. Una desventaja de estas propuestas es que la taxonomía de TRE se basa en analizar preguntas potenciales de usuarios, lo que puede no ser posible en aplicaciones reales, ya que la adquisición de un gran número de preguntas de dominio restringido es muy costosa.

Se debe resaltar que los sistemas de BR-DA tienen una cobertura ideal porque no garantizan que el sistema de BR pueda detectar el TRE, mientras que los sistemas de BR-DR intentan tener una cobertura realista para obtener una buena precisión mediante la inclusión en la taxonomía de aquellos TREs que el sistema es capaz de responder.

Teniendo en consideración lo expuesto anteriormente, es importante definir mecanismos para mejorar el diseño de las taxonomías de TRE mediante el uso de los recursos del conocimiento disponibles, incrementando así la precisión de los sistemas de BR-DR. Para este fin, en esta tesis se propone el uso del desarrollo dirigido por modelos para la creación de taxonomías de TRE de manera automática a partir de varios SOC mediante el uso de la colección de documentos en lugar del corpus de preguntas.

[14] AskJeeves: `http://www.ask.com`; Yahoo! Answers: `http://answer.yahoo.com`

3.3. Adaptación de patrones para sistemas de Búsqueda de Respuestas

En la actualidad, en los sistemas de BR-DA el análisis de la pregunta y la extracción de la respuesta se realizan principalmente mediante el uso de conocimiento lingüístico previamente adquirido y codificado en forma de patrones por un experto de manera manual. Entendiéndose por patrones todas aquellas estrategias seguidas para codificar las restricciones que deben cumplir la pregunta o respuesta para ser clasificadas o encontradas, respectivamente. Para nuestro trabajo consideraremos patrones a todas aquellas relaciones secuenciales de expresiones, es decir, cada expresión tendrá antecedente, consecuente o ambos. Ejemplos de tipos de patrones serían restricciones sintácticas, formas lógicas, expresiones regulares, etc.

Por lo tanto, la *adaptabilidad* de los patrones mediante el uso de conocimiento de un dominio específico es clave para la derivación de sistemas de BR-DR a partir de sistemas de BR-DA. Además, el diseño de sistemas de BR-DR sufre de otros problemas: (i) los patrones se crean para un sistema de BR específico (Peñas *et al.*, 2009), por lo que se debe tener un amplio conocimiento del código del sistema de BR, lo que dificulta la tarea de adaptar los patrones para un nuevo dominio, impidiendo la *reusabilidad* y *portabilidad* de patrones de sistemas de BR de otros dominios o de sistemas de BR-DA. (ii) El proceso de crear nuevos patrones para un dominio específico de manera manual (Peñas *et al.*, 2009; Roger *et al.*, 2008), consume mucho tiempo y es propenso a errores, lo que afecta a la *productividad*. (iii) Los patrones se adaptan mediante el uso de diferentes tipos de recursos de conocimiento o SOC de un dominio específico, teniendo que enfrentar el problema de la diversidad de sus formatos e interfaces de acceso.

En la actualidad el proceso de adaptación de patrones (patrones de preguntas y respuestas) de BR-DA existentes a dominios específicos se realiza manualmente usando recursos lingüísticos (Peñas *et al.*, 2009; Hermjakob, 2001; Harabagiu *et al.*, 2000; Roger *et al.*, 2008), siendo así costosos y propensos a fallos. Otras aproximaciones analizan preguntas potenciales para ser contesta-

das (Ravichandran & Hovy, 2002; Kosseim & Yousefi, 2008), lo cual es factible para dominios abiertos, donde los repositorios de preguntas pueden ser fácilmente adquiridos del CLEF, el TREC o desde la Web[5], pero de difícil aplicación en dominios restringidos puesto que es poco probable encontrar corpus de entrenamientos lo suficientemente exhaustivos (Mollá & González, 2007).

Nuestro punto de vista es que los patrones deben ser afinados usando diferentes tipos de recursos del conocimiento de dominio específico. Estas fuentes de conocimiento son también conocidas como Sistemas de Organización del Conocimiento (SOC) como se explicaba con anterioridad en este mismo capítulo. No obstante, estos SOC tienen sus propios formatos e interfaces de acceso, con lo cual la tarea de unificación de los mismos por el sistema de BR se hace extremadamente costosa (Mollá & González, 2007; Ferrés & Rodríguez, 2006; Nyberg *et al.*, 2005).

Para superar estos problemas, en el marco de esta tesis se define una aproximación dirigida por modelos para adaptar automáticamente los patrones de preguntas y respuestas de un sistema de BR-DA para un dominio restringido a partir de la colección de documentos y teniendo en cuenta los SOC disponibles.

3.4. Conclusiones

Resumiendo, todas estas aproximaciones dependen de un conocimiento profundo acerca de cómo aparece el ruido en el corpus para detectar patrones de errores ortográficos o tipográficos que ayuden al tratamiento del ruido. Por consiguiente, las aproximaciones actuales no son apropiadas para corpus de dominios restringidos ya que al ser pequeños cada tipo de error puede ocurrir en muy pocas ocasiones e incluso una sola vez, haciendo así muy difícil la detección de patrones de errores.

Nuestra aproximación supera este inconveniente desde que tomamos como ventaja la alta disponibilidad de Sistemas de Organización del Conocimiento (SOC) para los dominios restringidos

[5] AskJeeves: http://www.ask.com; Yahoo! Answers: http://answer.yahoo.com

y su utilidad para comparar los términos desde el corpus con ruido y los términos importantes del dominio, de tal manera que no es necesario un análisis previo de los errores que ocurren en los términos del dominio restringido en el corpus.

Finalmente, vale resaltar que aunque el impacto negativo del ruido ha atraído últimamente la atención de los investigadores (Lopresti *et al.*, 2009), sorprendentemente el tratamiento del ruido en corpus de dominio restringido no ha sido ampliamente considerado hasta el momento.

Parte III

Propuesta de Adaptación de Búsqueda de Respuestas a Dominios Restringidos

Capítulo 4

Estrategia de tolerancia al ruido textual para sistemas de recuperación de información en dominios restringidos

Uno de los primeros pasos en la adaptación y aplicación de un sistema de Búsqueda de Respuestas (BR) en un entorno restringido es la elaboración de los recursos del dominio, por ejemplo el propio corpus. El corpus puede estar formado por documentos de texto que provienen de fuentes heterogéneas, como sitios Web, ficheros PDF (*Portable Document Format*), o a partir de sistemas de reconocimiento óptico de caracteres (OCR: *Optical Character Recognition*) y sistemas de reconocimiento automático del habla (ASR: *Automatic Speech Recognition*). Alcanzar una conversión fiel de estas fuentes de datos a ficheros de texto plano no es una tarea fácil, dado que se puede introducir ruido a partir de errores ortográficos o tipográficos (*typeset*). Por otro lado, si el tamaño del corpus es lo suficientemente grande, como es el caso de los corpus de dominio abierto, la redundancia de información presente ayuda a controlar los efectos del ruido porque un mismo texto puede aparecer con o sin ruido a través del corpus. En cambio, el ruido se convierte en un problema serio en los dominios restringidos donde el corpus es usualmente pequeño y presenta poca o ninguna redundancia (Minock, 2005). Por consiguiente el ruido dificulta tareas como la Recuperación de Información (RI) y por consiguiente la Búsqueda de Respuestas (BR) en dominios restringidos, al provocar la generación de respuestas erróneas. A pesar de los avances propuestos, hasta la actualidad, en la resolución de problemas provocados por el ruido textual (ver la sección 3.1 del capítulo 3) aún no se alcanza una resolución óptima de los

mismos, y además no existe ningún estudio aplicado a dominios restringidos.

Teniendo en cuenta estas consideraciones, se presenta una aproximación novedosa en este capítulo con la finalidad de alcanzar un nivel aceptable de tolerancia al ruido en SBR-DR existentes, específicamente en el proceso de RI. La base de nuestra propuesta es el uso de recursos de conocimiento o Sistemas de Organización del Conocimiento (SOC[1]) frecuentemente disponibles, sobre todo en dominios restringidos, como son los tesauros, ontologías, diccionarios, etc. Nuestra hipótesis es que los términos importantes en un corpus de dominio restringido pertenecen a un vocabulario pequeño, específico y controlado que puede ser obtenido desde algún SOC. Por lo tanto, el ruido en el corpus puede ser esquivado por la comparación de los términos del SOC y los términos provenientes del corpus. Para llevar a cabo esta comparación se hace necesario emplear algún algoritmo de comparación o Distancia de Edición entre cadenas de caracteres (por ejemplo, distancia de Leveshtein (Levenshtein, 1966), de Jaro-Winkler (Winkler, 1999), etc.). Por tanto, un primer paso a realizar es la elección del algoritmo que se empleará para calcular la similitud o distancia entre las palabras. Para ello hemos realizado un estudio entre varios algoritmos, destacando ventajas y desventajas, en la primera sección 4.1 de este capítulo. Además de un análisis comparativo (sección 4.2) a partir de los resultados de algunos experimentos realizados, llegando a la elección de un algoritmo como base de nuestra propuesta.

El siguiente paso desarrollado en la sección 4.3 de este capítulo es nuestra propuesta de adaptación del algoritmo de distancia de edición seleccionado para hacer posible la comparación entre los términos, considerando también las multipalabras que comúnmente aparecen en la mayoría de los corpus de dominios restringidos (p.e. en dominios científicos, multipalabras como: “hidróxido de

[1] SOC: Sistemas de Organización del Conocimiento, en inglés *Knowledge Organization Systems*. Son sistemas que incluyen una variedad de esquemas a través de los cuales se organiza, maneja y recupera información. Según Hodge en (Hodge, 2000) el término SOC intenta abarcar todos los tipos de esquemas para promover el manejo de conocimiento.

calcio", "tejidos naturales", "tracto digestivo" u "hormonas suprarrenales").

Finalmente describimos nuestra propuesta, en la sección 4.4, para hacer tolerante a la presencia de ruido textual en el corpus a un sistema de RI cualquiera sobre dominios restringidos. De esta manera cumplimos con uno de los objetivos planteados en este trabajo de investigación: desarrollar una estrategia de tolerancia al ruido textual, que permita la mejora del proceso de RI sobre corpus pequeños y ruidosos.

Nuestra propuesta es totalmente independiente de la arquitectura del sistema de RI y precisamente este detalle constituye uno de sus principales beneficios. Por otro lado, nuestra aproximación logra mantener el rendimiento del sistema de RI aunque utilice un corpus de dominio restringido ruidoso, como se mostrará en diversos experimentos que se llevarán a cabo en el capítulo 7. Estos experimentos muestran la idoneidad de nuestra propuesta en un caso de estudio real, usando un sistema de RI llamado JIRS (Buscaldi *et al.*, 2010) y como dominio restringido las publicaciones de una revista científica agrícola llamada Revista Cubana de Ciencia Agrícola[2] (RCCA).

4.1. Algoritmos de Distancia de Edición

En esta sección definiremos algunas métricas de Distancia de Edición, haciendo énfasis en sus características.

4.1.1. Distancia de Levenshtein

La Distancia de Levenshtein (Levenshtein, 1966) o Distancia de Edición (DE) es una distancia que ofrece como resultado un valor numérico que establece la diferencia entre dos palabras. Este valor numérico se define por el número mínimo de operaciones necesarias para transformar una cadena de caracteres en otra. Entendiéndose por "operaciones" a la inserción, eliminación o sustitución de un caracter. Se le considera una generalización de la

[2] http://www.ica.inf.cu/productos/rcca/

distancia de Hamming (Manthey & Reischuk, 2003), que se usa para cadenas de la misma longitud y que sólo considera como operación la sustitución. A su vez existen otras generalizaciones de la distancia Levenshtein, por ejemplo, la distancia de Damerau-Levenshtein (Damerau, 1964) que considera también el intercambio de dos caracteres o transposición como una operación.

Frecuentemente se utiliza un algoritmo *bottom-up* de Programación Dinámica para calcular la distancia de Levenshtein (Wagner & Fischer, 1974). Este algoritmo emplea una matriz de enteros de $(n+1)*(m+1)$, donde n y m son las longitudes de las cadenas. Finalmente, cuando la matriz está completa, la celda inferior derecha indica el costo mínimo de la transformación para convertir una palabra en la otra. La distancia de Levenshtein está ampliamente difundida y se ha empleado en diversas aplicaciones (por ejemplo, correctores ortográficos, detección de fraude, etc.). Sin embargo, presenta algunas desventajas que explicaremos más adelante en la sección 4.2.

4.1.2. Distancia Needleman-Wunsch

El algoritmo propuesto por Needleman y Wunsch (Needleman & Wunsch, 1970) es una extensión del algoritmo de la DE o distancia Levenshtein, ya que sólo añade en la métrica de la distancia un ajuste variable (identificado por G) para el costo de los fallos, o sea inserción/eliminación. De esta manera la distancia de Levenshtein puede ser vista simplemente como la distancia Needleman-Wunsch con $G = 1$. Este algoritmo sirve para realizar alineamientos globales de dos secuencias o cadenas de caracteres. Se suele utilizar en el ámbito de la bioinformática para alinear secuencias de proteínas o de ácidos nucleicos.

4.1.3. Distancia Jaro-Winkler

La Distancia de Jaro (Jaro, 1989) es una métrica donde dada dos cadenas s_1 y s_2 la distancia d_j entre ellas se calcula según la Ecuación 4.1; donde m es el número de caracteres que emparejan o coinciden y t es el número de transposiciones.

$$d_j(s_1, s_2) = \frac{1}{3} * \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{(m-t)}{m} \right) \tag{4.1}$$

Un extensión de la Distancia de Jaro es la Distancia de Jaro-Winkler creada a partir del trabajo de Winkler en (Winkler, 1999). Por tanto, es una medida de similitud entre dos cadenas de caracteres y se utiliza principalmente en la vinculación de registros (específicamente, en la detección de duplicados). Mientras mayor es la distancia de Jaro-Winkler para dos cadenas de caracteres, mayor similaridad hay entre ellas. La distancia de Jaro-Winkler es una métrica diseñada de tal forma que funciona adecuadamente con cadenas de caracteres cortas, tales como nombres propios de personas. Esta medida propone que dos caracteres desde las cadenas s_1 y s_2 respectivamente, sean considerados como coincidentes sólo si no están más allá de: $\left[\frac{max(|s_1|,|s_2|)}{2}\right] - 1$.

$$d_{jw}(s_1, s_2) = d_j(s_1, s_2) + (l * p(1 - d_j(s_1, s_2))) \tag{4.2}$$

La distancia de Jaro-Winkler (o sea, d_{jw}) se calcula según la Ecuación 4.2, donde d_j es la distancia de Jaro entre las cadenas s_1 y s_2, l es la longitud del prefijo común al inicio de la cadena hasta un máximo de 4 caracteres, y p es un factor de escala constante para favorecer a las cadenas que tienen prefijos comunes. El valor estándar para esta constante es indicado en el trabajo de Winkler como $p = 0{,}1$

4.1.4. Distancia de Edición Extendida

La Distancia de Edición eXtendida (DEx) (Fernández *et al.*, 2009), es una extensión del algoritmo de Distancia de Levenshtein o Distancia de Edición (DE). Específicamente parte del análisis de la matriz de costos usada por DE. Además en DEx se obtienen la subsecuencia común más larga (en inglés, *Longest Common Subsequence* – LCS) (Hirschberg, 1977) y diferentes atributos que ayudan en la determinación de la similitud entre cadenas de caracteres en una sola iteración. También tiene en cuenta otros elementos como la posición donde se realizan las transformaciones,

así como su tipo (p.e., inserción, sustitución o eliminación). Finalmente, toda esta información se utiliza en DEx para realizar diferentes penalizaciones según:

- la posición de las transformaciones: si ocurren en la raíz (o sea, más a la izquierda) de la palabra la penalización será mayor; en contraste con las que ocurren más a la derecha de la palabra,
- el carácter involucrado en la transformación y el tipo de transformación: permite colocar un costo en dependencia del caracter involucrado en un tipo de transformación dado.

Pasos del algoritmo DEx definido en (Fernández *et al.*, 2009):

1. Generar la matriz de Levenshtein con las palabras a analizar. Por ejemplo, para las palabras "afrecho" y "afrechillo", la matriz quedaría como se puede ver en el Cuadro 4.1. Observando la celda inferior derecha de la matriz resultante se puede conocer el costo mínimo (en este caso hay que realizar 3 transformaciones) para convertir la palabra "afrecho" en la palabra "afrechillo".

Cuadro 4.1. Ejemplo paso 1 de la DEx: obtención de la matriz de Levenshtein.

		a	f	r	e	c	h	i	l	l	o
	0	1	2	3	4	5	6	7	8	9	10
a	1	0	1	2	3	4	5	6	7	8	9
f	2	1	0	1	2	3	4	5	6	7	8
r	3	2	1	0	1	2	3	4	5	6	7
e	4	3	2	1	0	1	2	3	4	5	6
c	5	4	3	2	1	0	1	2	3	4	5
h	6	5	4	3	2	1	0	1	2	3	4
o	7	6	5	4	3	2	1	1	2	3	**3**

2. Determinar el camino que corresponda con la subsecuencia común máxima (LCS) según (Hirschberg, 1977). Para obtener el camino buscado se parte de la posición en la casilla inferior derecha, luego se recorre hacia atrás pasando a la celda con el valor mínimo, priorizando siempre la diagonal en caso de tener valores iguales las casillas aledañas. Ver en el Cuadro 4.2 la subsecuencia común máxima, marcada con las celdas en color gris, que se obtuvo para el ejemplo "afrecho"-"afrechillo".

Cuadro 4.2. Ejemplo paso 2 y 3 de la DEx: obtención de la LCS y de la CO.

		a	f	r	e	c	h	i	l	l	o
	0	1	2	3	4	5	6	7	8	9	10
a	1	0	1	2	3	4	5	6	7	8	9
f	2	1	0	1	2	3	4	5	6	7	8
r	3	2	1	0	1	2	3	4	5	6	7
e	4	3	2	1	0	1	2	3	4	5	6
c	5	4	3	2	1	0	1	2	3	4	5
h	6	5	4	3	2	1	0	1	2	3	4
o	7	6	5	4	3	2	1	1	2	3	**3**

O	O	O	O	O	O	I	I	I	O

3. Generar la Cadena de Operaciones (CO) partiendo del camino (LCS) encontrado en el paso anterior. Para ello se interpreta cada movimiento como un tipo de operación:
   - Un movimiento por la vertical es interpretado como una operación de "eliminación" o ***D****elete*, asignándole la letra **D**
   - Un movimiento por la horizontal es interpretado como una operación de "inserción" o ***I****nsertion*, asignándole la letra **I**
   - Un movimiento por la diagonal es interpretado como una operación de "sustitución" o ***S****ubstitution*, asignándole la letra **S**
   - Por último, si la casilla de origen y la de destino tienen el mismo valor, se interpreta como una "NO operación", asignándole la letra **O**.

   Para el caso ejemplo anterior la cadena de operaciones sería: "OOOOOOIIIO", se puede ver al final del Cuadro 4.2.
4. Evaluar la Ecuación 4.3 con la cadena de operaciones encontrada y los caracteres implicados en cada operación.

$$DEx = \sqrt[8]{\frac{\sum_{i=0}^{l-1} V_{(O_i)} * \left(P_{(c1_j)}, P_{(c2_k)}\right) (2R_{max}+1)^{L-i}}{N}} \quad (4.3)$$

Donde:

O : Cadena de operaciones (O-No operación, I-Inserción, D-Eliminación, S-Sustitución).

O_i : Operación en la posición i-ésima.

V : está formalizado como el vector que se muestra a continuación.

$$V = \left\{ \begin{array}{l} (0,0) : o \\ (1,0) : i \\ (0,1) : d \\ (1,1) : s \end{array} \right\}$$

$c1$ y $c2$: palabras o cadenas de caracteres analizadas.

$c1_j$: el j-ésimo carácter de la palabra $c1$.

$c2_k$: el k-ésimo carácter de la palabra $c2$.

P : El peso asignado a cada carácter. El valor de los pesos se obtiene a partir de un cálculo de frecuencia de aparición de cada uno de los caracteres en un diccionario del lenguaje al que pertenezcan las palabras. Luego se ordenan los caracteres y se les coloca un número empezando por 1 hasta la cantidad de caracteres y en orden inverso, como se muestra a continuación.

$$P = \left\{ \begin{array}{llllllll} a:52 & c:44 & g:36 & \acute{\imath}:51 & \acute{u}:41 & 7:13 & / & :5 \\ i:51 & l:43 & b:35 & j:27 & w:20 & 9:12 & \ddot{u} & :41 \\ e:50 & t:42 & y:34 & \acute{a}:52 & 1:19 & 6:11 & \rightarrow & :3 \\ o:49 & u:41 & f:33 &):25 & \tilde{n}:18 & .:10 & = & :3 \\ s:48 & d:40 & v:32 & (:25 & 0:17 & 4:9 & \_ & :3 \\ r:47 & p:39 & \acute{o}:49 & q:24 & 2:16 & 5:8 & ' & :2 \\ n:46 & m:38 & x:30 & k:23 & -:15 & 8:7 & \grave{e} & :50 \\ :45 & h:37 & z:29 & \acute{e}:50 & 3:14 & ,:6 & \backslash & :1 \end{array} \right.$$

$P_{(c1_j)}$: es el peso del carácter en $c1_j$, donde,

$$j = \left\{ \begin{array}{ll} j+1 & \text{si } O_i \neq I \\ j & \text{si } O_i = I \end{array} \right\}$$

$P_{(c2_k)}$: es el peso del carácter en $c2_k$, donde,

$$k = \left\{ \begin{array}{ll} k+1 & \text{si } O_i \neq D \\ k & \text{si } O_i = D \end{array} \right\}$$

L : Longitud de la palabra más larga del lenguaje. Por ejemplo, para el lenguaje español se puede tomar 25 como la longitud de la palabra más larga.

l : Longitud de la cadena de operaciones de edición.

R_{max} : Cantidad de caracteres en P.

λ : Es el producto de las componentes del vector resultante de $V_{(O_i)}$.
N : está definido como sigue, $N = \sum_{i=0}^{L-1}(2R_{max}+1)^i$

Para concluir, en la Ecuación 4.3, se puede observar que el término $V_{(O_i)} * \Big(P_{(c1_j)}, P_{(c2_k)}\Big)$ es el producto cartesiano que analiza la importancia de efectuar la operación $V_{(O_i)}$ entre los caracteres $P_{(c1_j)}$ y $P_{(c2_k)}$. El término $(2R_{max}+1)^{L-i}$ penaliza la posición de la operación, de forma tal que mientras más a la izquierda (o sea, próximo a la raíz de la palabra) se realice la operación, mayor será la penalización. N es el término que normaliza la distancia en el intervalo $[0, 1]$ con el peor caso posible, o sea con una cadena de operaciones de longitud L llena de operaciones de sustitución entre los caracteres más costosos del alfabeto. La raíz octava se aplica buscando que los valores no resulten tan pequeños y que no quede afectada la relación de orden.

Para continuar con el ejemplo, que hemos estado desarrollando, de comparación entre las palabras "afrecho" y "afrechillo" en el Cuadro 4.3 se muestran los valores de todos los argumentos para calcular DEx. Aclaremos que la columna: **A** contiene los valores del producto escalar del vector $V_{(O_i)}$ por $P_{(c1_j)}$ y $P_{(c2_k)}$ (o sea, $V_{(O_i)} * \Big(P_{(c1_j)}, P_{(c2_k)}\Big)$), **B** contiene los valores de $(2R_{max}+1)^{L-i}$ y la última columna es la multiplicación de los valores de las columnas A y B, o sea, **A*B**=$V_{(O_i)} * \Big(P_{(c1_j)}, P_{(c2_k)}\Big)(2R_{max}+1)^{L-i}$. Finalmente, se obtiene que la distancia $DEx = 0,026$ entre "afrecho" y "afrechillo", afirmándose así que son palabras bastante similares.

Como la distancia DEx se evalúa a partir de la cadena de operaciones mínimas y ésta es generada por la aplicación del algoritmo de cálculo de la subsecuencia común máxima (LCS) en la matriz de programación dinámica para la DEx, el orden del algoritmo DEx es igual al algoritmo DE en $(O(m, n))$, donde m y n son las longitudes de las cadenas comparadas.

Cuadro 4.3. Ejemplo paso 4 de la DEx: evaluación de la Ecuación 4.3 con la CO y los caracteres implicados.

O_i	$V_{(O_i)}$	c1	P_{c1}	c2	P_{c2}	A	i	$L-i$	B	$A*B$
O	$(0,0)$	a	52	a	52	0	0	24	$1,12E+02$	0
O	$(0,0)$	f	33	f	33	0	1	23	$1,27E+04$	0
O	$(0,0)$	r	47	r	47	0	2	22	$1,43E+06$	0
O	$(0,0)$	e	50	e	50	0	3	21	$1,62E+08$	0
O	$(0,0)$	c	44	c	44	0	4	20	$1,83E+10$	0
O	$(0,0)$	h	37	h	37	0	5	19	$2,06E+12$	0
I	$(0,1)$	o	49	i	51	28	6	18	$2,33E+14$	$4,60E+38$
I	$(0,1)$		0	l	43	23	7	17	$2,63E+16$	$3,43E+36$
I	$(0,1)$		0	l	43	23	8	16	$2,98E+18$	$3,04E+34$
O	$(0,0)$		0	o	49	0	9	15	$3,36E+20$	0

$Datos:$			$CalculandoDEx:$	
$l=10$	$L=24$	$R_{max}=56$	$\sum_{i=0}^{l-1} A*B =$	$4,64E+38$
$N=\sum_{i=0}^{L-1}(2R_{max}+1)^i = 2,12E+51$			$\frac{\sum_{i=0}^{l-1} A*B}{N} =$	$2,18E-13$
			$DEx =$	0,026

4.2. Discusión sobre conveniencia de las Distancias de Edición

En esta sección primeramente citaremos los problemas que presentan cada una de las distancias analizadas para la determinación de la similitud entre palabras. Luego resumiremos los resultados de un experimento realizado en (Fernández *et al.*, 2009) sobre la determinación de familias de palabras, que demuestra la validez de esta propuesta frente a otras distancias de edición (usadas en aplicaciones de PLN como distancia de Levenshtein, Jaro, Jaro-Winkler y Needleman-Wunsch). Además describimos otro experimento realizado para medir el rendimiento de las diferentes distancias en la recuperación de palabras similares con o sin ruido. Todo este estudio constituyó la base por la cual decidimos emplear DEx como punto de partida de nuestra propuesta. Finalmente, resaltaremos las características de DEx que son útiles en nuestra aplicación y los problemas que debemos solucionar.

4.2.1. Desventajas de las Distancias de Edición

Entre las desventajas de las distancias anteriormente analizadas en su utilización para medir la similitud entre palabras, se pueden destacar:

- Distancia de Levenshtein: no tiene en cuenta la posición donde se producen las operaciones. Por tanto, su utilización para medir la similitud entre palabras o hallar familias de palabras (también conocidas como familias léxicas) no siempre resulta favorable. Por ejemplo, al analizar las cadenas "luego" y "juego" la distancia de Levenshtein entre ellas sería 1; pero estas dos palabras no guardan ninguna relación y deberían tener una distancia mayor. Por otro lado, las cadenas "campo" y "campesino" dan una distancia de 4, debido a las transformaciones de eliminación de las letras "e-s-i-n"; sin embargo deberían tener menor distancia entre ellas ya que son palabras de una misma familia.
- Distancia Needleman-Wunsch: realiza una generalización de la DE aplicando penalizaciones y permitiendo secuencias de caracteres erróneos en el alineamiento, pero no tiene en cuenta el tipo de transformación que se realiza. Sin embargo, no debería considerarse de igual importancia una operación de eliminación que una sustitución. Tampoco es igual la penalización que debería asignarse por el cambio de una vocal sin acento por una acentuada, que la sustitución de dos consonantes totalmente diferentes. Ejemplo: "enseñar" con "enseñár" y "enseñar" con "ensekar", en este caso debe ser mayor la penalización de la sustitución de la "ñ" por la "k" que la vocal "a" por la vocal acentuada "á".
- Distancia Jaro-Winkler: plantea el tratamiento especial de las transformaciones que se producen en el ámbito de un prefijo de longitud fija, mediante un factor de escala constante; encontrándose precisamente en este detalle su principal desventaja. El problema de esta distancia radica en que adoptan una longitud fija para el prefijo y un factor de escala estático en lugar de dinámico. Por tanto, el algoritmo fallaría en los casos donde

las palabras analizadas tuviesen un prefijo que no cumpliese con el valor fijado en el algoritmo.

- Distancia de Edición eXtendida: es una extensión de la distancia de Levenshtein, donde se realizan penalizaciones teniendo en cuenta la posición y el tipo de las operaciones o transformaciones que se llevan a cabo; además del caracter involucrado en la operación. De esta manera resuelve algunos de los problemas planteados anteriormente. Sin embargo, mantiene un problema común a todas las distancias anteriores: no tienen en cuenta el tratamiento de mutipalabras. Con esto queremos decir, que no son capaces de encontrar una similitud entre palabras simples y múltiples o entre multipalabras; por ejemplo la similitud entre los nombres científicos de las especies de abetos en el dominio agrícola: "abies"-"abies alba", "abies alba"-"abies balsamea", "abies alba"-"abies sachalinensis". Otros ejemplos más comunes, que demuestran la necesidad de tener en cuenta la comparación entre multipalabras, pudieran ser: "tracto"-"tracto digestivo", "tejido"-"tejidos naturales", "nutrición"-"nutrición humana", "animal"-"animales domésticos", etc.

4.2.2. Comparación entre las Distancias de Edición

En esta sección describimos dos experimentos para analizar la capacidad que tienen los algoritmos estudiados de calcular la distancia entre una palabra seleccionada como pivote[3], y el resto de palabras. El primer experimento versa sobre la comparación de palabras de una misma familia y el segundo sobre la determinación del nivel de similitud entre palabras limpias y ruidosas.

Experimento con familias de palabras En esta sección hemos realizado un experimento para analizar la capacidad que tienen los algoritmos estudiados de calcular la distancia entre una palabra seleccionada como pivote, en este caso será siempre el verbo en infinitivo, y todas las derivaciones (su familia de palabras). Se supone que todas las palabras pertenecientes a una familia deben

[3] Pivote: en inglés *pivot*, se usa en este contexto para hacer referencia al término sobre el que gira la comparación con otros términos, a través del cálculo de la distancia DEx

compartir la misma raíz o tallo. Esto presupone que la distancia desde todas ellas a la forma simple (o sea, el verbo en infinitivo) debe presentar variaciones muy pequeñas. Por tanto, la representación de todas las distancias resultantes en un gráfico debería tender lo más posible a una recta, debido a que se supone que entre todas las palabras de una familia deben haber distancias relativamente similares.

Se tomaron para el experimento aleatoriamente tres familias de palabras, la familia de los verbos "enseñar", "fabricar" y "cabalgar". Los conjuntos están compuestos por 64, 66 y 55 palabras respectivamente. En las figuras 4.1, 4.2 y 4.3 se muestran los resultados obtenidos y la conducta que siguen los valores de las distancias calculadas con respecto a un fragmento de sus familias de palabras.

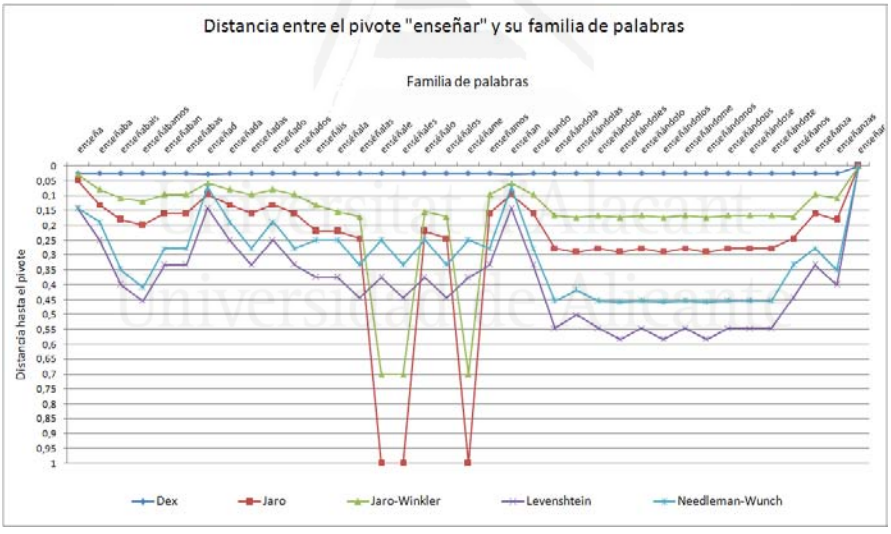

Figura 4.1. DEx entre el pivote "enseñar" y un fragmento de su familia de palabras.

Como se puede apreciar en las figuras 4.1, 4.2 y 4.3, la distancia DEx mantiene los valores de distancia entre todas las palabras con una tendencia a una línea recta, lo que es lógico, pues la distancia entre palabras de una misma familia debe ser muy pequeña. Se puede ver en las gráficas que las restantes distancias, comparadas

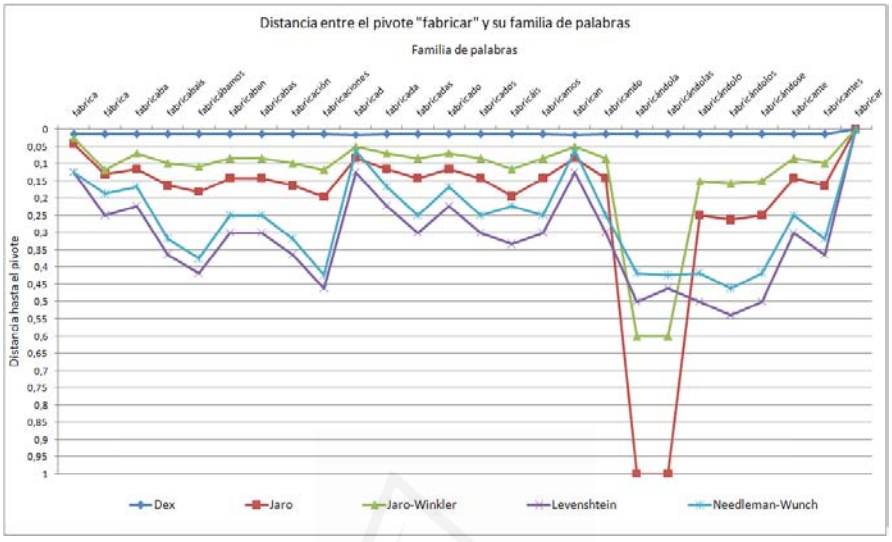

Figura 4.2. DEx entre el pivote "fabricar" y un fragmento de su familia de palabras.

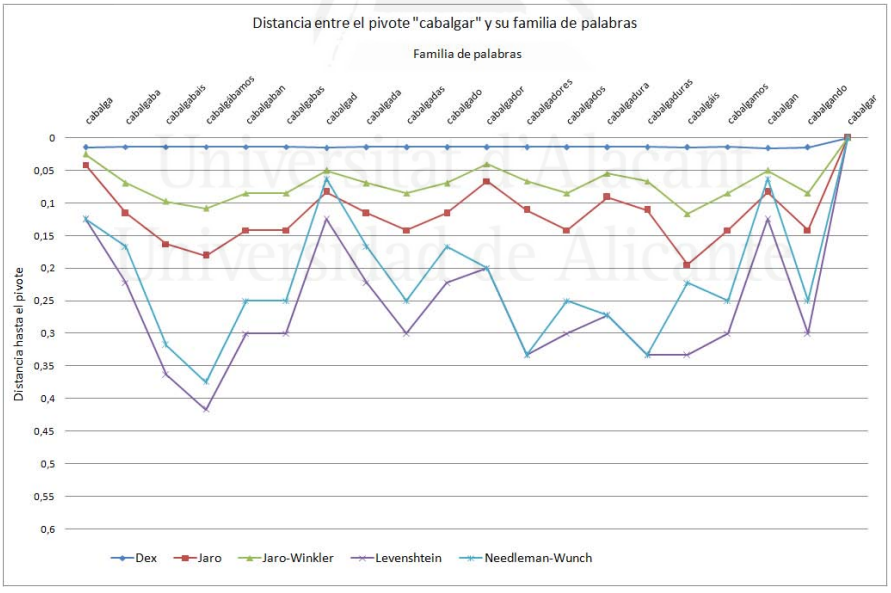

Figura 4.3. DEx entre el pivote "cabalgar" y un fragmento de su familia de palabras.

en nuestro estudio, dan valores muy grandes para las palabras acentuadas. También sucede lo mismo cuando se están compa-

rando las palabras más largas del conjunto, ya que no tienen en cuenta que se comparte un tallo o raíz común y las diferencias están en los sufijos. Por lo que todas las distancias, excepto DEx, demuestran en sus resultados que no tienen en cuenta el tipo de caracter involucrado en cada operación y que son muy dependientes de la longitud de las palabras comparadas.

Experimento con palabras limpias y ruidosas El objetivo de este segundo experimento fue estudiar el comportamiento de las diferentes distancias, sobre un corpus de dominio restringido (concretamente la revista científica agrícola RCCA) con palabras limpias y ruidosas. Se diseñó de tal manera que se pudiera comprobar la capacidad de los algoritmos para recuperar las palabras más relevantes del corpus a partir de una palabra pivote. Para la evaluación fueron seleccionadas manualmente, por un experto humano, 130 palabras relevantes al pivote seleccionado (en este caso fue la palabra "bacterias"). Se consideraron por el experto palabras relevantes aquellos términos que contenían, de alguna forma, la grafía de la palabra pivote, aunque presentasen ruido.

En el Cuadro 4.4 se muestran las 130 palabras consideradas relevantes a "bacterias". De ellas 49 (palabras destacadas en cursiva en el Cuadro 4.4) fueron consideradas por un experto como ruidosas, ya sea porque estaban en inglés en lugar del idioma del experimento (castellano), por tener errores tipográficos como la pérdida del espacio entre palabras o errores ortográficos. Otro aspecto a señalar es que del total (o sea, 130 palabras), 74 poseen un prefijo con relación a la palabra pivote ("bacterias").

Destaquemos que en este experimento se evaluaron las distancias analizadas previamente (Levenshtein, Jaro, Jaro-Winkler, Needleman-Wunch), además de otras que son muy utilizadas por la comunidad (Monge-Elkan, Smith-Waterman, Smith-Waterman-Gotoh, QGrams distance), por encontrarse disponible su implementación en una Librería de Métricas de Similaridad (en inglés, *Similarity Metric Library*), llamada *SimMetrics*[4], elaborada por la Universidad de Sheffield[5], en el Reino Unido.

[4] `http://sourceforge.net/projects/simmetrics/`
[5] `http://www.dcs.shef.ac.uk/~sam/stringmetrics.html`

Cuadro 4.4. Conjunto de 130 palabras (limpias o ruidosas) relevantes para el pivote "bacterias".

acetobacter	*bacteriascapacesde*	bacterizados	eubacterium
acetobacterium	*bacteriasviablcs*	*bacterjas*	fibrobacter
achromobacter	*bacteriaz*	*bacteroidacea*	flavobacterium
acinelobacter	bactericida	bacteroidaceae	*forbacteria*
acinetobacter	bacterina	*bacteroidacese*	fosfobacterias
aclaeobacter	*bacterins*	bacteroides	*fpasbacterianas*
aerobacter	bacteriocin	*bactertas*	fusobacterium
agrobacterium	bacteriocinas	bifidobacteria	*gnibacteriana*
ahwtobacter	*bacteriocins*	bifidobacteriales	*iacbacterias*
antibacterial	bacteriofagos	bifidobacterias	*identibacterioiogy*
antibacteriana	bacteriol	*bifidobacterim*	*lasbacterias*
antibacterianas	*bacteriolog*	bifidobacterium	*losbacteroides*
antibacteriano	bacteriologia	*bifidobacteriumlactis*	*masabacteriana*
antibacterianos	bacteriologica	campylobacter	methanobacterium
arotobacter	*bacteriological*	cianobacterias	methanobrevibacter
arqueobacterias	bacteriologicamente	citrobacter	microbacterium
arthrobacter	bacteriologicas	colibacteria	microbacteriums
azobacter	bacteriologico	colibacteriosis	polibacterias
azotobacter	bacteriologicos	corinebacterium	propionibacterias
bacteremia	*bacteriology*	corynebacteria	rhizobacterium
bacterhxs	bacteriophage	corynebacterium	rizobacterias
bacteri	bacterioprophylaxis	*debacteria*	robacterias
bacteria	bacteriostatica	*debacteriologia*	*sacchaacinetobacter*
bacteriaceae	bacteriostaticas	*enlerobacter*	*suspensibnbacteriana*
bacteriai	*bacteriotherapy*	*entcrobacterias*	*tbacterias*
bacterial	*bacteris*	*entembacter*	*therumenbacteria*
bacteriales	bacterium	enterobacter	*tietobacter*
bacteriana	*bacteriy*	enterobacteriaceae	*typhimuriumcampylobacter*
bacterianas	*bacterizacibn*	enterobacteriaceaes	
bacteriano	*bacterizacin*	enterobacterias	
bacterianos	bacterizacion	enterobacteriosis	
bacteriao	*bãcterizacion*	entorobacterias	
bacterias	*bacterizacisn*	*estabacteria*	
bacteriasgram	bacterizadas	*estasbacterias*	

Para evaluar los resultados de este experimento, se calcularon los valores de precisión y cobertura para cada algoritmo. La precisión fue calculada por el número de palabras relevantes recuperadas dividido por el número de palabras recuperadas. Por otro lado, la cobertura se calculó por la división entre el número de palabras relevantes recuperadas y el número total de palabras relevantes existentes. En las figuras 4.4 y 4.5 se puede observar que el algoritmo de la DEx funciona muy bien recuperando las palabras relevantes definidas por el experto humano (ver, por ejemplo, en el Cuadro 4.5 las primeras 50 palabras relevantes devueltas por DEx).

La DEx alcanza la máxima precisión (o sea, con valor igual 1) en sus primeras 50 palabras recuperadas (ver Figura 4.4). Mientras los restantes algoritmos obtienen valores fluctuantes, incluso recuperan algunas palabras señaladas por el experto como no relevantes en ese mismo intervalo. Por ejemplo, las tres prime-

Cuadro 4.5. Lista de las primeras 50 palabras devueltas por DEx.

Pos.	Palabra	Pos.	Palabra
1	bacterias	26	bacterioprophylaxis
2	bacteriasviablcs	27	bacteriotherapy
3	bacteriasgram	28	bacteriologicas
4	bacteriascapacesde	29	bacteriological
5	bacteriales	30	bacteriologicamente
6	bacteriaceae	31	bacteriologicos
7	bacterianos	32	bacteriologico
8	bacterianas	33	bacteriologica
9	bacteriano	34	bacteriologia
10	bacteriana	35	bacteriology
11	bacteria	36	bacteriolog
12	bacteriaz	37	bacteriocinas
13	bacterial	38	bacteriocins
14	bacteriao	39	bacteriocin
15	bacteriai	40	bacteriostaticas
16	bacterizados	41	bacteriostatica
17	bacterizadas	42	bacteris
18	bacterizacisn	43	bacteri
19	bacterizacibn	44	bacteriy
20	bãcterizacion	45	bacterium
21	bacterizacion	46	bacterins
22	bacterizacin	47	bacterina
23	bactericida	48	bacteriol
24	bacteriofagos	49	bacteroidacese
25	bacteriophage	50	bacteroidaceae

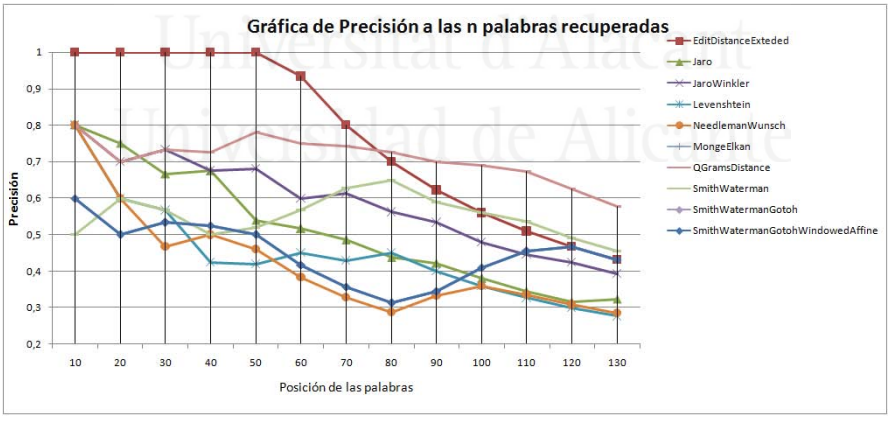

Figura 4.4. Comparación entre DEx y otros algoritmos de distancia entre palabras (precisión a las n palabras recuperadas).

ras palabras ("bac", "eri","eria") que recupera el algoritmo de Monge-Elkan no son relevantes; lo mismo sucede con la primera y quinta palabras ("ia" y "rias") recuperadas por el algoritmo de

Smith-Waterman; y así con el resto de algoritmos. Todas las distancias analizadas excepto la DEx se equivocaron en un promedio de 27 palabras incorrectamente recuperadas, o sea no relevantes según el criterio del experto, hasta la posición 50.

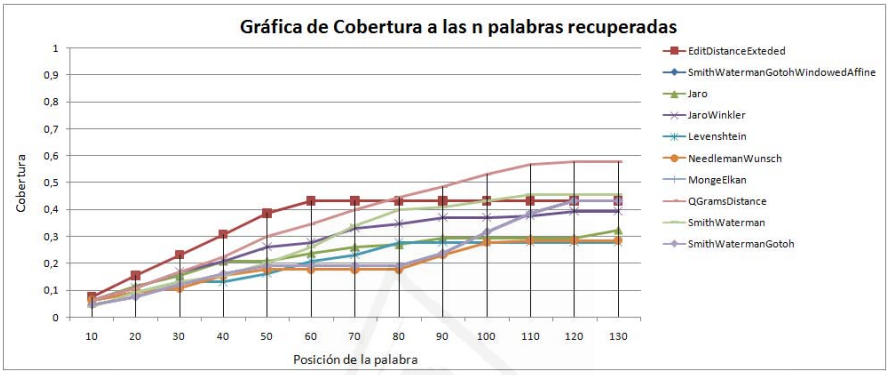

Figura 4.5. Comparación entre DEx y otros algoritmos de distancia entre palabras (cobertura a las n palabras recuperadas).

Los algoritmos que alcanzaron mayor cobertura total fueron QGrams-Distances con $0,58$ y Smith-Waterman con $0,45$. Nuestro algoritmo alcanzó una cobertura de $0,43$ (ver Figura 4.5). La diferencia radica en que el algoritmo de la DEx penaliza fuertemente las palabras que tienen cambios en la raíz o tallo, por ejemplo, este es el caso de "propionibacterias" y otros ejemplos con prefijos muy grandes. De las 130 palabras escogidas por el experto humano como relevantes, más del 50 % tiene la característica de poseer prefijos; esto lógicamente perjudica en particular la cobertura de la DEx. Sin embargo, en las primeras 50 palabras recuperadas si que DEx alcanza la cobertura máxima entre todas las distancias y obtiene además una precisión del 100 % en ese mismo rango, es decir, la DEx devuelve en las primeras posiciones las palabras relevantes recuperadas como se pudo ver en el Cuadro 4.5. Por otro lado, el algoritmo recupera algunas variantes ruidosas que parecen formas derivadas de la palabra "bacterias", aunque éstas no fueron consideradas por el experto humano (ver el Cuadro 4.6).

Cuadro 4.6. Palabras ruidosas recuperadas por DEx que tienen relación con el pivote "bacterias".

bact	bactcrizar	bactcria	bactenas
bactcriana	bactris	bactcrinna	bactcrialla
bactcrium	bactcri	bacteiias	bactcriris
bactfrias	bactcriax	bactt	bactetia
bactcrhs	bacte	bactcrial	
bactcriant	bactrrias	bactcriologico	

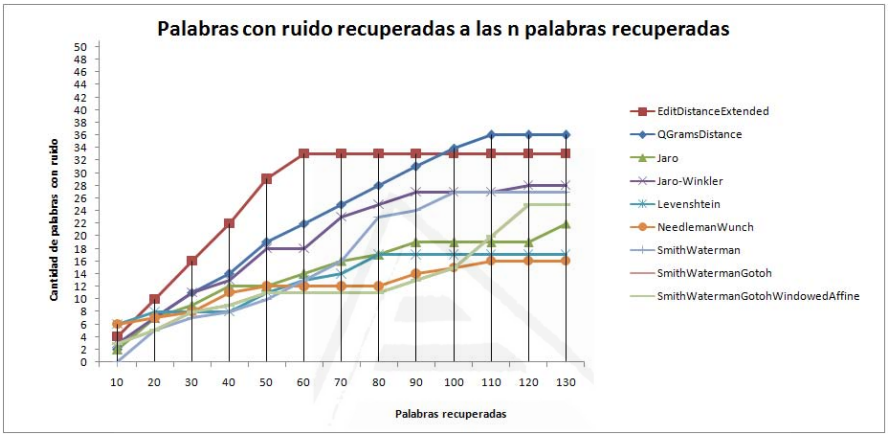

Figura 4.6. Comparación entre DEx y otros algoritmos sobre la cantidad de palabras con ruido recuperadas (a las n palabras recuperadas).

En la Figura 4.6 se puede ver también la elevada capacidad que tiene DEx para recuperar palabras con ruido, ya que se refleja en el gráfico que es la distancia que mayor cantidad de palabras captura, sobre todo en las primeras posiciones. Es sólo superado en 3 palabras por *QGrams-distance* y después de las 100 palabras recuperadas. También se comprobó el rendimiento de DEx con respecto al resto de las distancias en la recuperación de palabras limpias a partir de un pivote ruidoso como "bacteriascapacesde". En la Figura 4.7 se aprecia que alcanza la máxima precisión en las primeras 10 palabras recuperadas y siempre alcanza el mayor valor de precisión con respecto a las otras distancias para un total de 90 palabras recuperadas. Para que se vea más claramente lo que decimos en el Cuadro 4.7 se muestran las primeras 10 palabras recuperadas por todos los algoritmos.

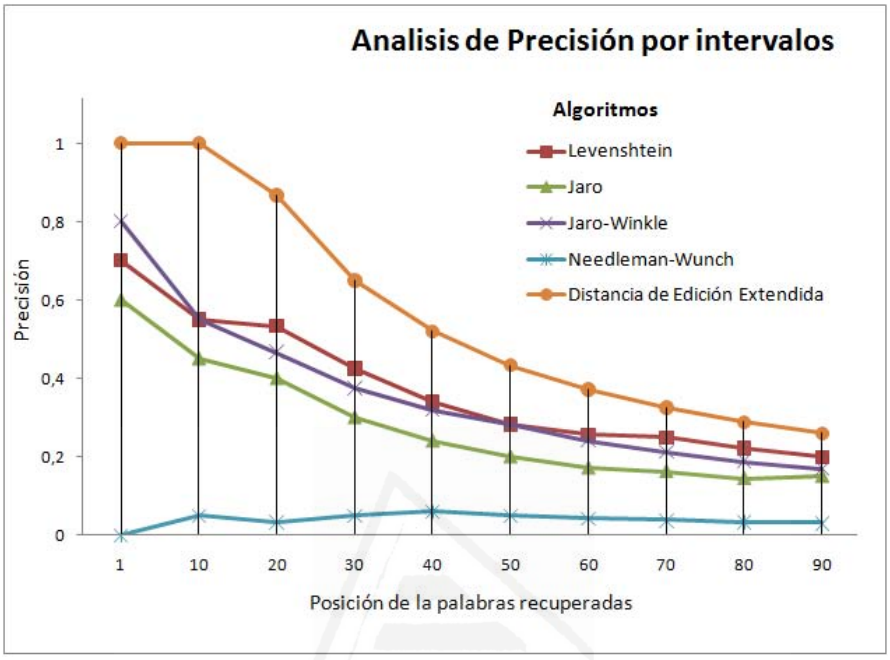

Figura 4.7. Comparación entre DEx y otros algoritmos sobre la precisión de palabras recuperadas a partir de un pivote con ruido.

Cuadro 4.7. Palabras recuperadas por las diferentes distancias a partir de un pivote con ruido "bacteriascapacesde".

Pos.	DEx	Levenshtein	Jaro-Winkler	Needleman-Wunsch
1	bacterias	bacterianas	bacteriana	discapacidad
2	bacteria	Valerianaceae	bacterianas	Mycoplasmataceae
3	bacteriano	Bacteriocinas	bacterianos	Arteriosclerosis
4	bacterianos	Datiscaceae	bactrianus	Dipterocarpaceae
5	bacterianas	Dipterocarpaceae	bacteriano	Hippocastanaceae
6	bacteriana	Bacteroidaceae	bactriano	Anaplasmataceae
7	bactericidas	bacteriana	bacteriophora	Spiroplasmataceae
8	bacterium	Bactericidas	batei	Menispermaceae
9	bacteriophora	bacterianos	baeri	Nitzschiaceae
10	bacteriólogos	bactrianus	bacteriológica	Circaeaceae

Finalmente podemos afirmar que los resultados mostrados en estos experimentos constituyen la razón fundamental por la cual hemos elegido el algoritmo de la DEx como base de nuestra propuesta de tolerancia a ruido textual en sistemas de RI sobre dominios restringidos. Además tuvimos en consideración otros criterios para esta elección, que expondremos en la siguiente sección.

4.2.3. Motivación del uso de DEx para el cálculo de similitud entre palabras

A continuación detallaremos algunas de las ideas que sostiene la propuesta DEx, definida en (Fernández *et al.*, 2009), y que constituyen las causas de la mejora en los resultados del cálculo de la similitud entre palabras, en comparación con el resto de las distancias.

1. Análisis de la etimología de la palabra.
   En aplicaciones que necesitan calcular la similitud entre palabras es indispensable utilizar algoritmos que realicen un análisis más profundo de los constituyentes de la palabra: la raíz o tallo y los afijos. El tallo (en inglés, *stem*) aporta el significado principal de la palabra y los afijos lo modifican, ofreciendo información complementaria (esto es, información léxica expresada a través del sufijo o información léxico-sintáctica expresada mediante las desinencias[6]). Por consiguiente, una medida de similitud (o Distancia de Edición) para la comparación entre palabras debe tener en cuenta en qué constituyente ocurre la transformación: si ocurre en los sufijos se le debe dar menor importancia a la transformación; por el contrario, si ocurre directamente en la raíz se le debe dar mayor peso pues provocarían un posible cambio de significado. Por ejemplo, algunos prefijos pueden cambiar el sentido de las palabras, causando indicativo de anterioridad (ayer-anteayer), oposición (feliz-infeliz), superioridad (tensión-hipertensión), inferioridad (tensión-hipotensión), exceso (millonario-multimillonario), etc.
   Precisamente, DEx implementa esta idea y para ello utiliza el parámetro $(2R_{max} + 1)^{L-i}$ (ver la Ecuación 4.3), a través del cual penaliza la posición donde se realiza cada operación (eliminación, sustitución e inserción).
2. Consideración de la alternancia prosódica y gráfica del idioma.
   En muchos lenguajes, como el español, existe la alternancia prosódica (o sea, alternativas fónicas como los acentos de un

[6] Desinencia: Morfema flexivo añadido a la raíz de adjetivos, sustantivos, pronombres y verbos. Dicha terminación variable de las palabras tiene una función morfológica o gramatical.

misma palabra) y gráfica (o sea, alternativas ortográficas de una misma palabra). Algunos ejemplos aceptados por la Real Academia Española son: "afrodisiaco-afrodisíaco" y "atmosfera - atmósfera" de alternancia prosódica; y "azimut-acimut" y "zeta-ceta" de alternancia gráfica. Por tanto, es importante que las métricas de distancia de edición para medir la similitud entre las palabras tengan en cuenta este aspecto.

Para tener en cuenta estas alternativas que se pueden presentar en un lenguaje, DEx propone colocar un peso a los caracteres, basado en la frecuencia de aparición de éstos en un diccionario del lenguaje. De esta forma, si se desean flexibilizar algunos casos se puede hacer igualando los pesos de los caracteres que se quieren considerar como similares. Por ejemplo, la DEx resuelve la alternancia prosódica asignando el mismo peso a las vocales acentuadas y no acentuadas. De igual manera (es decir, basándose en el peso de los caracteres) se propone resolver la alternancia gráfica, pero sólo en los casos donde se alterne sólo un carácter, los casos como "kiosco-quiosco" no son resueltos por el algoritmo, ni por el resto de medidas analizadas.

3. Flexibilidad ante errores tipográficos y ortográficos.
En la mayoría de los idiomas existen palabras de escritura dudosa, en las que observar errores ortográficos es algo muy común. El simple hecho de sustituir una letra "s" por "c" o "z" en castellano puede incrementar la distancia entre dos palabras y algunas veces esto no cambia su significado. Por otro lado, en muchos documentos, las diferencias son ocasionadas por errores tipográficos, humanos o provocados por las herramientas automáticas de procesamiento textual. Todos estos casos se tienen en cuenta en DEx a través de la colocación de los pesos a los caracteres. Por supuesto, la significación del error está en dependencia del dominio. Por esta razón es necesario dar la posibilidad de aplicar una penalización menos costosa en estos casos. Consecuentemente, con las técnicas de similitud basada en palabras se debe penalizar con un costo que tenga en cuenta qué carácter está siendo insertado, borrado o sustituido.

Precisamente, debido a todas las razones que hemos venido planteando y a los buenos resultados mostrados por DEx en la comparación con otras distancias de edición (que mostramos anteriormente en la Sección 4.2.2) hemos decidido tomar DEx como base para calcular la distancia entre los términos o palabras dentro de nuestra propuesta. Resumiendo algunas de las características estudiadas que hacen a DEx apropiada para nuestro propósito tenemos: (i) las distancias pueden ser calculadas con las palabras originales, o sea sin aplicar lematización, lo que es muy útil en las palabras ruidosas porque éstas no pueden ser correctamente lematizadas; y (ii) las penalizaciones son aplicadas acorde con el tipo, posición y caracter involucrado en la operación de modificación (inserción, eliminación, sustitución o no operación), siendo así útil para tratar con patrones de ruido que frecuentemente aparecen en el corpus (ruido altamente repetitivo). Sin embargo, queda un problema pendiente por solucionar para obtener una adecuación máxima de DEx en el contexto que deseamos utilizarla, y es la adaptación de DEx de manera tal que pueda realizar comparaciones entre multipalabras, y entre palabras simples y múltiples. Esta cuestión es de gran importancia en aplicaciones de dominio restringido, específicamente en dominios técnicos y científicos (p.e. médico, agrícola, químico, etc.) donde es muy frecuente la aparición de este tipo de palabras. En la siguiente sección detallaremos nuestra propuesta de extensión de DEx para que funcione adecuadamente con las multipalabras.

4.3. Extensión de la DEx para multipalabras

En esta sección describimos el trabajo realizado para obtener una extensión del algoritmo DEx, para considerar también las comparaciones entre palabras simples y multipalabras de manera eficiente.

Como fue previamente indicado, para que el uso de la distancia DEx sea apropiado en un dominio restringido, la misma debe ser capaz de tratar las multipalabras, ya que éstas aparecen a menudo en los dominios restringidos. La fórmula DM (Distan-

cia para Multipalabras) en la Ecuación 4.4) se basa en el cálculo de DEx para cada palabra que aparece en las multipalabras. Seguidamente explicamos la secuencia algorítmica propuesta para calcular DM. Además, desarrollaremos un ejemplo (comparación entre las palabras "afrecho de trigo" y "afrechillo") para alcanzar una mejor comprensión del algoritmo.

1. Tokenizar[7] los palabras a comparar siempre que sean multipalabras, con el objetivo de analizar cada término como entidad independiente.
2. Crear y llenar una matriz según la Distancia de Levenshtein o DE con las palabras analizadas, de la misma manera que se hacía para la DEx en la sección 4.1.4; con la ligera diferencia de que el elemento de comparación es el valor de la DEx (calculado según la ecuación 4.3 en la sección 4.1.4) entre las palabras y no la simple relación de igualdad (ver el siguiente paso en esta secuencia algorítmica). En el Cuadro 4.9 se muestra cómo queda la matriz para el ejemplo que estamos desarrollando después de realizar este paso y el siguiente (paso 3) de manera simultánea.
3. Determinar similaridad entre las palabras. Para ello se establece un umbral dinámico a partir de la evaluación de la DEx para el caracter de menor importancia o *ranking* del diccionario de caracteres (P detallado previamente en la sección 4.1.4) en la posición media de la cadena de operaciones. Teniendo en cuenta estos aspectos, se puede decidir la similitud entre las palabras comparadas: si la distancia DEx no supera el umbral se asume que los token son similares, en caso contrario son diferentes. Siguiendo con el ejemplo, este paso se puede ver en el Cuadro 4.8. Donde se observa que la DEx resultante de la comparación entre las palabras "afrecho" y "afrechillo" (es igual a $0,026$ como se calculó en la sección 4.1.4) no supera el umbral de $0,028$, por esta razón se consideran las palabras como parecidas; no siendo así con las restantes.

[7] Tokenizar: acción que se realiza para detectar los tokens. Un token o también llamado componente léxico es una cadena de caracteres que tiene un significado coherente en cierto lenguaje (ya sea natural o de programación). Las palabras en un lenguaje natural son ejemplos de tokens.

Cuadro 4.8. Ejemplo del cálculo de la similitud entre dos cadenas comparadas.

Pivote	Palabra analizada	Umbral	DEx	Similitud
afrecho	afrechillo	0,028	0,026	SI ($DEx < Umbral$)
de	afrechillo	0,052	0,908	NO ($DEx > Umbral$)
trigo	afrechillo	0,052	0,909	NO ($DEx > Umbral$)

Por tanto, la matriz quedaría de la siguiente manera:

Cuadro 4.9. Matriz para la comparación entre el pivote "afrecho de trigo" y la palabra "afrechillo".

		afrechillo
	0	
afrecho	1	0
de	2	1
trigo	3	2

4. Crear simultáneamente otra matriz para guardar los valores de la distancia DEx entre los token comparados en el paso anterior, con el objetivo de posibilitar posteriores consultas.
5. Determinar la cadena de operaciones a partir de la matriz anteriormente obtenida (en el Cuadro 4.9). Para el ejemplo que se analiza sería "ODD", porque hay una no operación ("O") entre "afrecho" y "afrechillo" y dos operaciones de eliminación ("DD") de las palabras "de" y "trigo".
6. Evaluar la cadena de operaciones en la ecuación de la DM, que se define según la ecuación 4.4

$$DM = \sum_{i=1}^{L} (V(O_i) \cdot CP + CD \cdot DE_{x_i})$$

donde :

$$V(O_i) = \begin{cases} 0 \; if \; O_i = o \\ 1 \; if \; O_i \neq o \end{cases} \tag{4.4}$$

La descripción de los parámetros y constantes usadas se muestra a continuación:

O_i: Cadena de operaciones en la posición i-ésima (O-No operación, I-Inserción, D-Eliminación, S-Sustitución).

$[V(O_i)]$: es el vector de operaciones. Que toma los valores cero o uno en dependencia de si la cadena de operaciones en i es una "no operación" o no.

$$V(O_i) = \begin{Bmatrix} 0 \text{ si } O_i = O \\ 1 \text{ si } O_i \neq O \end{Bmatrix}$$

L: Longitud de la cadena de operaciones (O_i).

CP: Grado de afectación por la posición en que se encuentra la palabra dentro de la cadena o multipalabra (CP =0.95 de FP). Este valor fue obtenido empíricamente a partir de varios experimentos, tomándose éste como el valor que aportaba los mejores resultados.

CD: Grado de afectación por DEx entre las palabras comparadas (CD =0.05 de FP). CD es usado para establecer un orden entre las comparaciones que resulten similares.

DE_{x_i}: Distancia de Edición eXtendida (DEx), según ecuación 4.3, entre las palabras en la posición i dentro de las multipalabras.

$FP = 2^{-1(i+1)}$: es un factor de penalización usado para darle peso a la posición donde ocurre la transformación entre las palabras comparadas. Gracias al comportamiento exponencial de este elemento se puede penalizar más rigurosamente aquellas transformaciones que ocurran más a la izquierda de la multipalabra.

Para el ejemplo la evaluación quedaría como se muestra en el Cuadro 4.10. Aclaremos que la columna **A** contiene los valores de calcular $V(O_i) \cdot CP$, la **B** los de calcular $CD \cdot DE_{x_i}$ y la **A+B** es igual a $(V(O_i) \cdot CP + CD \cdot DE_{x_i})$. Finalmente el valor de distancia entre "afrecho de trigo" y "afrechillo" según DM es de $0,37$; por lo que se pueden considerar palabras similares.

Cuadro 4.10. Ejemplo paso 7 de la DM: evaluación de la Ecuación 4.4.

i	$V_{(O_i)}$	FP	CP	CD	A	DE_{x_i}	B	$A+B$
0	0	0,5	0,475	0,025	0	0,026	0,00065	0,00065
0	1	0,25	0,2375	0,0125	0,2375	0,908	0,01135	0,24885
0	1	0,125	0,11875	0,00625	0,11875	0,909	0,00568	0,12443
							$DM =$	0,3739

Finalmente, es necesario resaltar que mientras más alto es el resultado de la DM, más alta es la distancia entre las palabras y por tanto, son menos similares.

En la siguiente sección se describe nuestra propuesta para obtener sistemas de RI tolerantes al ruido en dominios restringidos, en la cual es empleado el algoritmo DM que hemos definido. Además en el capítulo 7 se realizarán algunos experimentos con este distancia en función de compararla con las otras distancias que han sido analizadas en este capítulo.

4.4. Descripción de nuestra estrategia de tolerancia al ruido textual en la RI

En esta sección se describe nuestra propuesta para adicionar una funcionalidad tolerante al ruido a un sistema de RI en dominios restringidos. Como se decía anteriormente nuestra propuesta se basa en la premisa de que los términos más importantes de las consultas, realizadas a un sistema de RI sobre dominio restringido, están relacionados al propio dominio y precisamente el ruido presente en dichos términos en el corpus será la causa fundamental del decrecimiento de la precisión del sistema de RI. Consecuentemente, los sistemas de RI sobre dominios restringidos deben ser conscientes del ruido en los términos del dominio presentes en el corpus. Con este objetivo, nuestra propuesta compara los términos en el SOC del dominio disponible con los términos en el corpus y con los términos de la consulta realizada por el usuario al sistema de RI, por medio del algoritmo DM, definido en la Sección 4.3.

Se puede apreciar una visión general de nuestra aproximación en la Figura 4.8, la cual tiene dos fases (indexación y búsqueda) y tres etapas principales:

1. Proceso *offline* en la fase de indexación, que constituye la primera etapa, donde se calculan las distancias (usando la distancia DM) correspondientes entre cada término indexado del corpus y los términos del SOC. Estos datos se almacenan en un vector terminológico (ver sección 4.4.1).

2. Proceso *online* en la fase de búsqueda, donde primero se lleva a cabo la segunda etapa de nuestra propuesta: (i) calcular la distancia entre cada término relevante en la pregunta y los términos del SOC y (ii) definir así los vectores terminológicos para cada uno de esos términos (ver sección 4.4.2). Luego se lleva a cabo la tercera y última etapa que incluye diferentes pasos: (iii) definir la correspondencia entre los términos de la pregunta y los términos indexados del corpus (con o sin ruido), usando como criterio un número determinado de términos del SOC comunes a ambos y con una distancias DM determinada, (iv) realizar dos procesos de RI independientes, el primero usa el sistema de RI con los términos originales de la pregunta y el segundo con los términos correspondientes que aparecen con ruido en el corpus, y (v) unificar y reordenar los resultados de los dos procesos de RI para devolver el listado de respuestas. Una explicación más detallada de esta última etapa se puede consultar la sección 4.4.3.

4.4.1. Obtención del vector terminológico de cada término indexado (Etapa 1)

En la fase de indexación (ver el proceso *offline* en el lado izquierdo de la Figura 4.8) se llevan a cabo los siguientes pasos:

1. Convertir el corpus a ficheros de texto plano.
2. Tokenizar los términos del corpus. Resaltemos que en esta fase sería muy importante realizar la tokenización respetando a las multipalabras. Para ello se puede proponer una estrategia sencilla como es el uso de un analizador sintáctico, considerando a los constituyentes de determinadas relaciones sintácticas (p.e. nombres-nombres y nombres-adjetivos) anidadas como inseparables. Otras alternativas para detectar multipalabras propuestas actualmente pueden ser: (i) usar estructuras de representación de información tales como las redes neuronales y redes bayesianas (Martínez-Santiago *et al.*, 2002; Martínez-Santiago & Ureña-López, 2002), (ii) usar bigramas, es decir, secuencias fijas de dos términos como máximo (Ráez *et al.*,

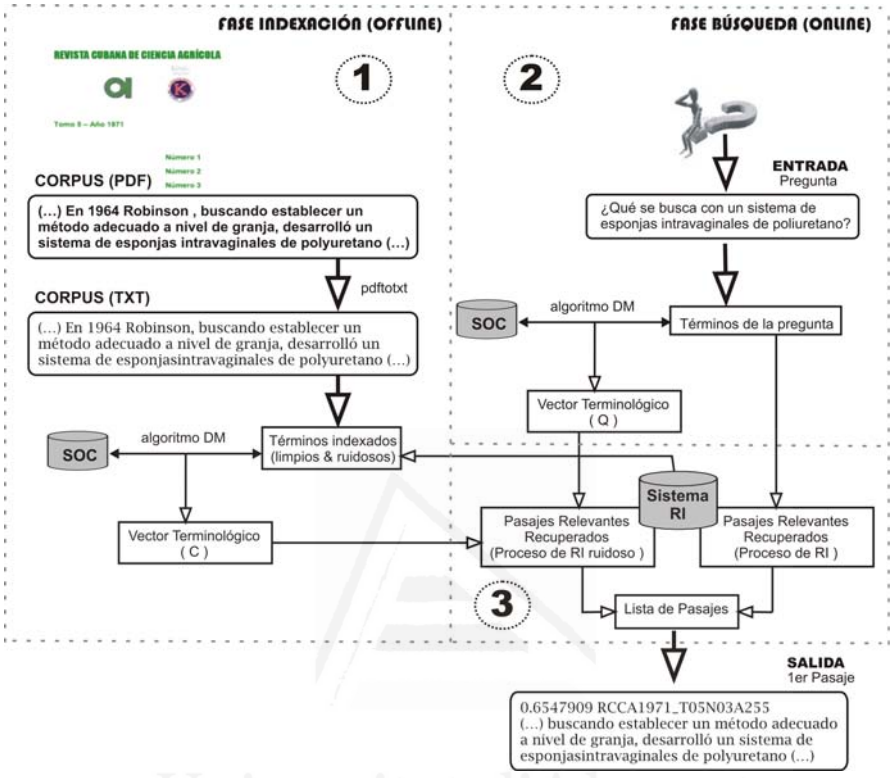

Figura 4.8. Visión general de la aproximación para adicionar tolerancia a ruido en un sistema de RI.

2010), etc. Este detalle no es el objetivo de nuestro trabajo por lo que no profundizaremos más en este sentido.

3. Indexar los términos previamente tokenizados, independientemente del ruido, presentes en el corpus textual usando un sistema de RI; calculando así las frecuencias de estos términos por documento y eliminando las palabras de parada o *stop words* del lenguaje.
4. Mapear (o encontrar la correlación de) cada término indexado con los términos en el SOC de dominio específico usando la distancia DM previamente detallada en la sección 4.3. Vale destacar que para calcular dicha distancia cada palabra (sean palabras simples o multipalabras) del SOC que se emplee constituirá una entrada del diccionario de búsqueda que usará DM. Además se hace necesario elaborar el ordenamien-

to (en inglés, *ranking*) de los caracteres de todos los términos contenidos en el SOC, con la finalidad de formar el diccionario de caracteres (P) que utilizará DM.

5. Crear un vector de términos (C) para cada término indexado, en el cual se almacenarán los términos del SOC mapeados con él y las distancias correspondientes calculadas en el paso anterior.
   El vector de términos puede definirse como sigue: siendo T el conjunto de n términos del SOC mapeado con el algoritmo DM. $t_r \in T$ denota el término r en el conjunto de términos. Luego, el vector de términos que representa al término t_s se define como el vector $V_{t_s} = [(t_1, w_1), (t_2, w_2), ..., (t_n, w_n)]$ donde w_r denota la distancia entre t_s y t_r.

Si tomamos como ejemplo el fragmento de texto representado en la Figura 4.8, se puede apreciar un error tipográfico, debido a la omisión del espacio entre los términos de una multipalabra, en el término ruidoso "esponjasintravaginales" (en lugar de "esponjas intravaginales"). Para desarrollar este ejemplo usamos como corpus la revista RCCA y como SOC el tesauro Agrovoc. Siguiendo los pasos descritos en esta primera etapa de nuestra propuesta, se obtiene el vector terminológico para el ejemplo y se muestra un fragmento del mismo en el cuadro 4.11. Señalemos que el término "esponja(s) intravaginal(es)" no se encuentra en el SOC empleado, pero si otros términos que tienen gran similitud como indican los valores de la distancia DM.

Cuadro 4.11. Ejemplo de vector terminológico definido para la palabra "esponjasintravaginales"

$C_{esponjasintravaginales} =$	[($esponjas$, 0,13), ($esponjilla$, 0,18), ($esponja\_vegetal$, 0,25), (...)]

4.4.2. Obtención del vector terminológico de cada término relevante en la pregunta (Etapa 2)

La segunda etapa tiene lugar en el proceso de análisis de la pregunta, en la fase de búsqueda o recuperación de información (ver el proceso *online* al lado derecho superior de la Figura 4.8). Esta etapa es muy similar a la anterior ya que se debe: (i) tokenizar los términos de la pregunta, (ii) determinar los términos de la pregunta con mayor peso para la RI, (iii) mapear dichos términos con los términos en el SOC usando DM, y (iv) obtener sus correspondientes vectores de términos (Q) de forma similar a como se crearon en la etapa anterior (detallado en la sección 4.4.1).

Como ejemplo de esta segunda etapa, consideremos la pregunta: *"¿Qué se busca con un sistema de esponjas intravaginales de poliuretano?"*. Sin realizar ningún tipo de detección de multipalabras en la pregunta, se tomaría la palabra "esponjas" como uno de los términos relevantes para la RI y en esta etapa se hallaría su vector terminológico (como se muestra en el Cuadro 4.12). Los vectores de términos C y Q calculados en las etapas 1 y 2 respectivamente, serán empleados en la siguiente etapa de nuestra propuesta.

Cuadro 4.12. Ejemplo de vector terminológico definido para la palabra "esponja-sintravaginales"

$Q_{esponjas} =$	[(*esponjas*, 0), (*esponjilla*, 0,1), (*esponja_vegetal*, 0,25), (...)]

4.4.3. Ejecución de dos procesos de RI independientes (Etapa 3)

La tercera etapa tiene también lugar en la fase de búsqueda(ver el proceso *online* al lado derecho inferior de la Figura 4.8). Esta etapa consiste en dos procesos de recuperación de información independientes, con y sin términos ruidosos en la pregunta. Para ello se realizan varios pasos:

1. Ejecutar el primer proceso de RI (llamémosle proceso de RI *baseline*), usando cualquier sistema de RI disponible, con los términos originales de la pregunta en función de devolver los documentos o pasajes[8] con mayor probabilidad de contener la respuesta. Vale destacar que tomamos como respuestas pasajes en lugar de documentos, con la única finalidad de hacer más precisa y sencilla la evaluación, ya que las respuestas candidatas serían más cortas. Pero esto no quiere decir en ningún momento que nuestra propuesta no sea aplicable para sistemas de RI que devuelvan documentos en lugar de pasajes como respuestas. Además los sistemas de recuperación de pasajes utilizan los mismos modelos tradicionales de RI pero sustituyendo al documento por el pasaje.
   Continuando con el ejemplo que estamos desarrollando (pregunta: *"¿Qué se busca con un sistema de esponjas intravaginales de poliuretano?"*) y empleando un sistema cualquiera de RI disponible como JIRS (Soriano, 2007; Buscaldi *et al.*, 2010) (ver el capítulo 7 para mayor descripción del sistema), este primer proceso de recuperación retornaría el siguiente pasaje (con una probabilidad de 0.34 de contener la respuesta correcta).

   > Input query: 174146 0.34165043 RCCA1989_T23N03A241 "... iguales para los tres sistemas. Sc concluye que en el sistema de dos cuartones las vacas tuvieron que realizar un mayor esfuerzo cn busca del alimento, por la menor disponibilidad dc pastos cn este sistema ..."

   La razón por la cual recupera este pasaje en primera posición es que uno de los términos relevantes de la pregunta ("sistemas") tiene una frecuencia de repetición elevada. No obstante, el bajo valor de peso de ese pasaje (probabilidad de 0.34) indica que otros términos importantes (p.e. "esponjas", "intravaginales", y "poliuretano") no pueden encontrarse en el corpus por causa del ruido que éste presenta.
2. Definir la correspondencia entre los términos de la pregunta y los términos indexados del corpus (con o sin ruido). Entonces, los vectores terminológicos C y Q (determinados en las

[8] Un pasaje se define como una secuencia contigua de texto dentro de un documento.

etapas anteriores) se usan para sustituir apropiadamente los términos en la pregunta por los términos correspondientes que aparecen con ruido en el corpus. Es decir, los términos de los vectores terminológicos (o sea, términos del SOC) se usan como lenguajes intermedio entre los términos de la pregunta y del corpus, entre términos limpios y términos ruidosos.
El criterio que seguimos para determinar la correspondencia entre términos es:
Si el término i representado por el vector de términos $Q_i \in Q$ y el término j representado por el vector de términos $C_j \in C$ cumplen que:

- $i \neq j$,
- *existen tres o más términos en Q_i que estén también contenidos en C_j y cuyas distancias estén entre 0 (que indica que son prácticamente iguales según DM) y un umbral máximo dado (que indica el grado de similitud según DM).* Dicho umbral depende del dominio de aplicación y por tanto, será definido de manera empírica a través de experimentos para evaluar la efectividad de DM en el dominio en cuestión (un ejemplo se puede ver en el capítulo 7), o
- *existen al menos tres elementos en Q_i que estén contenidos también en C_j y cuyas distancias sean todas igual a 0.*

Entonces son correspondientes el término i y j y pueden ser usados en el siguiente paso.
Analizando los vectores (ver cuadros 4.12 y 4.11) calculados en las etapas anteriores para el ejemplo en desarrollo, se concluye que "esponjas" puede ser sustituido por "esponjasintravaginales", ya que son términos correspondientes porque tienen tres términos comunes que cumplen las condiciones anteriores (ver la Figura 4.9).

3. Ejecutar el segundo proceso de RI (llamémosle Proceso de RI ruidoso) de forma independiente al primero, usando los términos correspondientes (limpios o ruidosos) hallados en el paso anterior. Específicamente, este segundo proceso consiste en lanzar un proceso de recuperación independiente para cada término sustituido en la pregunta sin aplicar ninguna penalización a la probabilidad de los pasajes recuperados, candidatos

Figura 4.9. Ejemplo de correspondencia entre un término de la consulta y otro del corpus con ruido.

de contener la respuesta correcta. Es decir, este proceso de RI se desglosa en tantos procesos de RI como términos sean sustituidos.

En este paso, el sistema de RI obtiene la respuesta correcta para la pregunta de ejemplo en la primera posición, con un peso de 0.65 en el pasaje mostrado en la parte de abajo de la Figura 4.8. Es necesario resaltar que este pasaje no fue recuperado por el sistema de RI *baseline*.

4. Unificar y reordenar los resultados de los dos procesos de RI, para devolver el listado de respuestas. En este paso los pasajes devueltos por todos los procesos de RI son mezclados para devolver el pasaje que contiene la respuesta. Los pasajes se combinan reordenándolos de acuerdo con sus pesos, sin aplicar ninguna modificación o penalización, excepto en el caso de que dos pasajes tengan el mismo peso. En este caso se tomaría el criterio de colocar primero el pasaje que haya sido más veces devuelto por diferentes procesos, realizados en el paso anterior, de RI.

   Siguiendo con el ejemplo, el mayor valor de peso de los pasajes retornados fue de 0.34 por el proceso de RI *baseline* y 0.65 por el proceso de RI ruidos, así que el pasaje recuperado en el paso anterior es el que contiene la respuesta a la pregunta de ejemplo.

Por tanto, a lo largo de la sección hemos desarrollado un ejemplo que demuestra la efectividad de la estrategia para hacer a

cualquier sistema de RI tolerante al ruido; ya que el pasaje devuelto contiene la respuesta correcta y ocupa la primera posición en el nuevo ordenamiento. Y lo más importante es que es una estrategia totalmente independiente de la arquitectura del sistema de RI y de los tipos de ruido presentes en el corpus, no siendo para nada necesario el estudio de los mismos. Eso si, es necesario disponer de algún SOC del dominio y es, por tanto, una estrategia diseñada para entornos restringidos.

4.5. Conclusiones

Los datos en el mundo real son intrínsecamente ruidosos, por tanto, las técnicas para procesar el ruido son cruciales en sistemas de RI si se desean obtener resultados útiles y prácticos (Knoblock *et al.*, 2007). Debido a la inmensa cantidad de redundancia inherente a los corpus de dominio abierto, éstos son poco sensibles a la presencia de ruido. Sin embargo, los corpus en dominios restringidos son usualmente bastante pequeños y, por lo tanto, con menor o ninguna redundancia. Consecuentemente, los sistemas de RI son más propensos a fallar cuando utilizan corpus de dominios restringidos.

Con la finalidad de superar este problema, en este capítulo hemos presentado una aproximación para adicionar tolerancia al ruido textual en el proceso de Recuperación de Información (RI) sobre corpus de dominios restringidos, pequeños y ruidosos. El objetivo era desarrollar una estrategia de tolerancia independiente del tipo de ruido presente en el corpus y de la arquitectura del sistema de RI. Por esta razón, la propuesta está basada en la utilización de cualquier recurso de representación del conocimiento (SOC) disponible en el dominio, que funcione como vocabulario estándar e intermedio entre las palabras ruidosas y las originales. Para calcular la similitud entre las palabras empleamos una distancia de edición extendida, la cual hemos adaptado para dotarla de la capacidad de calcular la similitud entre palabras simples y multipalabras.

Para llevar a cabo el objetivo, se realizaron varias tareas principales que expondremos seguidamente. Citaremos además, las conclusiones extraídas al desarrollar cada tarea:

- Estudio de las diferentes propuestas existentes para calcular la similitud entre cadenas de caracteres (a partir del análisis de los trabajos (Levenshtein, 1966; Damerau, 1964; Needleman & Wunsch, 1970; Jaro, 1989; Winkler, 1999; Fernández *et al.*, 2009)). En dicho estudio se hizo énfasis en las características principales y desventajas de cada algoritmo. Además se profundizó en la descripción de DEx (Fernández *et al.*, 2009) por ser la distancia candidata a ser usada en nuestro trabajo.
- Elección del algoritmo de distancia de edición que empleará nuestra estrategia tolerante al ruido para sistemas de RI. Para tomar esta decisión realizamos dos experimentos (uno con familias de palabras y otro con palabras limpias y ruidosas) que sirvieron como marco de comparación entre las distancias. La DEx fue la distancia que obtuvo mejores resultados en los dos experimentos. Por tanto se confirmaba como la distancia idónea a emplearse en nuestra investigación. Luego, se profundizó en las bases que sustentan a esta distancia y se definió su principal desventaja para nuestros propósitos. La resolución de esta desventaja fue la causa de la siguiente tarea.
- Adaptación de la DEx para que hacer posible la comparación entre palabras simples y multipalabras. Precisamente, una de las principales aportaciones de nuestro trabajo ha sido dar cumplimiento a esta tarea.
- Creación de una estrategia para hacer tolerante al ruido textual a cualquier sistema de RI sobre dominios restringidos. La estrategia que definimos es independiente de la arquitectura del sistema de RI y de los tipos de ruidos que estén presentes en el corpus. Por lo que no es necesario realizar ningún estudio previo sobre el corpus para detectar patrones de ruido repetitivos, ni alcanzar un conocimiento profundo de los mismos. Pero sí es necesario disponer de algún recurso de representación del conocimiento del dominio, ya sea un diccionario, tesauro, ontologías, etc. Por tanto, tomamos como premisa en nuestra propuesta que

en los dominios restringidos siempre hay disponibilidad de algún recurso de este tipo que se puede emplear, lo cual es cierto en la mayoría de los casos.

Uno de los principales beneficios de nuestro trabajo es que el rendimiento del sistema de RI se mantiene aunque se utilice un corpus de dominio restringido ruidoso, como mostrarán los experimentos llevados a cabo dentro del dominio agrícola en el capítulo 7.

Concluir que aunque parece que nuestra propuesta es independiente del idioma y dominio de aplicación, y de la arquitectura del sistema de RI, ya que teóricamente se ha diseñado de esta manera, existe la necesidad de demostrarlo en la práctica. Por tanto, nuestro trabajo futuro inmediato se enfoca en desarrollar experimentaciones más extensas en este sentido.

Capítulo 5

Método de adaptación de un sistema de búsqueda de respuestas de dominio abierto a dominios restringidos

Cuando se intenta aplicar un Sistema de Búsqueda de Respuesta de Dominio Abierto (SBR-DA) en entornos reales, con frecuencia se necesita un mayor grado de especificidad que hace que estos sistemas sean poco precisos, requiriendo su adaptación a dominios de aplicación más restringidos. Después de analizar la situación actual con respecto a la adaptación de SBR-DA a dominios restringidos (ver Capítulo 3), parece indiscutible la necesidad de diseñar estrategias que faciliten este proceso. Principalmente, las propuestas en este sentido deben (i) elevar el grado de automatización de la adaptación, evitando así que sea un proceso tedioso y complejo, (ii) explotar los recursos de conocimiento disponibles para un dominio concreto independientemente de su esquema de representación y (iii) utilizar el corpus textual como punto de partida y fuente de información principal, en detrimento del (poco realista en dominios restringidos) uso de corpus de preguntas.

Con este fin, en este capítulo se presenta en detalle una propuesta sistemática de adaptación de SBR-DA a nuevos dominios restringidos basada en una técnica ampliamente utilizada en ingeniería del software: el desarrollo dirigido por modelos (*Model Driven Development*, MDD). La propuesta tiene como objetivo principal facilitar el diseño y puesta en marcha de Sistemas de Búsqueda de Respuesta de Dominios Restringidos (SBR-DR) de manera automática, requiriendo sólo una mínima supervisión del desarrollador del sistema.

El presente capítulo se estructura de la siguiente forma. En primer lugar, se realiza una introducción sobre MDD, se presen-

tan sus características, la terminología usada y los beneficios que puede aportar al desarrollo de SBR-DR. Las secciones siguientes describen en detalle la propuesta basada en MDD para la adaptación a dominios restringidos de SBR-DA.

5.1. Desarrollo dirigido por modelos

El desarrollo dirigido por modelos (*Model Driven Development*, MDD) es una aproximación al diseño de software basado en el modelado del sistema software y su generación automática a partir de modelos (Mellor *et al.*, 2003). Por lo tanto, MDD enfatiza dos aspectos clave: los modelos y las transformaciones entre ellos para llegar a obtener el código fuente correspondiente del sistema software.

5.1.1. Modelos

Según varias de las acepciones del diccionario de La Real Academia Española[1] un modelo es, entre otras cosas, lo siguiente:

- Arquetipo o punto de referencia para imitarlo o reproducirlo.
- Representación en pequeño de alguna cosa.
- Esquema teórico, generalmente en forma matemática, de un sistema o de una realidad compleja, como la evolución económica de un país, que se elabora para facilitar su comprensión y el estudio de su comportamiento.
- Figura de barro, yeso o cera, que se ha de reproducir en madera, mármol o metal.
- En empresas, usado en aposición para indicar que lo designado por el nombre anterior ha sido creado como ejemplar o se considera que puede serlo. Empresa modelo. Granjas modelo.
- Objeto, aparato, construcción, etc., o conjunto de ellos realizados con arreglo a un mismo diseño. Auto modelo 1976. Lavadora último modelo.
- Vestido con características únicas, creado por determinado modista, y, en general, cualquier prenda de vestir que esté de moda.

[1] http://www.rae.es

- Persona u objeto que copia el artista.

Todas estas definiciones tienen en común que un modelo (i) es una abstracción de algo que existe en la realidad, (ii) se diferencia en algo de la "cosa real" que se modela (no se tienen en cuenta todos y cada uno de los detalles o cambia el tamaño, etc.) y (iii) puede usarse como ejemplo para producir algo que existe en la realidad. A partir de estas tres características, resulta necesario para la definición de un modelo el poder determinar qué es ese "algo que existe en la realidad". Para contestar a esto se debe resaltar que un modelo debe centrarse en aquellas partes importantes de la realidad representada, desechando aspectos superfluos, con el fin de poder predecir su calidad, razonar acerca de sus propiedades específicas, comunicar sus características, etc. La realidad representada, indudablemente, depende del contexto en el que nos encontremos, por ejemplo, un edificio en arquitectura o un automóvil en ingeniería industrial. En concreto, en ingeniería del software, los modelos deben ser, ciertamente, precursores de la implementación de un sistema software, o bien pueden derivarse de un sistema software existente con el fin de comprenderlo mejor y poder adaptarlo a nuevas necesidades (Beydeda *et al.*, 2005).

Dentro del contexto de MDD, son varias las definiciones de modelo, si bien una de las más extendidas es la propuesta en (Kleppe *et al.*, 2003) donde se define un modelo como "una descripción de (parte de) un sistema escrito en un lenguaje bien definido". Por tanto, un modelo siempre está escrito en un lenguaje, ya sea lenguaje natural, un lenguaje de programación o cualquier otro. Sin embargo, con el fin de poder realizar transformaciones automáticas a partir de los modelos, en MDD se deben usar "lenguajes bien definidos". Los mismos autores describen un lenguaje bien definido como "un lenguaje con una forma (sintaxis) y significado (semántica) bien definidos, el cuál se puede interpretar de manera automática por una computadora". Pero, ¿cómo se puede definir dicho lenguaje? Tradicionalmente, si estos lenguajes eran textuales se definían mediante una gramática, por ejemplo en BNF[2]

[2] BNF o notación de Backus-Naur: siglas en inglés de *Backus Naur Form*, es una metasintaxis usada para expresar gramáticas libres de contexto, es decir, una

(Knuth, 1964) para lenguajes de programación o XML Schemas[3] o DTD[4] para XML[5] (Wojnar *et al.*, 2010), lo que cumple con el requisito de que sean interpretables de manera automática (en este caso mediante un compilador o intérprete). Sin embargo, con el fin de elevar el nivel de abstracción de los modelos maximizando su utilidad en ingeniería del software, estos suelen utilizar una sintaxis gráfica, por lo que se necesita un mecanismo diferente para definir dichos lenguajes. Este mecanismo se llama metamodelado.

Con el fin de lidiar con estos aspectos, MDD dispone de una estructura jerárquica en cuatro niveles (Atkinson & Kühne, 2003), en la cual, el nivel inferior es "una instancia" del nivel superior (excepto el nivel superior que es reflexivo por lo que consiste en "una instancia" de sí mismo). El nivel más bajo se denomina M0 y se corresponde con el sistema software *real*. En el nivel M1 se encuentra el modelo que *representa* el sistema software, mientras que el nivel M2 contiene el metamodelo al cuál se *ajusta* el modelo. Por último el nivel M3 se corresponde con el metametamodelo al cuál se *ajusta* el metamodelo del nivel M2. Este último nivel tiene su fundamento en la reflexividad ampliamente utilizada en informática, por ejemplo en bases de datos, sistemas operativos o lenguajes que contienen descripciones de sí mismos llamadas esquemas, metadatos o metaclases (Thomas, 2004). Si se ejempli-

manera de describir lenguajes formales. Se utiliza extensamente como notación para las gramáticas de los lenguajes de programación, de los sistemas de comando y de los protocolos de comunicación, etc.

[3] XML Schema: es un lenguaje de esquema utilizado para describir la estructura y las restricciones de los contenidos de los documentos XML de una forma muy precisa, más allá de las normas sintácticas impuestas por el propio lenguaje XML. Se consigue así una percepción del tipo de documento con un nivel alto de abstracción. Fue desarrollado por el *World Wide Web Consortium* (W3C).

[4] DTD: siglas en inglés de *Document Type Definition*, es una descripción (a través de la especificación de restricciones) de la estructura y sintaxis de un documento XML o SGML. Su función básica es la descripción del formato de datos, para usar un formato común y mantener la consistencia entre todos los documentos que utilicen la misma DTD.

[5] XML: siglas en inglés de *Extensible Markup Language*, es un metalenguaje extensible de etiquetas desarrollado por el W3C. Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML.

fica la estructura jerárquica de MDD mediante un programa en Pascal, su ejecución estaría en el nivel M0, mientras que el propio programa se representaría en el nivel M1 y la gramática BNF que permitiera la definición de programas en Pascal sintácticamente correctos estaría en el nivel M2. EL nivel M3 estaría ocupado mediante la gramática de BNF definida según el propio BNF.

Uno de los metamodelos más usados en MDD es UML (*Unified Modeling Language*) (UML, 2010). UML contiene facilidades para visualizar, especificar, construir y documentar un sistema software, de tal manera que se puedan modelar todos los aspectos del sistema, tales como procesos de negocio y funciones del sistema, expresiones de lenguajes de programación o esquemas de bases de datos. Por lo tanto UML se situaría como un metamodelo a nivel M2 dentro de MDD. Se debe resaltar que UML es muy adecuado para el desarrollo de software de propósito general como aplicaciones de gestión o telecomunicaciones, pero sin embargo es difícil de aplicar cuando se desarrolla software para contextos más específicos, por ejemplo para medicina, cuyo principal interés es modelar directamente el dominio de aplicación en lugar de realizar una compleja adaptación de UML al contexto específico (Thomas, 2004).

Con el fin de desarrollar metamodelos útiles para dominios concretos se usa MOF (*Meta Object Facility*) (MOF, 2006). MOF es un estándar creado para definir lenguajes de modelado de manera formal, es decir, se situaría en el nivel M3 de MDD, teniendo la capacidad de definirse a sí mismo. De hecho, MOF es el lenguaje a partir del cual se define UML (o mejor dicho, el metamodelo de UML). MOF suministra los conceptos y notación gráfica necesaria para crear metamodelos, así como funcionalidades para la definición de identificadores, tipos primitivos de datos, etc. Un ejemplo de la estructura jeráquica de MDD se muestra en la Figura 5.1.

Una misión fundamental de los modelos es que, una vez construidos, deben guiar la implementación del sistema software, lo cuál no resulta sencillo ya que hay que salvar las diferencias de abstracción entre ambos. Con este fin, las propuestas de desarrollo de software tradicionales requieren la programación a mano del código correspondiente al modelo, significando un alto grado de

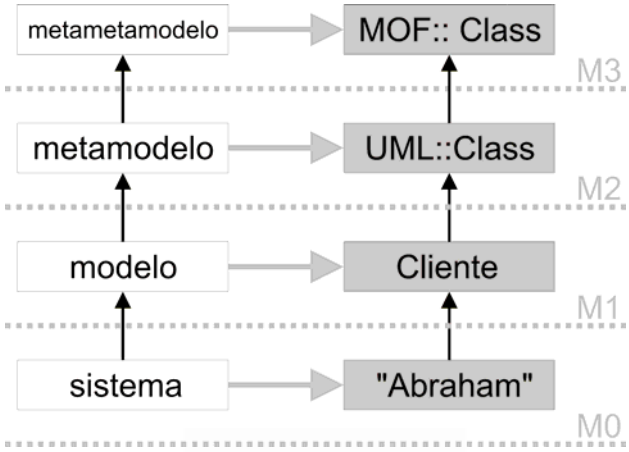

Figura 5.1. Ejemplo de estructura jerárquica de MDD.

complejidad y costo (France & Rumpe, 2007). MDD intenta reducir esta complejidad mediante el uso de tecnologías que permitan transformaciones sistemáticas entre el modelo y el sistema software, de esta manera un mismo modelo puede generar código para Java, C++, y C según las necesidades. Un objetivo primordial para poder conseguir esto es que los modelos en MDD deben ser independientes de la tecnología de implementación (computadora, *middleware*[6], interfaces gráficos, etc.).

Resulta pues indispensable en MDD el contar con mecanismos que permitan la definición de transformaciones formales entre modelos (MDA, 2003; Kleppe *et al.*, 2003). Estas transformaciones deben permitir derivar de manera automática modelos asegurando que sean correctos semánticamente (Czarnecki & Helsen, 2003; Gerber *et al.*, 2002). Además, deben ser fácilmente comprensibles, adaptables y mantenibles (Sendall & Kozaczynski, 2003). Una transformación debe estar formada por un conjunto de reglas de transformación las cuales describen cómo un modelo de entrada definido según un metamodelo origen puede convertirse

[6] Middleware: es un software de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. Funciona como una capa de abstracción de software distribuida, que se sitúa entre las capas de aplicaciones y las capas inferiores (sistema operativo y red).

en un modelo de salida según un metamodelo destino. Por lo tanto las transformaciones se definen a nivel de metamodelo (M2) pero se aplican a nivel de modelo (M1), tal y como se muestra en la Figura 5.2.

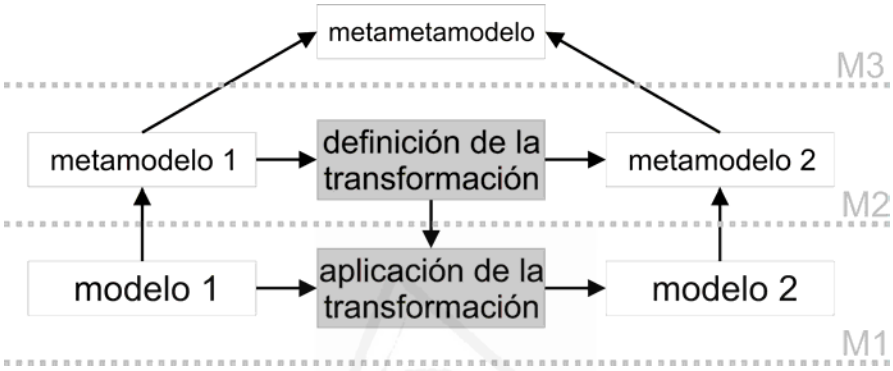

Figura 5.2. Ejemplo de transformación en MDD.

Para cumplir con estas características surge el lenguaje QVT (*Query/View/Transformation*) (QVT, 2005), un estándar desarrollado para la definición formal de transformaciones entre modelos basados en MOF. Este estándar provee mecanismos para definir reglas de transformaciones como un conjunto de relaciones que deben cumplirse entre elementos de un conjunto de modelos candidatos (origen y destino). Por lo tanto, QVT provee notación gráfica y textual para la especificación declarativa de relaciones que deben cumplirse entre elementos de metamodelos basados en MOF. Un conjunto de estas relaciones (o reglas de transformación) define una transformación entre modelos. Una relación se define por medio de los siguientes elementos (se muestra un ejemplo en la Figura 5.3):

- **Dos o más dominios**: cada dominio es un conjunto diferenciado de elementos de un metamodelo. Este conjunto de elementos se compara con los elementos del modelo candidato por medio de patrones. Un patrón de dominio puede considerarse como una plantilla para elementos de un metamodelo (incluyendo sus propiedades y sus asociaciones) que debe ser localizada, modi-

ficada o creada en un modelo candidato con el fin de satisfacer la relación. El tipo de relación entre dominios puede ser *checkonly* (marcado con C) o *enforced* (marcado con E). Cuando una relación se verifica en la dirección de un dominio *checkonly*, entonces sólo se comprueba si existe un emparejamiento válido en el modelo que satisfaga la relación (sin modificar ningún modelo si los dominios no pueden ser emparejados); mientras que para un dominio que sea *enforced*, se crean, eliminan o modifican los elementos del modelo destino con el fin de que se satisfaga la relación a nivel de metamodelo. Además, para cada dominio se debe especificar el nombre de su metamodelo subyacente.

- **Cláusula *When***: especifica las condiciones bajo las que la relación necesita cumplirse (es decir, las precondiciones).
- **Cláusula *Where***: especifica las condiciones que deben ser satisfechas por todos los elementos del modelo que participan en la relación una vez que ésta se lleva a cabo (postcondiciones).

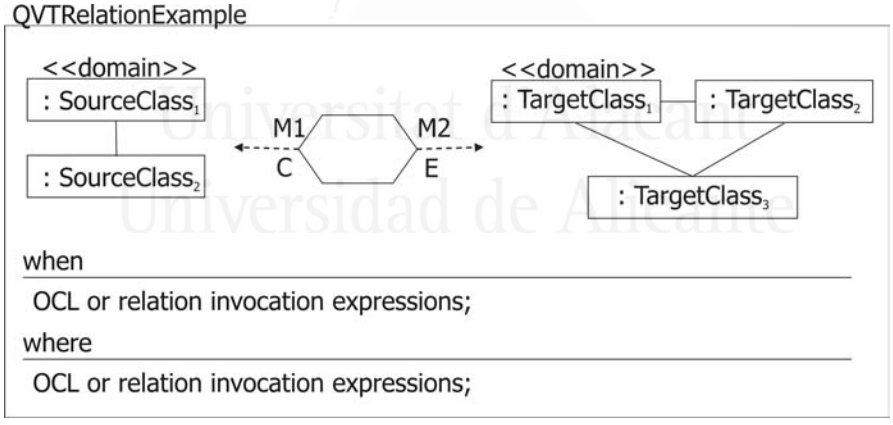

Figura 5.3. Ejemplo de relación QVT.

Con el fin de poder definir las precondiciones y postcondiciones de las cláusulas *when* y *where*, QVT debe de complementarse con un lenguaje que permita definir y evaluar expresiones sobre los modelos y metamodelos. Este lenguaje es OCL (*Object Constraint Language*) (OCL, 2010) que permite la formalización de estas expresiones de manera precisa. OCL es un lenguaje declarativo por

lo que se debe definir una expresión como una descripción del valor que se quiere calcular pero no de cómo debería calcularse.

Para hacer realidad la visión propuesta por MDD, además de las transformaciones "modelo a modelo", se necesita el concurso de otro tipo de transformaciones: aquellas que permiten obtener el código necesario a partir de un modelo. Estas transformaciones se denominan "de modelo a texto" y el lenguaje que es el estándar de facto para su diseño se denomina MOF *Model to Text Transformation Language* (*mof2text*) (MOFM2T, 2008). El lenguaje *mof2text* no sólo permite la generación de código para muchas plataformas diferentes partiendo de un mismo modelo, sino que permite la generación automática de cualquier representación textual del modelo, por ejemplo informes o archivos de configuración. Con el fin de generar representaciones textuales a partir de un modelo, *mof2text* utiliza plantillas, donde el texto a generar se parametriza con los elementos de los modelos. Concretamente, se especifica una plantilla textual que contiene parámetros que se sustituirán por datos provenientes de los modelos. Estos parámetros son expresiones que se deben especificar sobre las entidades del metamodelo y se aplicarán al modelo con el fin de seleccionar y obtener los valores correspondientes. El lenguaje *mof2text* posee una biblioteca de manipulación de cadenas que permite convertir estos valores a los fragmentos de texto requeridos. Como ejemplo, a continuación se muestra una plantilla que genera una definición de una clase de Java a partir de una clase UML:

```
[template public classToJava(c : Class)] class [c.name/]
{
    // Declaración de atributos
    [c.attribute.type.name/] [c.attribute.name/]

    // Constructor
    [c.name/]() { }
}
[/template]
```

Llegados a este punto, se deben resaltar las ventajas que aporta el uso de MDD al desarrollo o evolución de cualquier artefacto software. Quizás la mayor motivación es la mejora de la

productividad ya que la generación de código se hace de manera automática a partir de los modelos, por lo que el coste de programación decrece considerablemente. Sin embargo, existen otras ventajas (Atkinson & Kühne, 2003), por ejemplo la adaptabilidad. Los artefactos software dependen de una tecnología con la que son creados y mantenidos, por lo que es difícil su adaptación a otras tecnologías (más novedosas o necesarias por algún requisito específico). Por lo tanto, para aumentar la adaptabilidad de los artefactos software es necesario desarrollarlos de manera independiente de la tecnología de implementación y poseer mecanismos que permitan obtener fácilmente artefactos software dependientes de la tecnología a partir de artefactos independientes de ella. Esto se consigue con el uso de MDD, ya que los modelos son los artefactos de software independientes de la tecnología, mientras que el código obtenido a partir de ellos dependen de la tecnología de implementación.

Estas ventajas del uso de MDD en el desarrollo de software (como pueden ser los sistemas de Búsqueda de Respuestas) pueden contextualizarse para el caso concreto de la adaptación de SBR-DA a dominios restringidos:

1. **Productividad:** el sistema de BR se podría adaptar automáticamente al nuevo dominio mediante el diseño de transformaciones. Por lo tanto, el coste y el tiempo necesario para la adaptación del SBR-DA a un nuevo dominio decrecería.
2. **Adaptabilidad:** si surgen nuevas tecnologías o recursos de conocimiento, no sería necesario cambiar el sistema de BR completo, sino que sólo se deberían adaptar las transformaciones para obtener los modelos adecuados.
3. **Portabilidad:** los mismos modelos de patrones podrían ser automáticamente transformados en diferentes tipos de código dependiendo del sistema de BR objetivo. De esta manera se podría por ejemplo utilizar los patrones de un sistema de BR adaptados a otro sistema totalmente diferente o utilizar nuevos recursos de representación del conocimiento de manera sencilla.

4. **Reusabilidad:** las mejores prácticas en el diseño de sistemas de BR podrían ser incluidas en las transformaciones para garantizar una elevada calidad final en el sistema adaptado.
5. **Integración e Interoperabilidad:** los sistemas de BR usan recursos de conocimiento heterogéneos, por lo tanto su desarrollo necesita poder manejarlos de manera conjunta y homogénea. Usando MDD se podrían adaptar los patrones a través de la integración de diferentes recursos de conocimiento con facilidad por medio del desarrollo de un metamodelo común.

En la Figura 5.4 se muestra un ejemplo de cómo se podría usar MDD para SBR. En el nivel M0 se sitúa el código de un patrón según un SBR concreto, en el nivel M1 se encontraría el modelo del patrón independiente de cualquier SBR, mientras que en el nivel M2 estaría el metamodelo para la creación de patrones de SBR y en el nivel M3 se situaría MOF.

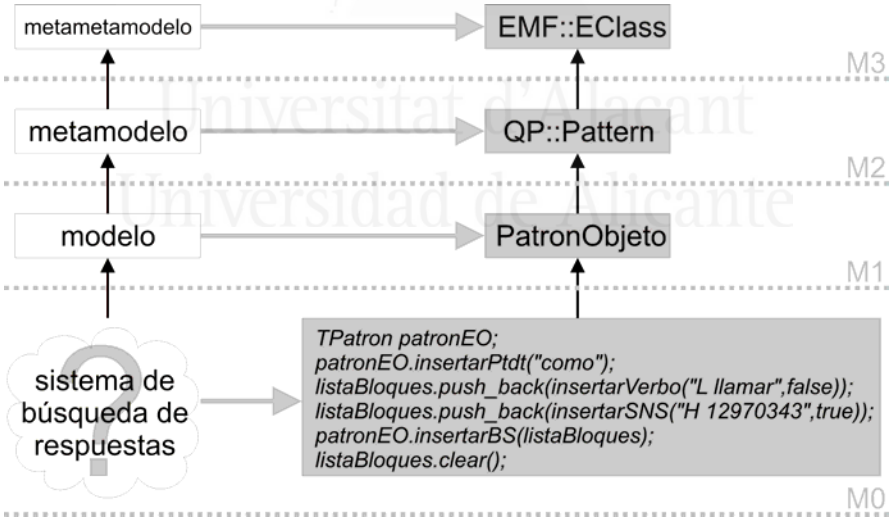

Figura 5.4. Ejemplo nuestra propuesta basada en MDD para SBR.

Finalmente, se debe resaltar que MDD sólo proporciona una estrategia general a seguir en el desarrollo de software, pero no define técnicas a utilizar, ni fases del proceso, ni ningún tipo de guía

metodológica. Por tanto, en las próximas secciones se describirán cada uno de los artefactos y transformaciones necesarios y como utilizarlos de manera sistemática en la adaptación de SBR-DA a dominios restringidos.

5.2. Adaptación dirigida por modelos de SBR a dominios restringidos

En esta sección se desarrolla un método para facilitar la adaptación de sistemas de BR-DA a dominios restringidos, empleando de manera transparente los recursos de conocimiento del dominio disponibles. Para alcanzar este objetivo se trazaron, a partir del estudio del estado de la cuestión, varios retos a superar: (i) explotación máxima de los recursos de conocimiento (SOC) disponibles independientemente del esquema de representación que sigan; (ii) creación automática de taxonomías de Tipos de Respuestas Esperadas (TRE) más refinadas sin necesidad de usar corpus de entrenamiento de preguntas, y empleando los recursos disponibles (corpus de documentos y SOC del dominio); (iii) adquisición automática de los patrones que utiliza el SBR-DA *baseline*; y (iv) creación automática de nuevos patrones adaptados al dominio, requiriendo sólo una ligera supervisión por parte del desarrollador del sistema.

Cabe destacar que en el contexto de esta investigación, se entiende por "patrones" todas las posibles estrategias (como pueden ser formas lógicas, expresiones regulares, relaciones sintácticas, relaciones de dependencia, y otras) que puede usar un sistema de BR, para detectar relaciones entre los elementos de la pregunta o de las respuestas candidatas y para hacer cumplir restricciones léxicas o semánticas a determinados elementos.

Con el fin de cumplir con todas estas características, la aproximación para la adaptación a dominios restringidos de SBR-DA descrita en esta tesis consiste en un proceso dirigido por modelos fundamentado en varios elementos: (i) una colección de documentos que representan un corpus del dominio restringido, (ii) conocimiento del dominio a través de uno o varios SOC y (iii) el SBR-DA

baseline que se desea adaptar. A partir de estos elementos los patrones de preguntas y respuestas existentes en el SBR-DA *baseline* se adaptan automáticamente al dominio restringido requerido.

En la figura 5.5 se muestra una visión global de la aproximación. En primer lugar, se obtiene un primer modelo del dominio restringido a partir de una colección de documentos del dominio (mediante la transformación T1). Este modelo se enriquece con conceptos del dominio restringido provenientes de diferentes SOC (T2). A partir de este modelo enriquecido se obtiene una taxonomía de tipos de respuesta esperada para el dominio (T3) que servirá para la adaptación de los patrones de pregunta y respuesta, cuyos modelos se adquieren del SBR-DA *baseline* (T4 y T5). Esta adaptación se realiza respectivamente en las transformaciones T6 y T7 mediante el concurso de los modelos de taxonomía de respuesta esperada y de los modelos de los patrones existentes (de pregunta y respuesta), obteniendo sendos modelos de patrones de pregunta y respuesta adaptados al nuevo dominio. La generación del código correspondiente a los patrones según el SBR-DA *baseline* se realiza en las transformaciones T8 y T9.

5.2.1. Obteniendo el modelo de dominio restringido a partir del corpus

Como nuestro método de adaptación se basa en los términos más relevantes que aparecen en el corpus, el primer paso consiste en obtener un modelo de dominio restringido (en inglés, *restricted domain model*) que contiene toda la información disponible de los términos previamente extraídos en el módulo de indexación (ver la transformación T1 en la Fig. 5.5). Este modelo debe ajustarse a un metamodelo de dominio restringido que incluya todos los elementos requeridos para especificar modelos que contengan los términos del corpus de dominio restringido y toda su información relacionada (como la información léxica, sintáctica, semántica, etc.), así como aquellos elementos que permitan enlazarlos posteriormente con sus conceptos correspondientes en los recursos del conocimiento o SOCs disponibles en ese dominio. Para alcanzar este objetivo hemos definido el metamodelo de **Domi-**

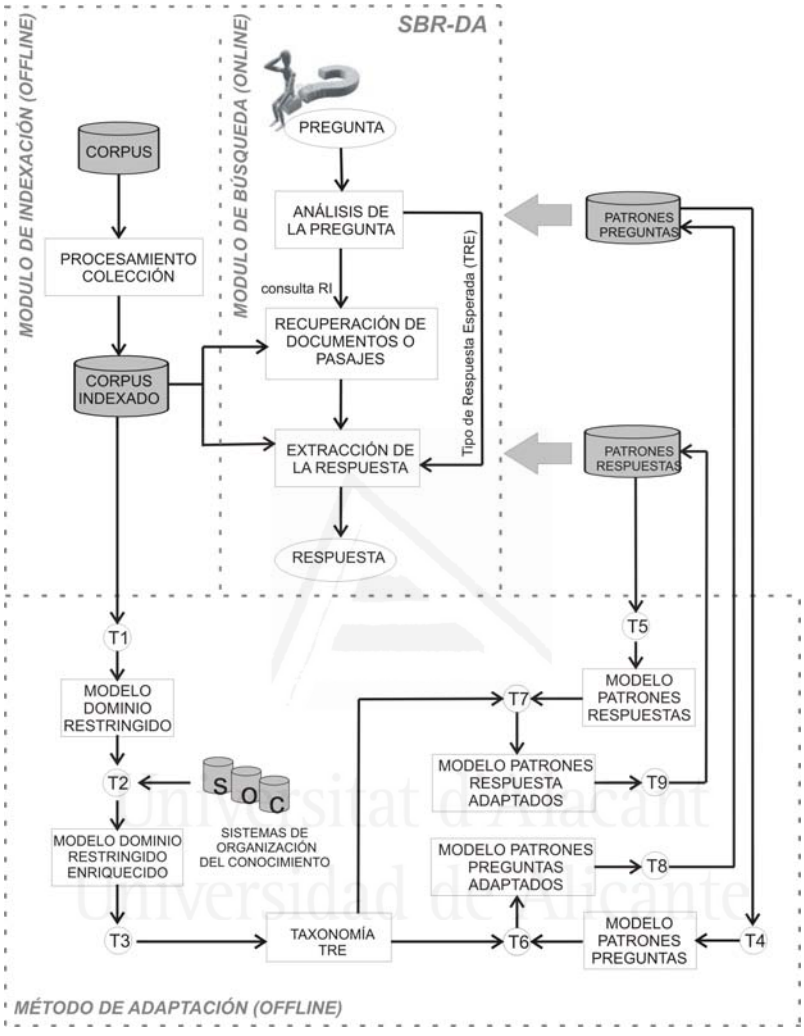

Figura 5.5. Propuesta dirigida por modelos para adaptar sistemas de BR a dominios restringidos.

nio Restringido, el cual contiene los elementos adecuados para crear tales modelos (ver el área resaltada en la Figura 5.6).

El elemento principal en este metamodelo es la clase *RestrictedDomainModel*, a partir de la cual se podrá definir cada uno de los demás elementos del modelo. Esta clase se compone de elementos de la clase *CorpusTerm*, los cuales se emplean para re-

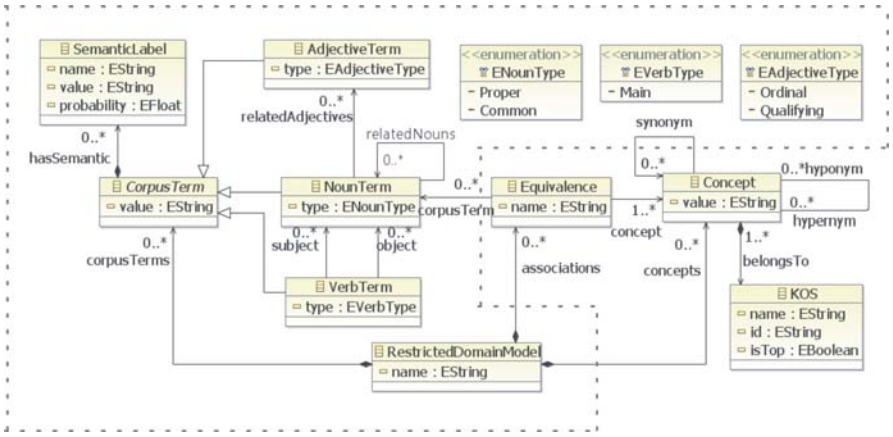

Figura 5.6. Visión general de nuestro Metamodelo de Dominio Restringido.

presentar cualquier término que aparece en el corpus. Esta clase contiene un atributo *value*, usado para indicar el valor lematizado[7] de cada término. Por otro lado cada elemento de la clase *CorpusTerm* puede corresponder a un tipo léxico de importancia para la búsqueda de respuestas: adjetivo, sustantivo o verbo. Cada uno de estos tipos está caracterizado en el metamodelo como una subclase de *CorpusTerm*, a saber, *AdjectiveTerm*, *NounTerm* y *VerbTerm*. Además, el metamodelo considera relaciones que pueden establecerse entre estos elementos: la clase *VerbTerm* tiene relaciones para indicar qué sustantivos (*NounTerm*) pueden actuar como sujeto (*subject*) u objeto (*object*) de la acción del verbo; mientras que la clase *NounTerm* puede estar relacionada (mediante *relatedAdjectives*) con varios adjetivos (*AdjectiveTerm*) u otros sustantivos (mediante *relatedNouns*), reflejando así en el metamodelo la relación sustantivo-adjetivo y sustantivo-sustantivo, las cuáles resultan de importancia para detectar las multipalabras que tan frecuentemente aparecen en dominios restringidos (por ejemplo, "hidróxido de calcio" u "hormonas suprarrenales" en el dominio

[7] Lematizar: según la diccionario de la RAE, es la acción de elegir convencionalmente una forma para remitir a ella todas las de su misma familia por razones de economía en un diccionario o repertorio léxico. Se puede definir en el contexto de la lexicografía como la concentración en un único lema de las formas de una palabra variable.

químico, o "abies alba", "abies sachalinensis" o "tracto digestivo" en el dominio agrícola, etc.). Cada una de estas relaciones pueden ser fácilmente obtenida por etiquetadores gramaticales y sintácticos usados en el procesamiento del corpus, es decir, en la fase de indexado de un sistema de BR.

El atributo *type* indica el tipo que puede tener cada una de las subclases de *CorpusTerm*. Este tipo toma valor de varias enumeraciones. Así, en la clase *AdjectiveTerm* puede tomar los valores ordinal (*Ordinal*) o calificativo (*Qualifying*) de *EAdjetiveType*, en la clase *NounTerm* los valores propio (*Proper*) o común (*Common*) se toman de *ENounType*, mientras que en *VerbTerm* se puede tomar el valor principal (*Main*) de *EVerbType*.

Finalmente, cada *CorpusTerm* puede tener alguna información semántica representada en la clase *SemanticLabel*. Esta información semántica se puede obtener de herramientas de dominio abierto usadas en la fase de indexación de BR, como pueden ser etiquetadores de roles semánticos, de entidades nombradas (en inglés, NER: *Name Entity Recognition*), de expresiones temporales o numéricas, etc. De manera detallada podemos decir que la clase *SemanticLabel* indica el nombre de la técnica utilizada para adquirir la información semántica en el atributo *name*, el valor del resultado de dicha técnica en *value* y por último, el grado de certeza de ese resultado en el atributo *probability*. Por ejemplo, para los términos "río Congo" y "lago Kariba" hay una relación semántica cuyo valor es "aguas interiores", el nombre de la técnica empleada es "NER", ya que el valor se obtiene usando un reconocedor de entidades nombradas, y la probabilidad es "1".

La transformación T1 (ver Fig. 5.5) ha sido diseñada para obtener automáticamente un modelo de dominio restringido, acorde al metamodelo anteriormente presentado, a partir de los términos más relevantes presentes en el corpus. El primer paso de esta transformación consiste en seleccionar aquellos términos que se deben considerar. Esta selección se basa en dos restricciones:

- Una restricción léxica para poder escoger aquellos términos que tienen un tipo relevante para la búsqueda de respuestas: sustantivos, verbos o adjetivos.

- Una restricción estadística para poder escoger aquellos términos que tengan cierta frecuencia, ya sea relativa (fr_i[8]) o *tf-idf* (Baeza-Yates & Ribeiro-Neto, 1999).

A partir de cada término relevante encontrado según estos criterios, se crea la clase *CorpusTerm* necesaria (*AdjectiveTerm*, *NounTerm* o *VerbTerm*), con su correspondiente información léxica, sintáctica y semántica obtenida del procesamiento del corpus en el módulo de indexación.

Ejemplo ilustrativo de la obtención del modelo de Dominio Restringido De ahora en adelante, durante todos los pasos de nuestro método de adaptación, mostraremos un ejemplo de aplicación de nuestra aproximación basado en el dominio agrícola, específicamente usando como corpus la Revista Cubana de Ciencia Agrícola (RCCA). Para este ejemplo se ha usado un único documento de este corpus, con el fin de poder explicar convenientemente cada paso de nuestra propuesta (el documento "RCCA2001_T35N02A141"), mientras que los términos más relevantes se han considerado como aquellos que se encuentran en un párrafo concreto de este documento. El párrafo es el siguiente:

"Sin embargo, en ocasiones, estos compuestos, en concentraciones bajas , pueden ejercer efectos beneficiosos. Por ejemplo, las saponinas tienen un efecto defaunante en el rumen, lo que puede contribuir al aumento de la razón energía: proteína de los productos absorbidos debido al incremento del flujo de bacterias y de aminoácidos de la dieta hacia el intestino (Díaz et al. 1993)."

Los términos más relevantes a considerar son entonces los siguientes: "compuestos", "concentraciones", "bajas", "ejercer", "efectos", "beneficiosos", "saponinas", "defaunante", "rumen", "razon", "energia", "proteina", "productos","absorbidos", "flujo", "bacterias", "aminoacidos", "dieta", "intestino" y "Diaz".

Estos términos se intentan detectar en el documento previamente procesado por un etiquetador léxico (por ejemplo, el PoS-tagger MACO (Acebo, 1994)) y por un analizador sintáctico (por

[8] $fr_i = \frac{f_i}{N}$, donde fr_i es la frecuencia relativa del término t_i en el corpus, f_i es la frecuencia absoluta calculada a partir del número de observaciones del término t_i en el corpus y N es el número total de términos en el corpus.

ejemplo, el analizador sintáctico parcial SUPAR (Ferrández *et al.*, 1999)). El objetivo es poder determinar si los términos son sustantivos, adjetivos o verbos, así como las relaciones existentes entre ellos.

Con el fin de ejemplificar claramente nuestra propuesta, nos vamos a centrar en el párrafo de ejemplo anteriormente descrito, esta vez etiquetado usando MACO y SUPAR, que sirve de entrada a la transformación T1:

```
<@OOO,21,Frase maco castellano>
<@CCC>
<@SPS>
Sin SPS00 sin NOSTEM
<@SNS,sp,sustAdj,,>
<@NSN>
embargo NCMS000 embargo NOSTEM
<@/NSN>
<@/SNS,sp,sustAdj,,>
<@/SPS>
, Fc , NOSTEM
<@SPS>
en SPS00 en NOSTEM
<@SNS,sp,sustAdj,,>
<@NSN>
ocasiones NCFP000 ocasión NOSTEM
<@/NSN>
<@APO>
, Fc , NOSTEM
<@SNS,,sustAdj,,>
estos DDOMP0 este NOSTEM
<@NSN>
compuestos NCMP000 compuesto NOSTEM
<@/NSN>
<@/SNS,,sustAdj,,>
, Fc , NOSTEM
<@/APO>
<@/SNS,sp,sustAdj,,>
<@/SPS>
<@SPS> en SPS00 en NOSTEM
<@SNS,sp,sustAdj,,>
<@NSN> concentraciones NCFP000 concentración
NOSTEM
<@/NSN>
<@ADJ>
bajas AQ0FP0 bajo NOSTEM
<@/ADJ>
<@/SNS,sp,sustAdj,,>
<@/SPS>
, Fc , NOSTEM
<@VBC> pueden VMIP3P0 poder NOSTEM
ejercer VMN0000 ejercer NOSTEM
<@/VBC>
<@SNS,,sustAdj,,>
<@NSN>
efectos NCMP000 efecto NOSTEM
<@/NSN>
<@ADJ>
beneficiosos AQ0MP0 beneficioso NOSTEM
<@/ADJ>
<@/SNS,,sustAdj,,>
. Fp . NOSTEM
<@/CCC>
<@/OOO,21,Frase maco castellano>
```

```
<@OOO,22,Frase maco castellano>
<@CCC>
Por_ejemplo RG por_ejemplo NOSTEM
, Fc , NOSTEM
<@SNS,,sustAdj,,>
las DA0FP0 el NOSTEM
<@NSN>
saponinas NCFP000 saponina NOSTEM
<@/NSN>
<@/SNS,,sustAdj,,>
<@VBC>
tienen VMIP3P0 tener NOSTEM
<@/VBC>
<@SNS,,sustAdj,,>
un DI0MS0 uno NOSTEM
<@NSN>
efecto NCMS000 efecto NOSTEM
<@NSN>
defaunante NC00000 defaunante NOSTEM
<@/NSN>
<@/NSN>
<@SPS>
en SPS00 en NOSTEM
<@SNS,sp,sustAdj,,>
el DA0MS0 el NOSTEM
<@NSN> rumen NCMS000 rumen NOSTEM
<@/NSN>
<@/SNS,sp,sustAdj,,>
```

```
<@/SPS>
<@/SNS,,sustAdj,,>
, Fc , NOSTEM
lo DA0NS0 el NOSTEM
<@ORL>
que PR0CN000 que NOSTEM
<@VBC>
puede VMIP3S0 poder NOSTEM
contribuir VMN0000 contribuir NOSTEM
<@/VBC>
<@SPS>
al SPCMS al NOSTEM
<@SNS,sp,sustAdj,,>
<@NSN>
aumento NCMS000 aumento NOSTEM
<@/NSN>
<@SPS>
de SPS00 de NOSTEM
<@SNS,sp,sustAdj,,>
la DA0FS0 el NOSTEM
<@NSN>
razon NCFS000 razon NOSTEM
<@NSN>
energia NCFS000 energia NOSTEM
<@/NSN>
<@/NSN>
<@/SNS,sp,sustAdj,,>
<@/SPS>
<@/SNS,sp,sustAdj,,>
<@/SPS>
<@/ORL>
<@/CCC>
<@CCC>
: Fd : NOSTEM
<@SNS,,sustAdj,,>
<@NSN>
proteina NCFS000 proteina NOSTEM
<@/NSN>
<@SPS>
de SPS00 de NOSTEM
<@SNS,sp,sustAdj,,>
los DA0MP0 el NOSTEM
<@NSN>
productos NCMP000 producto NOSTEM
<@/NSN>
<@/SNS,sp,sustAdj,,>
<@/SPS>
<@/SNS,,sustAdj,,>
<@VBC>
absorbidos VMP00PM absorber NOSTEM
<@/VBC>
<@SPS>
debido_al SPCMS debido_al NOSTEM
<@SNS,sp,sustAdj,,>
<@NSN>
incremento NCMS000 incremento NOSTEM
<@/NSN>
<@SPS>
del SPCMS del NOSTEM
<@SNS,sp,sustAdj,,>
<@NSN>
flujo NCMS000 flujo NOSTEM
<@/NSN>
<@SPC>
<@SPS>
de SPS00 de NOSTEM
<@SNS,sp,sustAdj,,>
<@NSN>
bacterias NCFP000 bacteria NOSTEM
<@/NSN>
<@/SNS,sp,sustAdj,,>
<@/SPS>
y CC y NOSTEM
<@SPS>
de SPS00 de NOSTEM
<@SNS,sp,sustAdj,,>
<@NSN>
aminoacidos NCMP000 aminoacido NOSTEM
<@/NSN>
<@SPS>
de SPS00 de NOSTEM
<@SNS,sp,sustAdj,,>
la DA0FS0 el NOSTEM
<@NSN>
dieta NCFS000 dieta NOSTEM
<@/NSN>
<@SPS>
hacia SPS00 hacia NOSTEM
<@SNS,sp,sustAdj,,>
el DA0MS0 el NOSTEM
<@NSN>
intestino NCMS000 intestino NOSTEM
<@/NSN>
<@/SNS,sp,sustAdj,,>
<@/SPS>
<@/SNS,sp,sustAdj,,>
<@/SPS>
<@/SNS,sp,sustAdj,,>
<@/SPS>
<@/SPC>
<@/SNS,sp,sustAdj,,>
<@/SPS>
<@/SNS,sp,sustAdj,,>
<@/SPS>
( Fpa ( NOSTEM
<@SNS,,sustAdj,,>
<@NSN>
Diaz NP00000 Diaz NOSTEM
<@NSN>
et NC00000 et NOSTEM
<@/NSN>
<@/NSN>
```

```
<@/SNS,,sustAdj,,>                    <@/CCC>
al SPCMS al NOSTEM                    <@/OOO,22,Frase maco castellano>
. Fp . NOSTEM
```

Los términos marcados por MACO con etiquetas que comienzan por "NC (nombre común o *common noun*)" o "NP (nombre propio o *proper noun*)" se especifican en el modelo de dominio restringido como elementos de la clase *NounTerm* con el tipo "NC" o "NP", respectivamente. Los términos cuya etiqueta asignada comienza por "VM (verbo principal o *main verb*)" se especifican con elementos de la clase *VerbTerm* con el tipo "VM". Finalmente los términos con etiquetas que empiezan por "AQ (adjetivo calificativo o *qualifying adjective*)" o "AO (adjetivo ordinal u *ordinal adjective*)" se modelan como *AdjectiveTerm* con el tipo "AQ" o "AO", respectivamente. Por otra parte, las relaciones sintácticas entre los términos se obtienen teniendo en cuenta las etiquetas de SUPAR "SNS (sintagma nominal simple o *nominal simple syntagma*)" y "CCC (cláusula de cada oración o *clauses of every sentences*)" de la siguiente manera: (i) un sustantivo y un adjetivo dentro de la misma etiqueta "SNS" se relacionan mediante el atributo *Related Adjectives* del sustantivo; (ii) los sustantivos dentro del mismo "SNS" se relacionan mediante el atributo *Related Nouns* y (iii) un sustantivo y un verbo dentro del mismo "CCC" se relacionan por medio de los atributos *Subject* y *Object*, según la función que tenga el sustantivo respecto al verbo.

En el párrafo etiquetado anteriormente se observa, por ejemplo, que "concentraciones" es un sustantivo y que está relacionado con "bajas" que es un adjetivo. Por otro lado, el término "pueden ejercer" se etiqueta como un verbo, estando asociado con el sustantivo "efectos", que a su vez se relaciona con el adjetivo"beneficiosos". Para cada uno de los términos frecuentes (más exactamente para su forma lematizada) se crea un elemento en el modelo de dominio restringido acorde a su clase, a saber, NounTerm, AdjectiveTerm o VerbTerm. La Figura 5.7 representa parte de este modelo, donde se pueden observar los términos previamente descritos y las relaciones entre ellos.

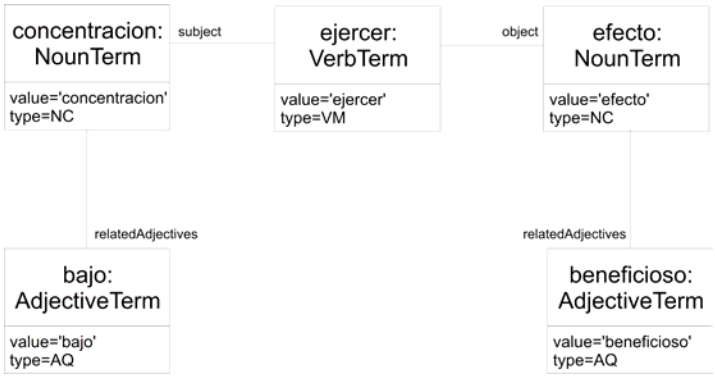

Figura 5.7. Parte del modelo de dominio restringido de nuestro ejemplo.

5.2.2. Enriquecimiento del modelo de dominio restringido.

El modelo de dominio restringido creado con la transformación anterior debe considerar información semántica relativa a los conceptos y sus relaciones presentes en varios SOC, para crear un modelo de **Dominio Restringido Enriquecido** (en inglés, *enriched restricted domain model*). Para ello se ha desarrollado la transformación T2 (ver Fig. 5.5) que permite añadir información proveniente de diferentes tipos de SOC de manera integrada (desde una simple taxonomía hasta una compleja ontología). La razón es que nuestro metamodelo es lo suficientemente completo para poder especificar en un modelo aquellas partes del SOC que son útiles para definir una taxonomía de TRE para el dominio restringido, abstrayendo los detalles innecesarios. De esta manera se puede manejar la heterogeneidad de los SOC para asegurar la integración e interoperabilidad entre ellos mediante su incorporación a un modelo basado en nuestro metamodelo común. Además, cabe destacar que esta transformación hace al sistema adaptable ya que si se necesita considerar información procedente de otro SOC, con diferente esquema de representación, no hace falta cambiar todo el sistema de BR; ya que sólo se hace necesario adaptar T2 para el nuevo SOC.

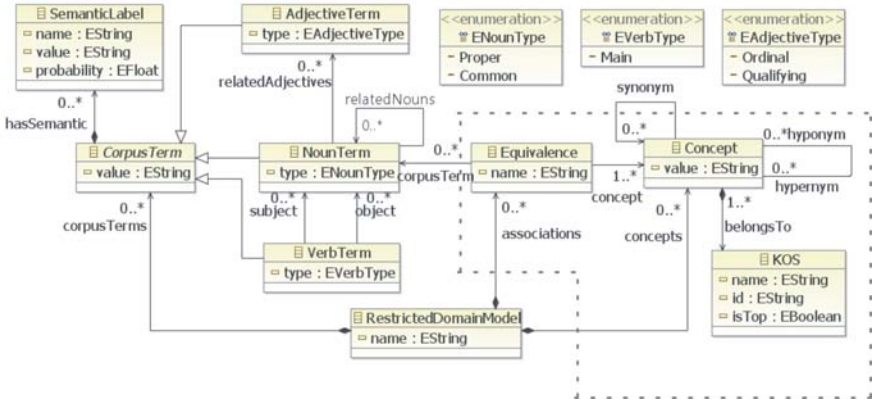

Figura 5.8. Visión general de nuestro Metamodelo de Dominio Restringido Enriquecido.

Para poder tener en cuenta esto, se han introducido nuevas clases en el metamodelo de Dominio Restringido presentado anteriormente (ver el área resaltada de la figura 5.8). En concreto, la clase *ResrictedDomainModel* se compone de las clases *Concept* y *Equivalence* para permitir el enriquecimiento semántico de los elementos del metamodelo de **Dominio Restringido** (*Restricted Domain*) con los conceptos (y sus relaciones) provenientes de algún SOC. La clase *Concept* hace referencia a un elemento de un SOC determinado, cuyo valor está representado con un atributo *value*. Además, cada concepto puede estar relacionado con uno o más conceptos a través de relaciones semánticas, por ejemplo de sinonimia (*synonym*), hiperonimia (*hypernym*) o hiponimia (*hyponym*). Cada concepto puede aparecer en más de un SOC a través de su relación con la clase *KOS* (siglás del inglés *Knowledge Organization Systems*). Sus atributos *name* e *ID* hacen referencia al nombre del SOC y al identificador del concepto dentro de ese SOC, respectivamente. Esta clase *KOS* tiene también un atributo llamado *isTop* que establece si el concepto es un concepto tope (en inglés, *top concept*) en ese SOC.

Con el objetivo de poder enriquecer los términos del corpus con elementos de los SOC y sus relaciones de sinonimia, hiperonimia e hiponimia, se crea la clase *Equivalence*, para representar

una asociación entre *Concept* y *NounTerm*. El metamodelo refleja que un concepto puede generalizar varios sustantivos, por ejemplo el concepto "insectos" puede estar relacionado con los sustantivos "insectos chupadores", "insectos nocivos", e "insectos acuáticos". Por otro lado, un sustantivo puede encontrarse dentro de varios conceptos similares, por ejemplo el sustantivo "control" está asociado a los conceptos "control biológico", "control químico" y "control de enfermedades" en el SOC, porque el concepto "control" no existe de forma exacta en ese SOC.

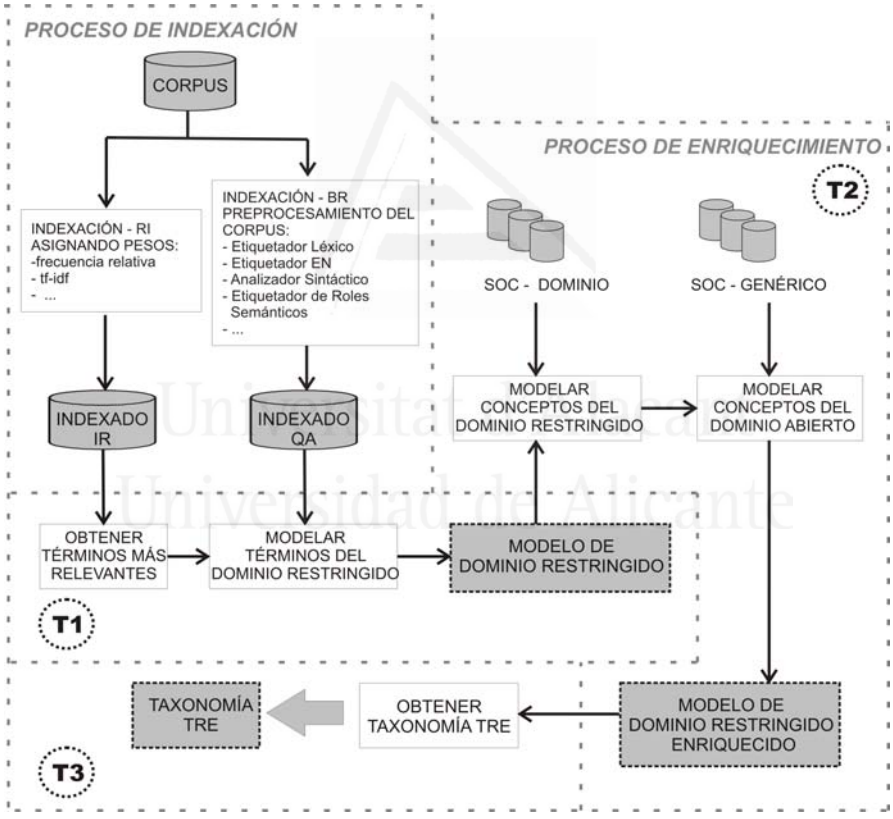

Figura 5.9. Descripción de los pasos para obtener el modelo de Dominio Restringido y la taxonomía de TRE.

Como se puede apreciar en la figura 5.9 la transformación T2 asocia cada término del corpus previamente detectado (cla-

ses *NounTerm*) con algún concepto del SOC de dominio, creando una nueva clase *Concept* y una clase *Equivalence* en el modelo de dominio restringido. Cabe destacar que también se tienen en cuenta las multipalabras cuando un elemento de la clase *NounTerm* está relacionado con algún otro *NounTerm* o con un *AdjectiveTerm*. Para cada uno de estos nuevos conceptos, el siguiente paso es la búsqueda de sus sinónimos, hipónimos e hiperónimos en el SOC de dominio hasta alcanzar un concepto tope (es decir, aquél que no tenga ningún hiperónimo). Para cada uno de estos conceptos se incluye una nueva clase *Concept* en el modelo de dominio restringido enriquecido.

A continuación cada concepto tope del SOC de dominio se intenta asociar con algún concepto de un SOC genérico, o en su defecto se trata de asociar alguno de sus sinónimos o hipónimos al SOC genérico, creándose las clases *Concept* correspondientes. Entonces, de manera análoga al SOC de dominio, para cada concepto perteneciente a un SOC genérico se adicionan como clases *Concept* al modelo de dominio sus hiperónimos y sus sinónimos hasta hallar un concepto tope.

Realizar primero el emparejamiento de las clases *NounTerm* del modelo de dominio restringido con el SOC de dominio para luego realizar la adición de conceptos del SOC genérico, tiene como objetivo evitar la introducción de polisemia, ya que las palabras que pueden presentar ambigüedad en el SOC genérico están usualmente desambiguadas en el SOC de dominio. Además, si se usara directamente el SOC genérico se correría el riesgo de sobrecargar el modelo de dominio restringido –y por tanto los pasos siguientes de nuestra propuesta como la obtención de la taxonomía de TRE– con conceptos que raramente se usan dentro del dominio en cuestión. Por otro lado, es necesario emparejar los conceptos topes del SOC de dominio adicionados al modelo con el SOC genérico, para alcanzar siempre que se pueda esa referencia y así se podrá adaptar la taxonomía de TRE (mayormente formada por conceptos genéricos) y los patrones existentes en el sistema de BR-DA a los requisitos de información del nuevo dominio restringido.

Ejemplo ilustrativo de la obtención del modelo de Dominio Restringido Enriquecido Siguiendo con nuestro ejemplo, se chequea si el término del corpus "saponinas" aparece en el SOC de dominio (en este caso el tasauro Agrovoc para el dominio agrícola). Es hallado con código 6795, por tanto se crea un nuevo elemento de la clase *Concept* "saponinas" y también un nueva equivalencia entre este concepto y el *NounTerm* "saponinas". Además, se crean conceptos para todos los hipónimos de "saponinas", esto es "glicoalcaloides" y "ginsenosidos" (ver Figura 5.10).

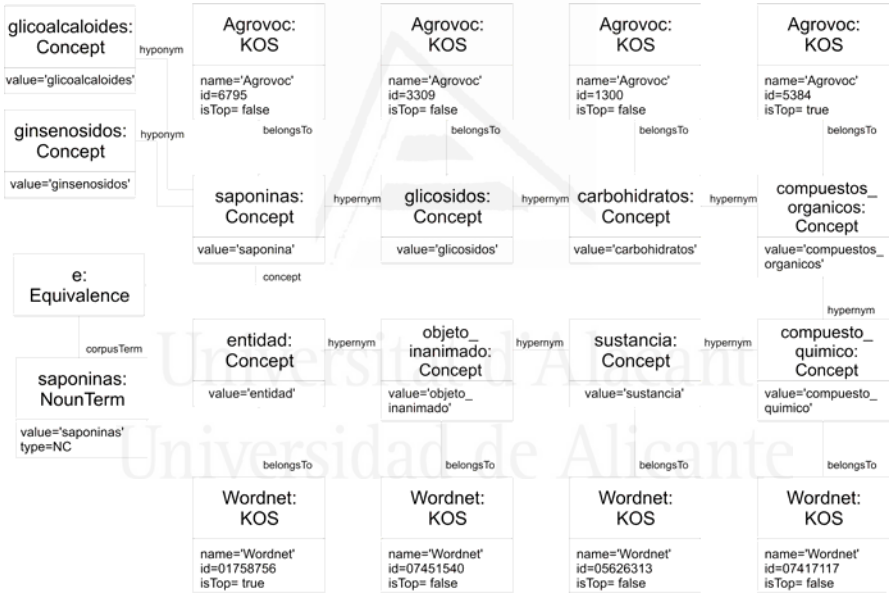

Figura 5.10. Ejemplo del enriquecimiento de parte del modelo de dominio restringido de nuestro ejemplo.

Entonces, se navega por el SOC a través de las relaciones con los términos más genéricos (en inglés, *broader term relationships*) con el fin de encontrar aquellos conceptos que son hiperónimos. De esta forma se encuentran el concepto "glicosidos" (ver Figura 5.10). A partir de estos conceptos se obtendría toda la jerarquía de hiperónimos, creando los elementos de la clase *Concept*

correspondiente hasta llegar a un concepto tope en Agrovoc, estos serían: "carbohidratos" y "compuestos_organicos" tal y como se muestra en la Figura 5.10. Seguidamente, este concepto tope se pretende emparejar con algún concepto en el SOC genérico (WordNet en este caso) y así se encuentra "compuesto_quimico" con *synset* 07417117. Entonces, recorriendo las relaciones de hiperonimia en el SOC genérico se encuentran los conceptos "sustancia", "objeto_inanimado" y "entidad". Todos estos nuevos conceptos se crean en el modelo de dominio enriquecido (ver el ejemplo en la Fig. 5.10).

El modelo de dominio restringido enriquecido se usará para adaptar los patrones existentes en el SBR-DA al dominio restringido, previa obtención de una nueva taxonomía de TRE adaptada al dominio como se mostrará a continuación.

5.2.3. Obteniendo taxonomías de TRE a partir del Modelo de Dominio Restringido enriquecido

Una taxonomía de TRE para dominio restringido puede obtenerse a partir del modelo de Dominio Restringido enriquecido obtenido en los pasos previos, aplicando la transformación T3 de la figura 5.5. Dentro de esta transformación se pueden aplicar diferentes criterios para obtener una TRE más o menos refinada según las necesidades. Estos criterios nos permiten ajustar el nivel de granularidad [9] de la taxonomía de TRE. Por ejemplo, (i) si se elige un criterio de granularidad gruesa, como puede ser "incluir en la taxonomía de TRE sólo aquellos conceptos que no tengan ningún hiperónimo", entonces se obtendrá una taxonomía genérica, (ii) si se define un criterio de granularidad fina, como sería "incluir en la taxonomía de TRE todos los conceptos que tengan un número de hipónimos mayor que N", entonces se obtendrá una taxonomía más refinada. Un criterio de granularidad fina es más apropiado para la BR-DR, ya que es aconsejable que estos tipos de taxonomías estén bien refinadas para mejorar la precisión del

[9] Granularidad: nivel de descomposición o el grado en que puede ser dividido un contenido. Entiéndase por contenido aquellos trozos de información u objetos (pueden ser textos, imágenes, gráficos, videos, etc.) a organizar, estructurar y clasificar.

proceso de BR. Sea cual sea el criterio elegido, la transformación T3 toma como entrada un modelo de dominio restringido enriquecido (es decir, la salida de la transformación T2), dando como resultado un submodelo formado con aquellos conceptos (y sus conceptos relacionados según el metamodelo de dominio restringido) que cumplan con el criterio de granularidad elegido. En la figura 5.9 se resumen los pasos para obtener una taxonomía de TRE para dominios restringidos siguiendo nuestra propuesta.

Por tanto, es esencial fijar el nivel de granularidad requerido para la nueva taxonomía de TRE. Se debe hacer notar que la elección de este nivel de granularidad es dependiente de las características del nuevo dominio a adaptar. Además vale destacar, que el nivel mínimo de granularidad de un contenido es aquel grado de descomposición en el que una determinada información sigue manteniendo su significación comunicativa, o el grado en el que físicamente no puede seguir descomponiéndose; en nuestro caso serán los nodos hoja en la cadena jerárquica, es decir, los conceptos que no tengan más hipónimos. En nuestro caso de ejemplo se ha elegido un criterio de granularidad que permita elegir todos aquellos conceptos con más de dos hipónimos para que formen parte de la taxonomía de TRE.

La transformación T3 se ha diseñado mediante el lenguaje QVT descrito anteriormente. Tanto el modelo de entrada como el modelo de salida de esta transformación se basa en el metamodelo para dominios restringidos desarrollado, ya que el modelo de salida proviene de la aplicación de cierto criterio a los conceptos del modelo enriquecido de dominio restringido. En concreto el criterio elegido es considerar aquellos conceptos que tengan más de dos hipónimos (incluyendo toda su jerarquía de hiperonimia).

Para llevar a cabo la transformación T3 se han diseñado tres reglas de transformación: *RestrictedDomainmodel2EAT*, *NewConcept4EAT*, y *Concept2Concept4EAT*.

La regla *RestrictedDomainmodel2EAT* se puede observar en la Figura 5.11. En la parte izquierda de la regla existe un único elemento de la clase *RestrictedDomainModel*. Cuando se encuentra este elemento en el modelo de entrada se crea en el modelo de salida un elemento de la clase *RestrictedDomainModel* con el

mismo nombre. En la cláusula *where* se indica que, una vez esta regla de transformación se cumpla, se debe ejecutar la regla *Concept2Concept4EAT*.

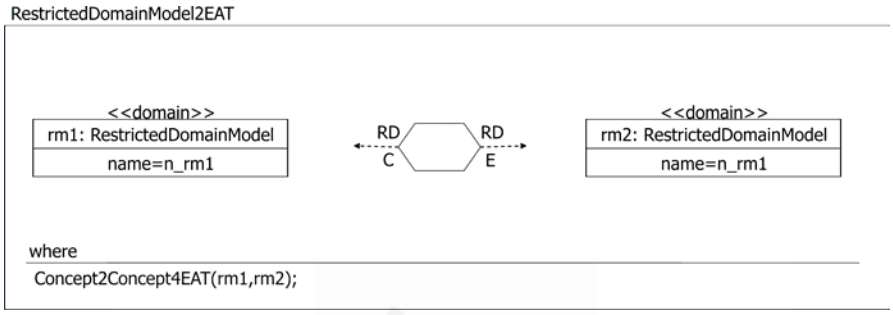

Figura 5.11. Regla de transformación que permite iniciar la obtención de un modelo para la taxonomía de TRE.

La regla *Concept2Concept4EAT* (ver Figura 5.12) tiene el objetivo de determinar aquellos conceptos de nivel más alto de la taxonomía de TRE, a partir del cuál se determinarán los demás conceptos de la taxonomía. Para este fin, en la cláusula *when* se especifica una precondición: el concepto emparejado en el modelo de entrada no debe tener hiperónimos. Además, es aquí donde se especifica el criterio que determina la granularidad de la taxonomía TRE mediante una precondición. En este caso, el criterio elegido es que el concepto debe tener algún descendiente con más de dos hipónimos, según el criterio elegido (la función *hasDescendentsWithMoreThan2Hyponyms* comprueba este criterio tal y como se observa en la Figura 5.13). Si el elemento de la clase *Concept* emparejado en el modelo de entrada cumple con estas precondiciones se creará en el modelo de salida el correspondiente elemento de la clase *Concept*. Por último, una vez llevada a cabo esta regla de transformación, se debe ejecutar la regla *NewConcept4EAT* según se especifica en la cláusula *where*. Cabe destacar que si se require otro criterio para obtener otra granularidad, lo único que habría que hacer es modificar la función *hasDescendentsWithMoreThan2Hyponyms*.

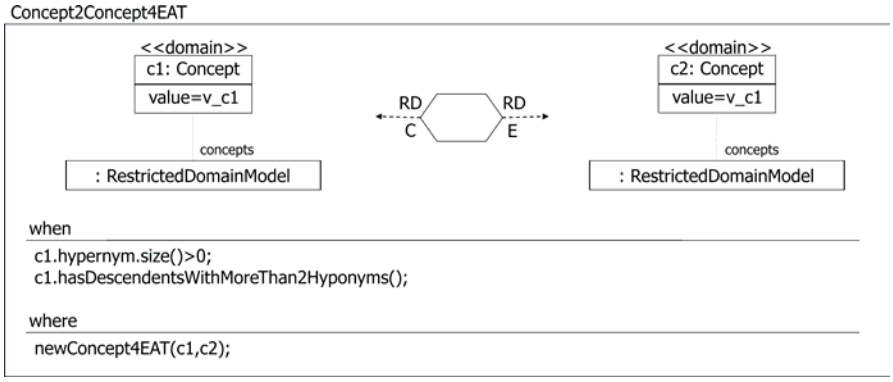

Figura 5.12. Regla de transformación que permite obtener los conceptos tope para la taxonomía de TRE.

```
helper context Concept def : hasDescendentsWithMoreThan2Hyponyms() : Boolean =
    if self.hyponym.size()=0 then
        false
    else
        if self.hyponym.size()>2 then
            true
        else
            self.hyponym->exists(c|c.hasDescendentsWithMoreThan2Hyponyms())
        endif
    endif;
```

Figura 5.13. Definición en OCL del criterio de granularidad de la taxonomía de TRE.

La regla *NewConcept4EAT* se muestra en la Figura 5.14. Para cada hipónimo del concepto anteriormente emparejado por la regla *Concept2Concept4EAT* en el modelo de entrada, la regla *NewConcept4EAT* crea un elemento de la clase *Concept* que será hipónimo del elemento de la clase *Concept* creado por la regla anterior en el modelo de salida. Esta regla se ejecuta siempre y cuando el elemento de la clase *Concept* emparejado cumpla con el criterio de granularidad, es decir, tenga algún descendiente con más de dos hipónimos (tal y como asegura la función *hasDescendentsWithMoreThan2Hyponyms* en la cláusula *when*). Con el fin de completar la taxonomía con los hipónimos del nuevo concepto

creado por esta regla, en la cláusula *where* se vuelve a ejecutar con los nuevos conceptos.

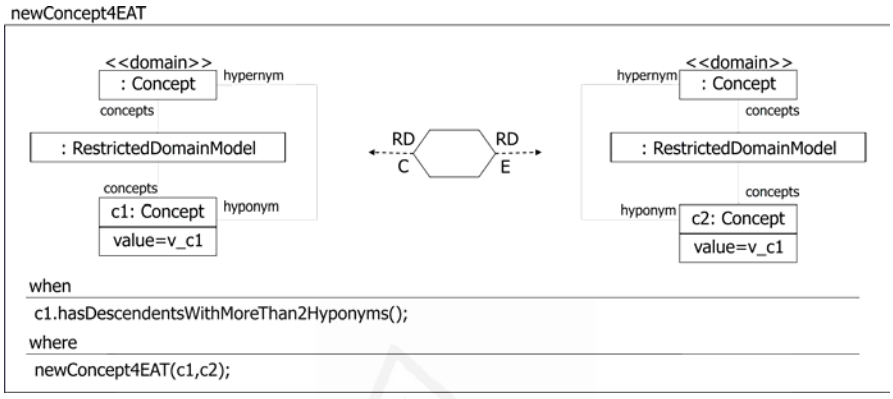

Figura 5.14. Regla de transformación que permite obtener los conceptos hipónimos de los nuevos conceptos de la taxonomía de TRE.

Ejemplo ilustrativo de la obtención de la taxonomía de TRE Para ilustrar este paso de nuestra propuesta consideraremos el siguiente ejemplo de pregunta acerca del dominio agrícola: *¿Qué glicosidos tienen un efecto defaunante en el rumen?*.

Si intentamos responder esta pregunta con un SBR-DA *baseline* para el español, como puede ser AliQAn (Roger *et al.*, 2008), la pregunta sería clasificada como *profesion*. Esto es debido a que AliQAn tiene una taxonomía de TRE con dos niveles, basada en los tipos base de WordNet y los conceptos topes de EuroWordNet, específicamente contiene las categorías que se muestran en la parte *TTRE-DA* de la Fig. 5.15 como una de las taxonomías de TRE más extendidas en SBR-DA, excepto por los tipos *causa-efecto*, y *razon* para los que no está capacitado para brindar una respuesta.

Por tanto AliQAn clasifica incorrectamente la pregunta de ejemplo, ya que su taxonomía de TRE no incluye conceptos como *glicosido*, ni siquiera ninguno de sus hiperónimos en la parte etiquetada como *TTRE-DA* de la Figura 5.15.

A continuación se desarrollará la secuencia de pasos necesarios para llegar a definir la nueva taxonomía de TRE necesaria (ver

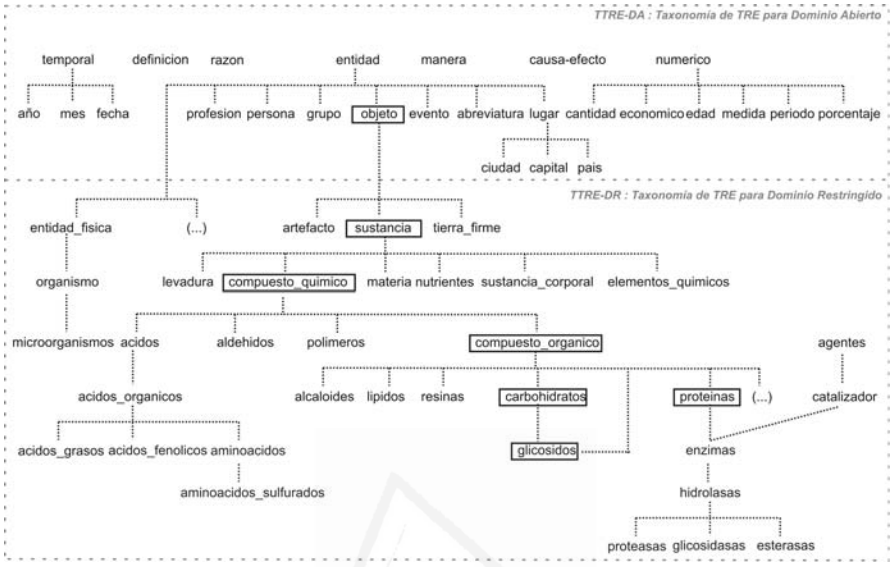

Figura 5.15. Fragmento de la taxonomía de TRE para dominio abierto y restringido.

la figura 5.9). Es decir, veremos cómo se lleva a cabo la transformación T3 una vez realizadas en las secciones anteriores los pasos T1 y T2. Además se resumirán algunos datos estadísticos de los conceptos que han ido incorporándose en cada paso de nuestra propuesta.

Siguiendo con el fragmento de ejemplo usado a lo largo de este capítulo y después de aplicar la transformación T2 se obtiene un modelo de dominio restringido de 845 nuevos conceptos (748 de Agrovoc, 415 de Wordnet y 318 presentes en ambos SOC) y 11 niveles de acuerdo con su estructura jerárquica fijada por la relación de hiperonimia-hiponimia. A partir de este modelo de dominio restringido y aplicando la transformación T3 se obtendría una taxonomía con 9 niveles y 66 conceptos (ver un fragmento de dicha taxonomía por medio de los rectángulos en la Fig. 5.15).

Siguiendo esta taxonomía, la pregunta de nuestro ejemplo se clasificaría como *glicosidos* y el espacio de búsqueda estará sólo restringido a tipos de glicosidos (como son las *saponinas*, *nucleosidos*, *nucleotidos*, *glucosidos*, *glicosidos cianogenicos* o sus hipóni-

mos) los cuales podrán ser aceptados como respuestas correctas. Finalmente, usando esta taxonomía de TRE ajustada al dominio, la respuesta a la pregunta Q1 podrá ser "*saponinas*". Además, esta taxonomía de TRE será útil para dar respuesta a preguntas mucho más específicas, por ejemplo una versión más refinada de la pregunta: *"¿Qué saponinas tienen un efecto defaunante en el rumen?"*. En ese caso las respuestas podrían ser algunos de sus hipónimos: *glicoalcaloides* o *ginsenosidos*.

Si ponemos otra pregunta como ejemplo, el procedimiento sería el mismo. Por ejemplo, *"¿Qué enzima aumenta la digestibilidad del fósforo orgánico por parte de los animales?"*. Usando la taxonomía de TRE de AliQAn, la clasificación de esta pregunta es como *objeto*. Aunque la clasificación de esta pregunta puede ser considerada correcta ya que *enzima* tiene como hiperónimo el concepto tope *objeto inanimado* (en la figura 5.15 se puede ver el camino completo de hiperonimia resaltando también por rectángulos). Sin embargo, el concepto *objeto inanimado* es demasiado amplio provocando que el sistema de BR pueda aceptar respuestas candidatas incorrectas, ya que serían semánticamente correctas, por ejemplo *artefacto*, *ácidos* o *lípidos*, ya que todas ellas se consideran objetos.

Una vez obtenida la nueva taxonomía de TRE para el dominio agrícola que nos ocupa, esta pregunta puede clasificarse como *enzimas* y el espacio de búsqueda estará sólo restringido a tipos de enzimas (como son las *hidrolasas* o sus hipónimos) los cuales podrán ser aceptados como respuestas correctas. Cabe mencionar que, usando esta taxonomía de TRE ajustada al dominio, la respuesta a la pregunta Q1 podrá ser "*fitasa*".

Con estos ejemplos hemos mostrado la importancia de nuestra propuesta para el desarrollo de nuevas taxonomía de TRE adaptadas al dominio y la necesidad de dichas taxonomías para mejorar el rendimiento de los SBR en dominios restringidos.

5.2.4. Obteniendo modelos de patrones de preguntas y respuestas

Para poder adaptar los patrones existentes a un nuevo dominio es necesario primero adquirirlos desde el sistema de BR existente. La transformación T4 y T5, en la figura 5.5, son las responsables de obtener los patrones de preguntas y respuestas existentes respectivamente. Estos patrones se representarán en un modelo de patrones de preguntas y respuestas (en inglés, *question pattern model* o *answer pattern model*) a partir del sistema BR-DA *baseline* usando dos metamodelos desarrollados para representar los patrones en modelos.

En primer lugar hemos definido el metamodelo de **Patrones de Preguntas** (en inglés, *Question Pattern*) el cual contiene los elementos adecuados para crear una variedad de estos modelos, tal como conceptos y tipo de respuesta que definirán la tipología de preguntas del sistema de BR. La tipología de preguntas incluye los tipos de preguntas que el sistema será capaz de clasificar para detectar el tipo de respuesta esperada y las palabras claves de la pregunta. La figura 5.16 muestra este metamodelo.

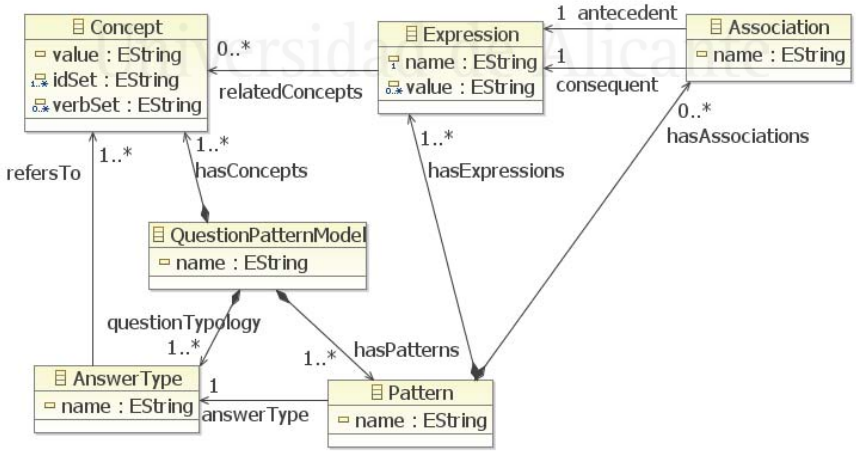

Figura 5.16. Vista General del Metamodelo Patrones de Preguntas.

Un patrón es representado como una metaclase *Pattern* a fin de tener varias expresiones asociadas (metaclases *Expression* y *Association*) que representen a un patrón. Además, un patrón está asociado a un tipo de respuesta (metaclase *AnswerType*), lo que será muy útil para clasificar la respuesta esperada de ciertos tipos de preguntas. La metaclase *Expression* se usa para considerar todo tipo de expresiones. Por ejemplo, etiquetas sintácticas como preposición (en inglés, PP-*preposition*), pronombre determinante o interrogativo (PtDt-*interrogative pronoun or determinant*), núcleo verbal (VBC-*verbal head*), sintagma nominal simple (SNP-*simple noun phrase*), sintagma preposicional simple (SPP-*simple preposition phrase*), y sus valores (p.e., una expresión PtDt puede tener "cuál" como valor). También las expresiones pueden tener algunos conceptos relacionados (p.e. un SNP puede tener hipónimos de ciertos conceptos dentro de su expresión). La metaclase *Association* relaciona las expresiones en función de conocer el orden secuencial, a través de un antecedente y consecuente. Si estas asociaciones son sintácticas, entonces un ejemplo podría ser una asociación con nombre "PtDt-VBC" donde el antecedente es una expresión "PtDt" y el consecuente "VBC". Por otro lado, la metaclase *AnswerType* se refiere a uno o más conceptos para determinar el tipo de la respuesta esperada, un ejemplo puede ser "objeto", el cual se puede referir a conceptos como "instrumento_musical" y "edificio". La metaclase *Concept* contiene uno o más IDs almacenados en el atributo que identifica los diferentes SOC donde este concepto supuestamente aparece y también varios verbos que frecuentemente son asociados al concepto. Por ejemplo, el *valor* del concepto "instrumento_musical" es "instrumento_musical", mientras el conjunto de *IDs* es "09552147_wn, 11500145_wn, 12255091_wn" (aclaremos que "_wn" indica que esos IDs son *synsets* de WordNet) y "tocar" es el verbo relacionado.

Además también se ha desarrollado el metamodelo de **Patrones de Respuestas** (en inglés, *Answer Pattern*) contiene los elementos adecuados para crear una variedad de estos modelos (ver la Figura 5.17).

Un modelo de patrones de respuestas, cuyo núcleo está formado por la metaclase *AnswerPatternModel*, puede contener uno

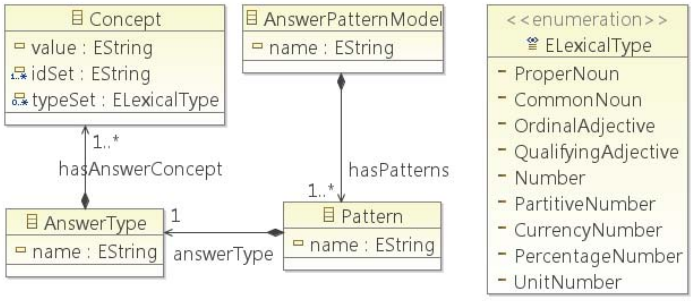

Figura 5.17. Vista General del Metamodelo de Patrones de Respuestas.

o muchos patrones de respuestas representandos en la metaclase *Pattern*. Cada uno de estos patrones contiene un tipo de respuesta representado por la metaclase *AnswerType*, que a su vez está relacionada con uno o muchos conceptos del dominio restringido representado previamente en un modelo. Estos conceptos están representados por medio de la metaclase *Concept*. Los meta-atributos que forman parte de esta metaclase son: *value* para especificar el concepto que se está representando en el modelo, *idSet* que contiene un conjunto de identificadores de ese concepto en diferentes SOC y *typeSet* como el conjunto de tipos léxicos válidos para ese concepto. Finalmente, esos tipos léxicos están definidos a través de la enumeración *ELexicalType*.

Estos metamodelos se usan en las transformaciones T4 y T5 para obtener los modelos de patrones de preguntas y respuestas del SBR-DA que se desee considerar. Estas transformaciones dependen del tipo de implementación del sistema, por eso nuestra aproximación puede manejar todo tipo de patrones actualizando sólo la transformación T4 y T5.

La Figura 5.18 muestra un ejemplo de patrón, de nuestro SBR-DA anteriormente desarrollado (AliQAn (Roger *et al.*, 2008)), llamado "patronEO5". Este patrón tiene los siguientes elementos de la clase *Expression*: ptdtEO5 (con valor "*qué*") y snsEO5 (relacionado con el elemento de la clase *Concept* y todos sus hipónimos, y con varios valores de dominio abierto como "*edificio*" o "*instrumento_musical*", aunque varios de estos elementos no están repre-

sentados en la figura en aras de la claridad). También contiene un elemento de la clase *Association* que sirve para relacionar ambas expresiones, llamada "ptdtEO5-snsEO5". Finalmente, este patrón tiene como tipo de respuesta a "entidad_objeto", lo que hace al SBR-DA capaz de identificar preguntas del tipo: "*Qué instrumento musical tocaba Beethoven?*".

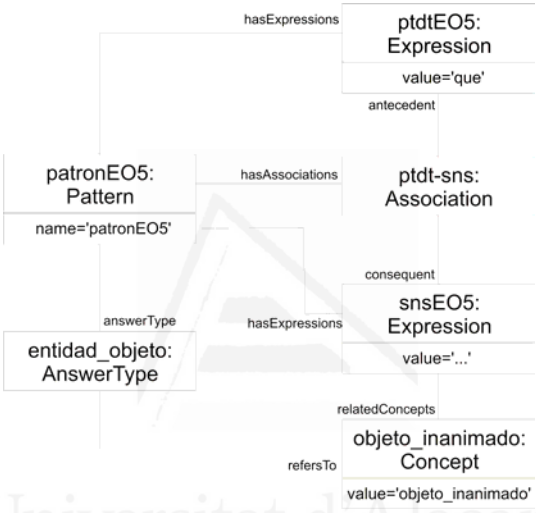

Figura 5.18. Ejemplo de modelo obtenido para un patrón de pregunta de un SBR-DA.

De manera análoga se debe extraer los patrones que permiten la extracción de las respuestas. En este ejemplo, se comenta un patrón de respuesta cuyo modelo se puede observar en la Figura 5.19. En este modelo, un elemento de la clase *Pattern* llamado "patronRespEO" tiene un tipo de respuesta "entidad_objeto", tal y como se aprecia con el elemento asociado de la clase *AnswerType*. Este tipo de respuesta está asociado a varios conceptos (concepto "objeto_inanimado" y sus hipónimos) mediante elementos de la clase *Concept*. Este tipo de patrón permite encontrar la respuesta a la pregunta anteriormente formulada en el siguiente texto "[...] Beethoven, último gran representante del clasicismo vienés, tocaba el piano desde joven [...]".

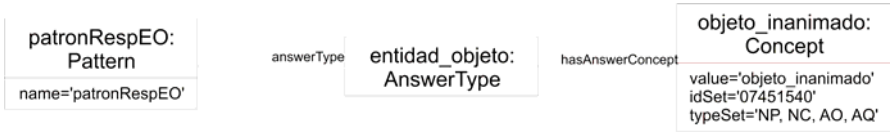

Figura 5.19. Ejemplo de modelo obtenido para un patrón de respuesta de un SBR-DA.

5.2.5. Adaptación de modelos de patrones de pregunta a un dominio restringido.

Las transformación T6 (ver Figura 5.5) es el núcleo de nuestra aproximación para la adaptación de patrones de pregunta a un dominio restringido. Esta transformación toma como entrada el modelo de dominio restringido que representa la taxonomía TRE (tal y como se indicaba en la Sección 5.2.3) y el modelo de patrones de preguntas existente en el sistema de BR. La salida de esta transformación será el modelo de los nuevos patrones de pregunta afinados específicamente para el dominio restringido.

A grandes rasgos, la transformación T6 crea nuevos patrones de preguntas a partir de los patrones existentes siempre que los conceptos a los que se refiere el tipo de respuesta del patrón existente (*conceptos existentes*) estén relacionados con algún concepto del modelo de dominio restringido (*conceptos nuevos*) representado por el modelo de taxonomía de TRE. Esta relación se debe determinar a priori por medio de alguna condición que dependerá del dominio restringido al cuál se desee adaptar los patrones. Cabe destacar que se asume que todos los conceptos tope del SOC genérico utilizado tienen patrones definidos en el sistema de BR-DA baseline. Esta asunción está justificada por el hecho de que la mayoría de las aproximaciones actuales de SBR-DA, en competiciones como el CLEF o TREC, basan sus taxonomías de TRE en los conceptos tope de WordNet o EuroWordNet; por tanto tienen patrones definidos para dichos conceptos. Nosotros proponemos establecer las siguientes condiciones sobre los conceptos, para crear un patrón nuevo derivado de uno existente, a partir de la comparación del concepto existente y el concepto nuevo deben cumplir que: (i) son iguales, (ii) tienen un hiperónimo común, es

decir, son hermanos en la estructura jerárquica de la taxonomía y (iii) mantienen una relación de hipónimo-hiperónimo, es decir, son padre e hijo en la jerarquía. Además el concepto del dominio restringido debe tener más de dos hipónimos. Vale destacar que estas condiciones pueden ser fácilmente modificadas de acuerdo al contexto de aplicación de nuestro método; sin embargo éstas serán las condiciones que usaremos para el dominio que nos ocupa en esta tesis doctoral (dominio agrícola) y los resultados de la misma se pueden consultar en el capítulo 7.

Con el fin de llevar a cabo la adaptación de patrones de preguntas a un dominio restringido concreto, la transformación T6 se ha diseñado mediante un conjunto de reglas QVT: *QuestionPatternModel*, *QuestionPattern*, *Concept*, *Association*, *ExpressionWithNewConcepts* y *ExpressionWithExistingConcepts*.

La regla *QuestionPatternModel* (ver Figura 5.20) crea un modelo de patrones de pregunta a partir del modelo de patrones de pregunta de entrada.

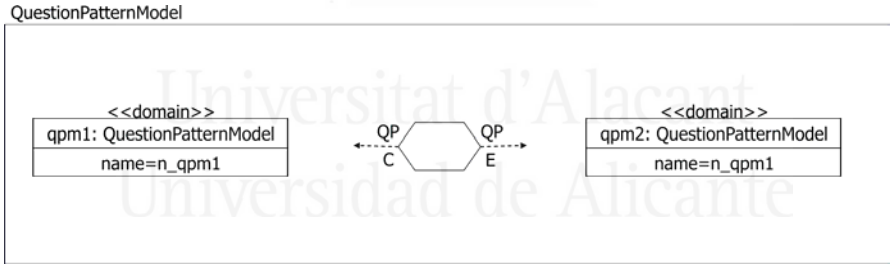

Figura 5.20. Creación de un modelo de patrones de pregunta.

La regla *QuestionPattern* (ver Figura 5.21) tiene como precondición la ejecución de la regla QuestionPatternModel para asegurar que se crea el modelo de patrones de pregunta donde se encontrarán todos los nuevos patrones de pregunta adaptados (según la cláusula *when*). La regla QuestionPattern tiene como entrada (i) un conjunto de elementos del modelo de patrón de pregunta existente: qpm1 que es el elemento de la clase QuestionPatternModel donde se encuentra el patrón de pregunta p1 (elemento de la clase Pattern) y el tipo de respuesta a1 (elemento

de la clase AnswerType); y (ii) un concepto del modelo de dominio restringido (elemento c1 de la clase Concept) representado por la taxonomía de TRE. A partir de estas entradas, si se cumplen las condiciones indicadas anteriormente y especificadas como precondición mediante una sentencia OCL en la cláusula *when*, se creará un nuevo patrón (elemento p2 de la clase Pattern), con un tipo de respuesta (elemento a2 de la clase AnswerType) cuyo nombre es el valor concepto c1. El tipo de respuesta a2 hace referencia a un nuevo concepto creado en el nuevo modelo de patrones de preguntas (elemento c2 de la clase Concept) que toma valor del concepto c1, así como sus identificadores de los SOC de donde procede. Las condiciones para la creación del nuevo patrón de pregunta adaptado se sirven de un par de funciones auxiliares que han sido definidas en OCL (ver Figura 5.22): *getHypernyms* y *getSiblings*. Estas funciones recorren los elementos de un modelo de dominio restringido a partir de un elemento de la clase *Concept* con el fin de obtener, respectivamente, el conjunto formado por todos los conceptos que sean sus hiperónimos y el conjunto formado por todos los conceptos que sean hipónimos de sus hiperónimos. Una vez se cumpla la regla QuestionPattern, se deben ejecutar las reglas que aparecen en la cláusula *where* con el fin de completar el patrón de pregunta adaptado con los elementos necesarios según el metamodelo.

La regla de transformación *Concept* se muestra en la Figura 5.23. Esta regla crea los nuevos conceptos que se usarán en el nuevo patrón de respuesta adaptado. Estos conceptos corresponden a los hipónimos del concepto del modelo de dominio restringido emparejado anteriormente por la regla QuestionPattern. Por tanto, la regla Concept crea nuevos conceptos (elemento c2 de la clase Concept), asociados al nuevo tipo de respuesta (elemento a2 de la clase AnswerType) a partir del elemento p1 de la clase Pattern (previamente emparejado por la regla QuestionPattern) y del elemento c1 de la clase Concept, que es hipónimo del concepto anteriormente emparejado por la regla QuestionPattern.

La regla de transformación *Association* (ver Figura 5.24) empareja el patrón p1 (elemento de la clase Pattern) junto con una

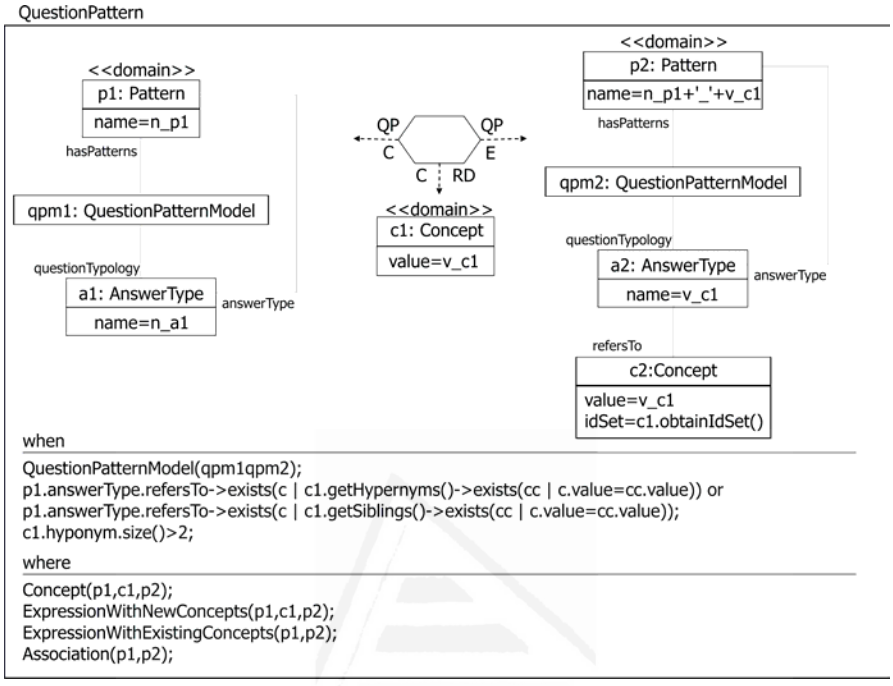

Figura 5.21. Creación de un patrón de pregunta adaptado a partir de uno existente.

```
helper context RD!Concept def : getHypernyms() : Set(RD!Concept) =
    self.hypernym;

helper context RD!Concept def : getSiblings() : Set(RD!Concept) =
    if self.hypernym.size()<>0 then
        self.hypernym->collect(h|h.hyponym)->flatten()->asSet()
    else
        Set{}
    endif;
```

Figura 5.22. Funciones auxiliares en OCL para la adaptación de patrones de pregunta.

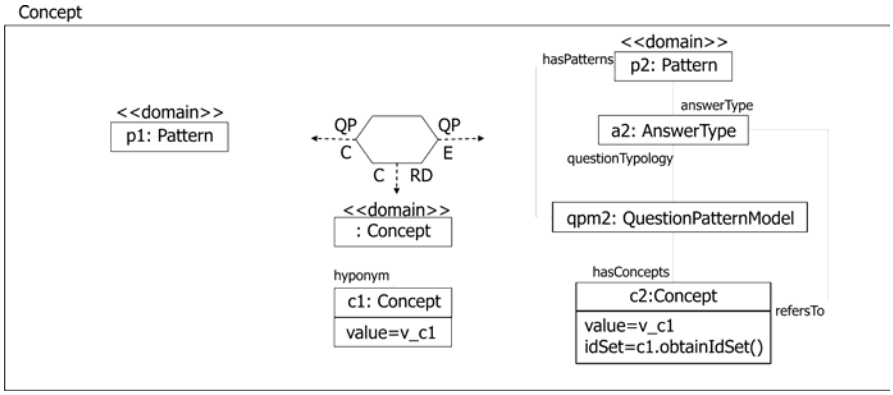

Figura 5.23. Creación de los nuevos conceptos pertenecientes al modelo de patrón de pregunta adaptado.

asociación a1 (elemento de la clase Association) con el fin de crear en el modelo de salida la asociación correspondiente (a2).

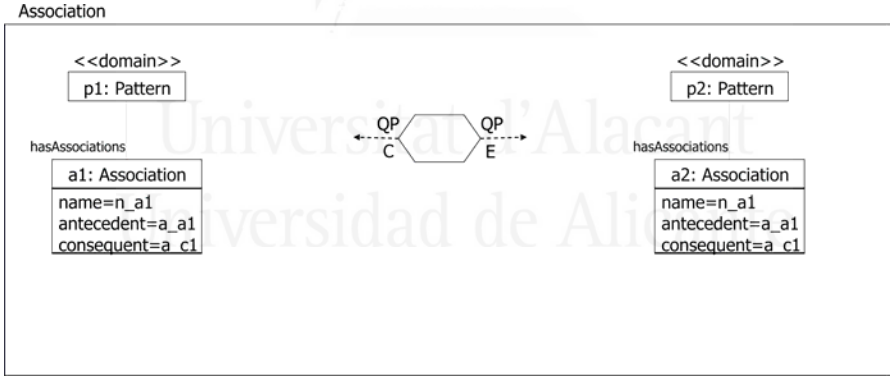

Figura 5.24. Creación de las nuevas asociaciones pertenecientes al patrón de pregunta adaptado.

A la hora de generar las expresiones (elementos de la clase Expression) que forman parte de un patrón se debe tener en cuenta si cada una de las expresiones del patrón del modelo de patrones de pregunta de entrada está relacionada con los conceptos del modelo de dominio restringido. En este caso la relación se debe corresponder con las condiciones que deben cumplir los concep-

tos para crear un patrón nuevo derivado de uno existente (tal y como se explicó anteriormente). De cumplirse esta condición, se ejecuta la regla de transformación *ExpressionWithNewConcepts* (ver Figura 5.25), donde la condición se comprueba en la cláusula *when*. Esta regla comprueba que existe un elemento e1 de la clase Expression en el modelo de patrones de pregunta de entrada y un concepto c1 de la clase Concept en el modelo de dominio restringido de entrada, creando en el modelo de patrones de pregunta de salida la expresión e2 (de la clase Expression) correspondiente. Cabe destacar que el nuevo concepto c2 generado (elemento de la clase Concept) se ha creado previamente con la regla de transformación Concept (ver cláusula *when*).

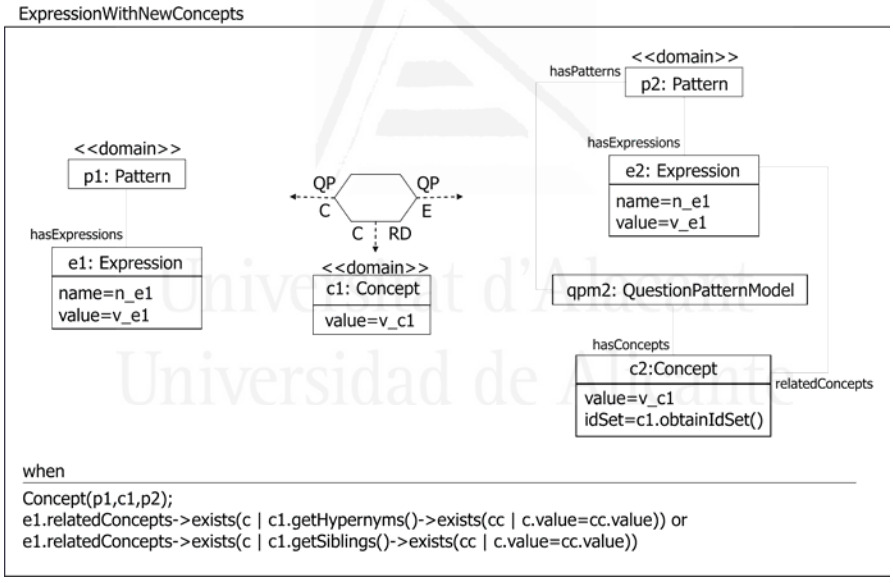

Figura 5.25. Creación de las nuevas expresiones pertenecientes al patrón de pregunta adaptado.

Por otro lado, si la expresión del patrón del modelo de patrones de pregunta de entrada no está relacionada con los conceptos del modelo de dominio restringido según las condiciones previamente descritas, las nuevas expresiones se crearán a partir de la regla de transformación *ExpressionsWithExistingConcepts*. Esta regla

empareja un conjunto de elementos formados por un patrón (p1 de la clase Pattern), una expresión (e1 de la clase Expression) y un concepto (c1 de la clase Concept), con el fin de crear sus correspondientes elementos en el modelo de patrones de pregunta de salida (ver Figura 5.26).

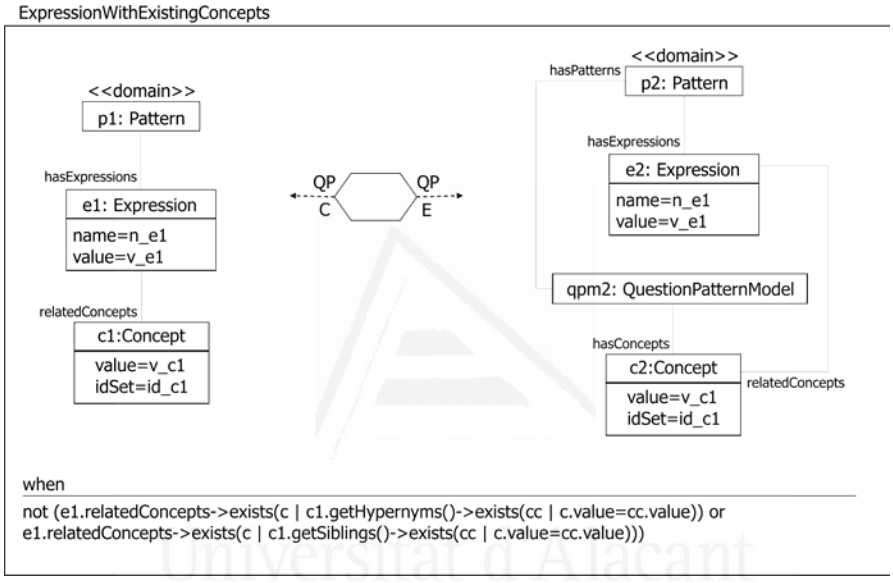

Figura 5.26. Creación de las nuevas expresiones pertenecientes al patrón de pregunta adaptado.

Con el fin de generar los modelos de patrones de pregunta más refinados, la transformación T6 se debe aplicar de nuevo sobre los nuevos patrones hasta que no se obtengan nuevos modelos de patrones de pregunta.

Finalmente cabe destacar, que los conceptos del modelo de dominio restringido de la taxonomía de TRE que no estén relacionados con un nuevo patrón de pregunta se incorporan a un patrón genérico (patrón bolsa) con el fin de que pueda ser refinado manualmente por el desarrollador.

Ejemplo ilustrativo de la adaptación de un modelo de patrones de preguntas a un dominio restringido En nuestro

ejemplo, el concepto "glicosidos" proveniente del modelo de dominio restringido que representa la taxonomía de TRE tiene un concepto ("objeto_inanimado") en el camino de hiperonimia que se relaciona con el concepto existente en el patrón del SBR-DA llamado "patronEO5" (ver Figura 5.18). Por tanto según la transformación T6, se crea el nuevo patrón de la Figura 5.27. Este nuevo patrón se crea con la regla de transformación *QuestionPattern* por lo que toma de nombre "patronE05_glicosidos", y se crea un nuevo elemento de la clase "AnswerType" que tendrá el mismo nombre del concepto extraído del modelo de dominio restringido ("glicosidos") y que hará referencia a un nuevo elemento de la clase *Concept* llamado "glicosidos". A partir de esta regla de transformación, ejecutan las reglas que crean los nuevos elementos de la clase *Association* y *Expression*, que son *Association*, *ExpressionWithNewConcepts* y *ExpressionWithExistingConcepts*. En este caso se crean las expresiones "ptdtEO5" y "snsEO5", así como una asociación entre ambas ("ptdt-sns"). La nueva expresión "snsEO5" proviene de una expresión cuyos conceptos relacionados eran el concepto "objeto_inanimado" y sus hipónimos, por lo que ahora se relacionará con el nuevo concepto "glicosidos" y con sus hipónimos ("nucleosidos", "glucosidos", "nucleotidos", "saponinas", "glicosidos", "glicosidos_cianogenicos") creados mediante la regla de transformación *Concept*. El nuevo patrón se muestra en la Figura 5.27.

Cabe destacar que el nuevo modelo de patrón obtenido mediante la transformación T6, hará posible la contestación de un nuevo tipo de pregunta de dominio restringido, por ejemplo, la anteriormente mencionada *"¿Qué saponinas tienen un efecto defaunante en el rumen?"*, ya que "saponinas" es un hipónimo de "glicosidos".

5.2.6. Adaptación de modelos de patrones de repuesta a un dominio restringido.

Las transformación T7 (ver figura 5.5), al igual que la T6, resulta clave dentro de nuestra aproximación, ya que realiza la adaptación de patrones de respuesta a un dominio restringido.

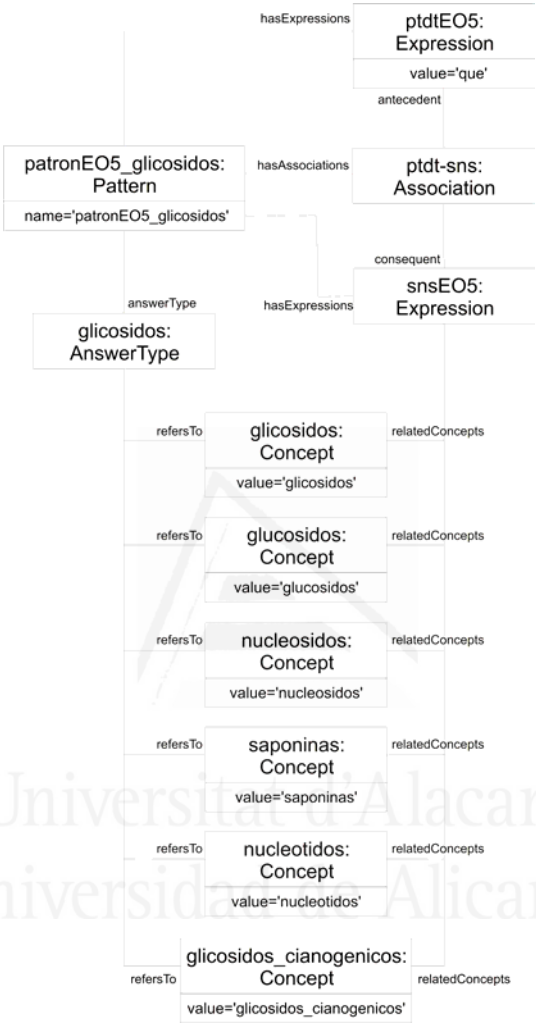

Figura 5.27. Ejemplo de modelo de patrón de pregunta obtenido para un patrón de pregunta de un SBR-DA.

Esta transformación toma como entrada el modelo de dominio restringido que representa la taxonomía TRE (tal y como se indicaba en la Sección 5.2.3) y el modelo de patrones de respuesta existente en el sistema de BR. La salida de esta transformación será el modelo de los nuevos patrones de respuesta adaptados al nuevo dominio restringido.

La transformación T7 crea nuevos patrones de respuestas a partir de los patrones existentes siguiendo los mismos criterios que la transformación T6. Concretamente, la transformación T7 se compone del siguiente conjunto de reglas QVT: *AnswerPatternModel*, *AnswerPattern* y *Concept*.

La regla *AnswerPatternModel* (ver Figura 5.28) crea un modelo de patrones de respuesta a partir del modelo de patrones de respuesta de entrada.

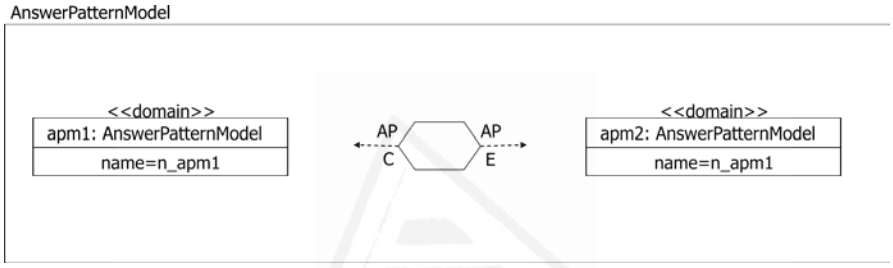

Figura 5.28. Creación de un modelo de patrones de respuesta.

La regla *AnswerPattern* (ver Figura 5.29) tiene como precondición, en la cláusula *when*, la ejecución de la regla AnswerPatternModel para asegurar que se crea el modelo de patrones de respuesta donde se encontrarán todos los nuevos patrones de respuesta adaptados. La regla AnswerPattern tiene como entrada dos patrones. El primero es un conjunto de elementos del modelo de patrón de respuesta existente: apm1 que es el elemento de la clase AnswerPatternModel donde se encuentra el patrón de respuesta p1 (elemento de la clase Pattern) y el tipo de respuesta a1 (elemento de la clase AnswerType). El segundo es un concepto del modelo de dominio restringido (elemento c1 de la clase Concept) representado por la taxonomía de TRE. A partir del emparejamiento de estos dos patrones, siempre y cuando se cumplan las condiciones especificadas como precondición mediante una sentencia OCL en la cláusula *when* (estas condiciones son equivalentes a las especificadas anteriormente para la regla *QuestionPattern* en la sección 5.2.5), se creará un nuevo patrón (elemento p2 de la clase Pattern), con un tipo de respuesta (elemento a2 de la clase

AnswerType) cuyo nombre es el valor concepto c1. El tipo de respuesta a2 hace referencia a un nuevo concepto creado en el nuevo modelo de patrones de respuesta (elemento c2 de la clase Concept) que toma valor del concepto c1, así como sus identificadores de los SOC de donde procede. Las condiciones para la creación del nuevo patrón de pregunta adaptado se sirven de un par de funciones auxiliares que han sido definidas en OCL (ver Figura 5.22): *getHypernyms* y *getSiblings*. Estas funciones fueron explicadas anteriormente. Una vez se cumpla la regla QuestionPattern, se debe ejecutar la regla *Concept* que aparecen en la cláusula *where* con el fin de completar el patrón de respuesta adaptado mediante la incorporación de los conceptos necesarios según los modelos de entrada.

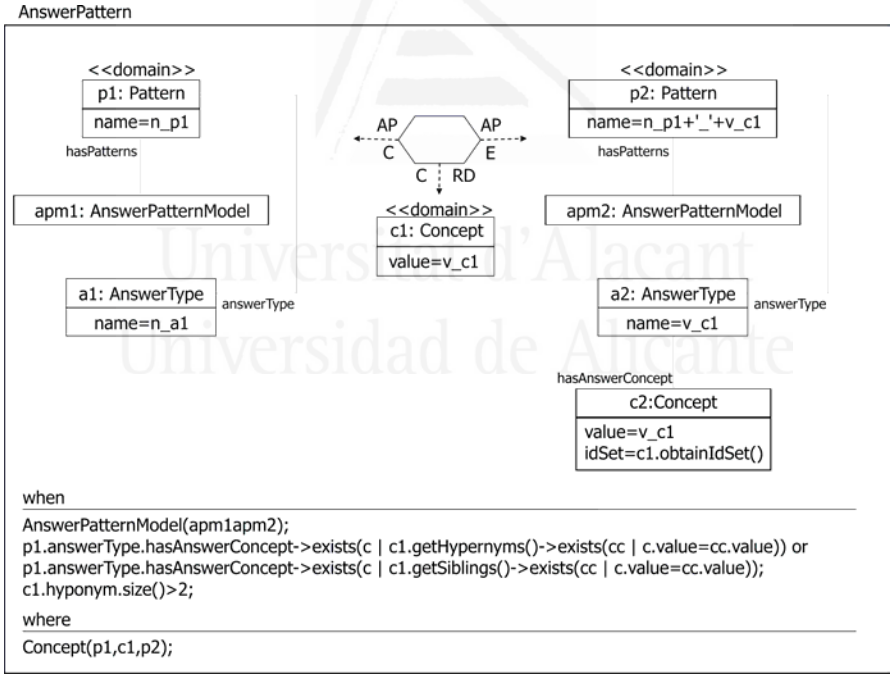

Figura 5.29. Creación de un patrón de respuesta adaptado a partir de uno existente.

La regla de transformación *Concept* para patrones de respuesta se muestra en la Figura 5.30. Esta regla crea los nuevos conceptos que se usarán en el nuevo patrón de respuesta adaptado. Estos conceptos corresponden a los hipónimos del concepto del modelo de dominio restringido emparejado anteriormente por la regla AnswerPattern. Por tanto, la regla Concept crea nuevos conceptos (elemento c2 de la clase Concept), asociados al nuevo tipo de respuesta (elemento a2 de la clase AnswerType) a partir del elemento p1 de la clase Pattern (previamente emparejado por la regla AnswerPattern) y del elemento c1 de la clase Concept, que es hipónimo del concepto anteriormente emparejado por la regla AnswerPattern.

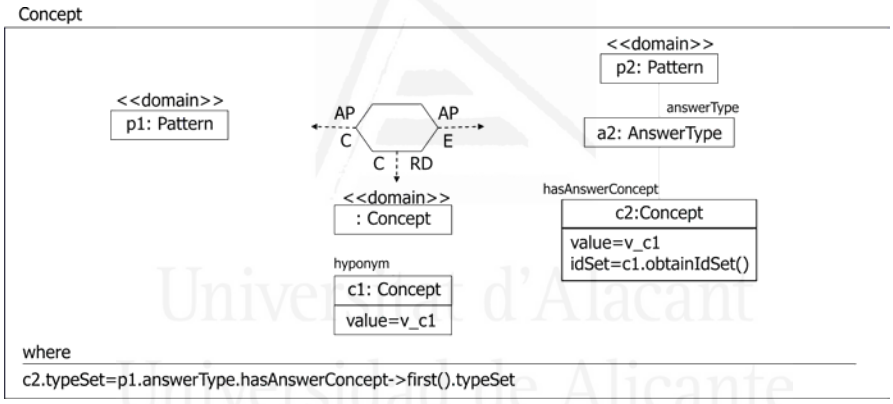

Figura 5.30. Creación de los nuevos conceptos pertenecientes al modelo de patrón de respuesta adaptado.

Al igual que para los patrones de pregunta, con el fin de generar los modelos de patrones de respuesta más refinados, la transformación T7 se debe aplicar de nuevo sobre los nuevos patrones hasta que no se obtengan nuevos modelos de patrones de respuesta.

Finalmente cabe destacar, que los conceptos del modelo de dominio restringido de la taxonomía de TRE que no estén relacionados con un nuevo patrón de respuesta se incorporan a un patrón genérico (patrón bolsa) con el fin de que pueda ser refinado manualmente por el desarrollador.

Ejemplo ilustrativo de la adaptación de un modelo de patrones de respuestas a un dominio restringido El nuevo patrón de respuesta creado según el ejemplo desarrollado en este capítulo proviene de considerar que el concepto "glicosidos" del modelo de dominio restringido que representa la taxonomía de TRE forma parte del camino de hiponimia de un concepto ("objeto_inanimado") que está asociado a un patrón de respuesta existente en el SBR-DA llamado "patronEO" (ver Figura 5.19). Por tanto según la transformación T7, se crea el nuevo patrón de la Figura 5.31. Este nuevo patrón se crea con la regla de transformación AnswerPattern por lo que toma de nombre "patronRespE0_glicosidos", y se crea un nuevo elemento de la clase "AnswerType" que tendrá el mismo nombre del concepto extraído del modelo de dominio restringido ("glicosidos") y que hará referencia a un nuevo concepto llamado "glicosidos" y a todos sus hipónimos, creados como elementos de la clase *Concept* a partir de la regla de transformación Concept.

Cabe destacar que el nuevo modelo de patrón obtenido mediante la transformación T7, hará posible que se encuentra la respuesta a la pregunta formulada anteriormente en una frase como la que sigue: *"[...] por ejemplo , las saponinas tienen un efecto defaunante en el rumen [...]"*.

5.2.7. Generando el código de los nuevos patrones de preguntas y respuestas.

Por último, las transformaciones T8 y T9 (ver Figura 5.5) automáticamente despliegan el código correspondiente a cada nuevo patrón de preguntas y de respuestas, respectivamente para adaptar los patrones de un SBR-DA. Para llevar a cabo este paso, estas transformaciones están basadas en la idea de plantillas configurables para capturar reglas de traducción de los modelos de patrones de preguntas y respuestas en el código correspondiente para diferentes sistemas de BR. En este caso se muestran los patrones de pregunta y respuesta generados según nuestro ejemplo en AliQAn.

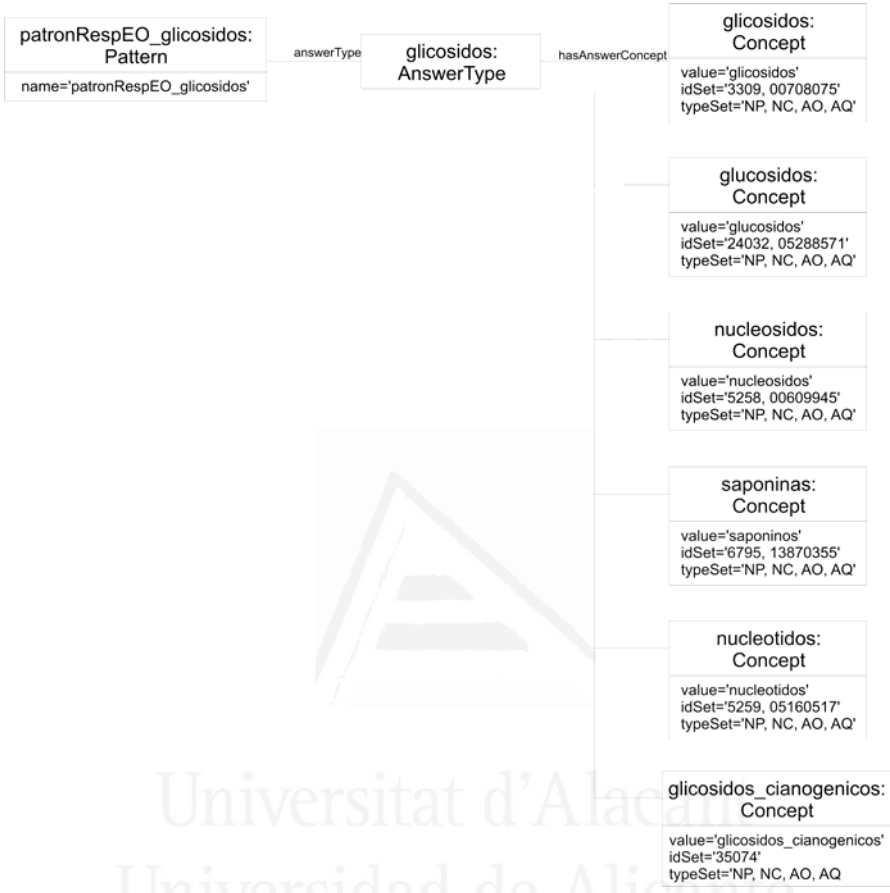

Figura 5.31. Ejemplo de modelo de patrón de respuesta obtenido para un patrón de pregunta de un SBR-DA.

```
TGrupo generarGlicosidos()
{
    list<TBS> listaBloques;
    TGrupo glicosidos("glicosidos");
    string LEltosGlicosidos = "H 5258 H 00609945 H 24032
     H 05288571 H 5259 H 05160417 H 6795 H 13870355 H 3309
     H 00708075 H 35074";

    TPatron patronEO5_glicosidos;
    patronEO5_glicosidos.insertarPtdt("que");
    listaBloques.push_back(insertarSNS(LEltosGlicosidos,true));
    patronEO5_glicosidos.insertarBS(listaBloques);
    listaBloques.clear();
    glicosidos.insertarPatron(patronEO5_glicosidos);
}
```

```
map<string,TPatronResp> generarPatronesResp()
{
    map<string,TPatronResp> patronesResp;

    TPatronResp patronRespGlicosidos;
    patronRespGlicosidos.insertarTipo("NP NC AO AQ");
    patronRespGlicosidos.insertarTipoPregunta("glicosidos");
    patronRespGlicosidos.insertarBP(insertarSNS2("H 3309 H 00708075
     H 6795 H 13870355 H 5258 H 00609945 H 5259 H 05160417 H 24032
     H 05288571 H 35074"));
    patronesResp.insert(make_pair("glicosidos",patronRespGlicosidos));
}
```

Los detalles de las transformaciones T8 y T9 se comentarán en el capítulo de implementación 6.

5.3. Conclusiones

En este capítulo hemos presentado nuestra aproximación dirigida por modelos para abordar la compleja tarea de adaptar automáticamente un SBR-DA a un dominio restringido de manera sistemática y bien estructurada. Nuestra propuesta se basa en la necesidad de disponer de una adaptación de SBR-DA a dominios restringidos (i) con un grado de automatización lo más elevado posible, (ii) integrando cualquier tipo de SOC según el dominio a tratar e (iii) independiente de corpus de preguntas de dominio restringido artificiales, considerando pues solamente la colección de documentos disponible.

Para llevar a cabo esta propuesta con éxito, cumpliendo con estas premisas, se ha utilizado una técnica ampliamente utilizada en ingeniería del software: el desarrollo dirigido por modelos (MDD). Nos basamos en la creación de tres metamodelos que nos permiten definir de manera correcta y formal, en sus respectivos modelos, aquellos elementos necesarios del dominio restringido, de los patrones de preguntas y de los patrones de respuestas. Esto se consigue mediante la abstracción de detalles superfluos y no necesarios para la tarea de adaptación de SBR-DA, objetivo de esta tesis. Nuestra propuesta se fundamenta en una serie de transformaciones definidas sobre estos metamodelos mediante el

lenguaje estándar QVT, lo que nos permite automatizar la creación de una taxonomía de TRE y la adaptación de los patrones de preguntas y respuestas del SBR-DA al nuevo dominio restringido de aplicación.

Los principales beneficios alcanzados en cuanto al método de adaptación de los patrones son los siguientes:

1. **Productividad:** el sistema de BR puede ser fácilmente adaptado a partir de los patrones existentes, ya que los modelos están basados en metamodelos formales y las transformaciones se han diseñado mediante QVT para ser ejecutadas automáticamente. Por lo tanto, la productividad es mejorada y decrece el costo y el tiempo de desarrollo necesario para la adaptación del SBR-DA a un nuevo dominio.
2. **Adaptabilidad:** si surgen nuevas tecnologías o recursos de conocimiento, no es necesario cambiar el sistema de BR completo, si no que sólo se deben adaptar las transformaciones para obtener los modelos adecuados. De esta manera los sistemas de BR-DA son preservados porque son independientes del dominio y los desarrolladores sólo tienen que preocuparse por las transformaciones en el módulo de adaptación.
3. **Portabilidad:** los mismos modelos de patrones pueden ser automáticamente transformados en diferentes tipos de código dependiendo del sistema de BR objetivo. De esta manera se podría por ejemplo utilizar los patrones de un sistema de BR adaptados a otro sistema totalmente diferente. Además, se pueden utilizar diferentes tipos de SOC de manera sencilla al haber desarrollado un metamodelo que permite abstraer aquellos elementos de los SOC útiles para los sistemas de BR.
4. **Reusabilidad:** las mejores practicas en el diseño de sistemas de BR pueden ser incluidas en las transformaciones para garantizar una elevada calidad final en el sistema adaptado.
5. **Integración e Interoperabilidad:** los sistemas de BR usan recursos de conocimiento heterogéneos, por lo tanto su desarrollo necesita poder manejarlos de manera conjunta y homogénea. Nuestro método permite la adaptación de los patrones a través de la integración de diferentes recursos de conoci-

miento con facilidad por medio del desarrollo de un metamodelo común.

Finalmente cabe destacar que la generación de taxonomías de TRE para dominios restringidos evita la adición de conceptos que carecen de interés para ese dominio; obteniéndose así una taxonomía de TRE ajustada al dominio. Además, se realiza de forma automática y no es necesario disponer de grandes colecciones de preguntas para su diseño. Por otra parte, la cobertura de la taxonomía de TRE que obtenemos es real, ya que su diseño es a partir de los recursos de conocimiento (SOC y corpus textual) disponibles en el dominio. Por tanto, el uso de la taxonomía generada usando nuestro método garantiza una buena precisión del sistema de BR que la utilice.

Capítulo 6

MARAQA: una herramienta para la adaptación dirigida por modelos de sistemas de búsqueda de respuestas a dominios restringidos

MARAQA (***M**odel-driven **A**daptation for **R**estricted-dom**A**in **Q**uestion **A**nswering*) es la herramienta que implementa la propuesta de adaptación de SBR-DA a dominios restringidos descrita en esta tesis. Esta implementación se ha realizado mediante la plataforma Eclipse[1], ya que dicha plataforma provee muchas facilidades para la definición de propuestas dirigidas por modelos en un entorno de desarrollo integrado. Una vez presentada la plataforma Eclipse, en este capítulo se describe como se ha realizado la implementación de los metamodelos y de las transformaciones que dan soporte a la propuesta descrita en esta tesis.

6.1. Visión general de Eclipse

Eclipse es un Entorno Integrado de Desarrollo (IDE) abierto y extensible con el que se puede trabajar prácticamente con cualquier lenguaje. Está desarrollado íntegramente con Java, pero es adaptable a cualquier tipo de lenguaje, por ejemplo C++, Cobol, C#, XML, etc. En principio fue un proyecto de IBM y luego pasó a la comunidad OpenSource. Eclipse es un sistema modular por lo que es una plataforma extensible y configurable para una gran cantidad de entornos y lenguajes de programación. El mecanismo de extensibilidad de Eclipse permite añadir nuevas funcionalidades por medio de *plug-ins* que interactuan mediante

[1] `http://www.eclipse.org`

interfaces o puntos de extensión, de este modo las nuevas aportaciones se integran fácilmente, minimizando posibles conflictos. Eclipse constituye una plataforma ideal para el desarrollo del software, compuesta de un entorno, herramientas y ejecutables que permiten diseñar, desarrollar, implementar y gestionar el software a lo largo de su ciclo de desarrollo.

En la Fig. 6.1 se presenta la arquitectura de la plataforma Eclipse. La característica principal de esta arquitectura es que permite la incorporación de nuevos *plug-ins* que especializan las características de la plataforma de desarrollo (ver la parte etiquetada como *Your tool here* en la Fig. 6.1). Un *plug-in* es la menor unidad de funcionalidad en la plataforma Eclipse, el cual puede ser desarrollado de manera independiente. Usualmente una herramienta pequeña se desarrolla mediante *plug-in* simple, mientras que una herramienta compleja divide su funcionalidad en varios *plug-ins*.

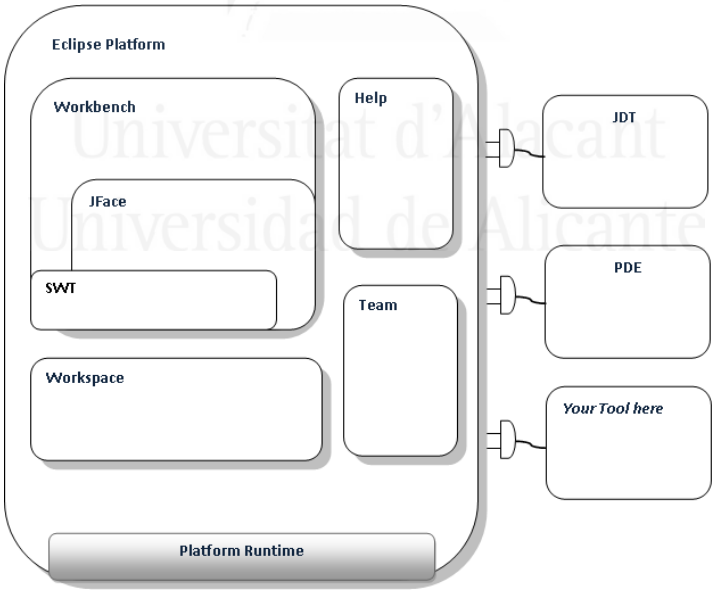

Figura 6.1. Arquitectura de la plataforma Eclipse.

La plataforma Eclipse está formada por el entorno de trabajo (*Workbench*), el espacio de trabajo (*Workspace*), la ayuda (*Help*) y los componentes de grupo *Team* cuya meta es proporcionar integración del repositorio de herramientas en Eclipse. El entorno de trabajo (*Workbench*) proporciona una interfaz de usuario para Eclipse, su propósito es facilitar la integración de las herramientas. Estas herramientas contribuyen a la definición mediante puntos de extensión (*Extension points*) definidos por el entorno de trabajo, que además es el responsable de la presentación y coordinación de la interfaz de usuario. El entorno de trabajo comprende: SWT (*Standard Widget Toolkit*) y JFace. SWT es un conjunto de herramientas de código abierto diseñadas para facilitar que Java proporcione facilidades de portabilidad a los sistemas operativos en los cuales está diseñada la interfaz de usuario, por ello se puede usar independientemente de la plataforma Eclipse. JFace es un marco de aplicación de Java basado en SWT. Su meta es proporcionar un conjunto de componentes reutilizables para facilitar la escritura de aplicaciones de interfaz gráfica de usuario (*Graphical User Interface*, GUI).

Usando toda la potencialidad que ofrece la arquitectura de Eclipse, se ha llevado a cabo un proyecto llamado Eclipse Modeling Framework (EMF)[2] que permite dotar a Eclipse de funcionalidades para la definición de propuestas de desarrollo de software dirigidas por modelos.

EMF es una plataforma para el diseño e implementación de herramientas basadas en modelos, sirviendo por tanto para implementar herramientas que den soporte propuestas de desarrollo de software dirigido por modelos.

6.2. Arquitectura general de MARAQA

La herramienta MARAQA se divide en cuatro módulos: metamodelos (módulo 1), transformaciones para la obtención de modelos (módulo 2), transformaciones entre modelos (módulo 3) y transformaciones para la generación de código (módulo 4). Esta

[2] `http://www.eclipse.org/modeling/emf`

arquitectura se muestra en la Fig.6.2, donde se puede observar en qué módulos se encuentran cada una de las partes de la propuesta y se muestra la interacción entre los módulos.

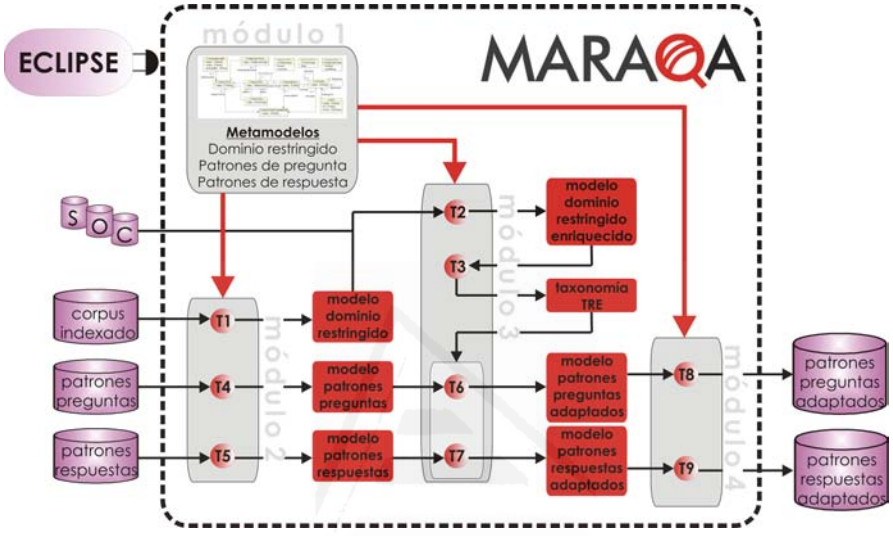

Figura 6.2. Arquitectura general de MARAQA.

En el módulo 1 se encuentra la implementación de los metamodelos desarrollados en la propuesta: metamodelo de dominio restringido, metamodelo de patrones de pregunta y metamodelo de patrones de respuesta. El módulo 2 hace uso del módulo 1 para poder obtener modelos del dominio restringido (a partir del corpus del dominio), de los patrones de pregunta existentes y de los patrones de respuesta existentes. El módulo 3 generaa los modelos de los patrones de pregunta y respuesta adaptados al nuevo dominio, previa obtención de la taxonomía de TRE. Este módulo 3 usa los modelos generados por el módulo 2, así como los metamodelos del módulo 1. Finalmente el módulo 4 toma como entrada los modelos de patrones de preguntas y respuestas generados por el módulo 3 para obtener el código correspondiente. A continuación se definirá cómo se ha implementado el contenido de cada uno de estos módulos, es decir, los metamodelos y las diferentes transformaciones.

6.2.1. Metamodelos

En concreto, EMF provee una manera intuitiva y gráfica de definir las clases (sus atributos y relaciones) de un metamodelo a través de ECore, un metametamodelo alineado con MOF. ECore permite definir metamodelos mediante el uso de las siguientes clases (ver Fig. 6.2.1):

- EClass representa una clase que puede contener o no atributos y referencias a otras clases.
- EAttribute representa un atributo con un nombre y un tipo.
- EReference representa un extremo de una asociación entre dos clases.
- EDataType representa el tipo de atributo, por ejemplo int, float oo java.util.Date.

Además existe otra clase llamada EFactory que contiene métodos para gestionar elementos del un modelo. Esto resulta muy conveniente cuando se diseña una propuesta de desarrollo de software dirigido por modelos, donde los elementos contenidos en éstos deben ser manipulados (cambiados, eliminados o creados) por las transformaciones.

Un modelo en ECore se muestra en un editor en forma de árbol. Se representa un modelo completo como un objeto raíz, a partir del cual cuelgan las diferentes clases. Las clases pueden contener otras clases que cuelgan de ellas. Los atributos se representan como propiedades de las clases.

Los tres metamodelos descritos en esta tesis doctoral (dominio restringiso, patrones de preguntas y patrones de respuestas) se han definido mediante el ECore.

6.2.2. Transformaciones

Las transformaciones definidas en la propuesta se han clasificado en tres tipos con el fin de realizar su adecuada implementación. En primer lugar existen transformaciones que se encargan de generar modelos a partir de recursos ya existentes (transformaciones T1, T4 y T5). También existen transformaciones entre modelos

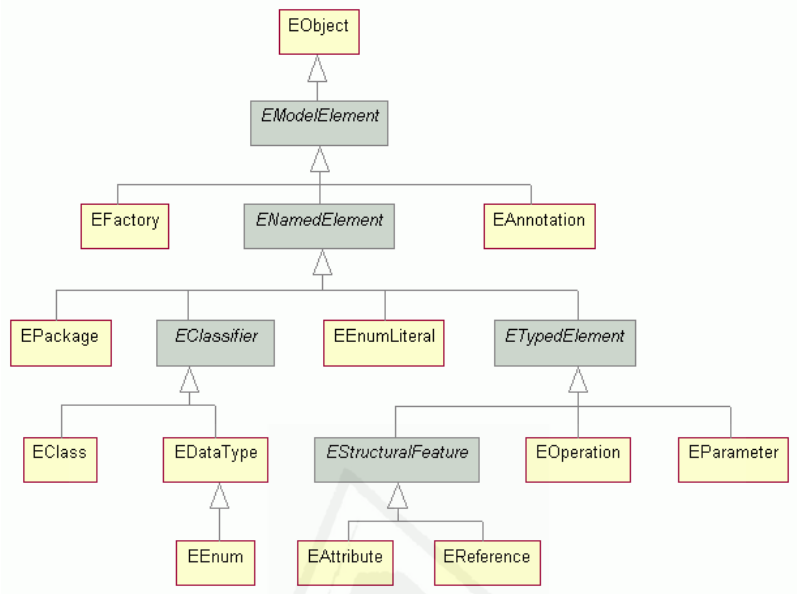

Figura 6.3. Extracto del metamodelo ECore.

(T2,T3,T6 y T7). Finalmente, en la propuesta también se definen transformaciones para generar código (transformaciones T8 y T9).

Transformaciones que generan modelos El primer paso para adaptar un SBR-DA a dominios restringidos es obtener modelos de los patrones existentes (de pregunta y respuesta), así como del dominio al cuál se realiza la adaptación a partir del corpus existente. Esta tarea se ha realizado mediante tres transformaciones (T1, T4 y T5), todas ellas implementadas mediante el uso de Java y EMF, que aporta un API reflectiva para poder crear y modificar modelos basados en Ecore de manera dinámica.

Transformaciones entre modelos Las transformaciones entre modelos se diseñaron en QVT (T2,T3,T6 y T7). Para su implementación se ha optado por el uso del motor de transformaciones de ATL (Atlas Transformation Language)[3]. ATL se integra

[3] `http://www.eclipse.org/atl/`

fácilmente con la plataforma Eclipse permitiendo la definición y ejecución de transformaciones entre modelos basados en ECore.

ATL posee un mecanismo para especificar reglas de transformación que indiquen cómo se generan elementos de un modelo destino a partir de un modelo origen. Las reglas en ATL permiten especificar:

1. Qué elementos del modelo origen debe intentar emparejarse.
2. El número y el tipo de elementos generados en el modelo destino.
3. La manera en la que los elementos generados en el modelo destino deben inicializarse a partir de los elementos emparejados en el modelo origen.

Con este fin, la especificación de una regla en ATL debe seguir la siguiente sintaxis:

```
rule rule_name {
    from
        in_var : in_type [in model_name]? [(
            condition
        )]?
    [using {
        var1 : var_type1 = init_exp1;
        ...
        varn : var_typen = init_expn;
    }]?
    to
        out_var1 : out_type1 [in model_name]? (
            bindings1
        ),
        ...
        out_varn : out_typen [in model_name]? (
            bindingsn
        )
    [do {
        statements
    }]?
}
```

Cada regla se identifica con su nombre (*rule_name*) que debe ser único para cada transformación. Una regla se compone de dos partes obligatorias (la parte *from* y la parte *to*) y dos partes opcionales (la parte *using* y la parte *do*).

La parte *from* corresponde con el patrón origen de la regla. Este patrón contiene la declaración de variable (*in_var*) que indica el

tipo de elementos del modelo origen que serán emparejados por la regla (*in_type*). También puede contener una expresión booleana opcional para indicar que sólo los elementos del tipo indicado del modelo origen que cumplan cierta condición podrán ser emparejados (*condition*).

La parte opcional *using* hace posible declarar variables locales, tanto el nombre (*vari*) como el tipo (*var_typei*). Estas deben inicializarse con expresiones OCL.

La parte *to* corresponde con el patrón destino de la regla, pudiendo contener varios elementos. El primer elemento de este patrón es el elemento por defecto de la regla, por lo que este elemento del modelo destino es el elemento correspondiente al elemento emparejado en el modelo origen por la parte *from* de la regla. Cada elemento del patrón destino corresponde con una declaración de variable con un nombre (*out_vari*) y un tipo (*out_typei*). Un patrón destino se especifica, por tanto, como un conjunto de vínculos que definen la manera de inicializar los atributos y referencias de los elementos generados.

Otra parte opcional es la parte *do*, que permite especificar una secuencia de sentencias imperativas a ejecutar una vez que los elementos generados por la regla se inicialicen. Se puede usar para inicializar algún elemento del modelo que no haya sido inicializado con anterioridad o para realizar alguna modificación.

A continuación se muestra la implementación en ATL de la regla de transformación QVT concept2concept4EAT, definida en el capítulo 5.

```
rule concept2concept4EAT {
    from c1:RD!Concept (c1.hypernym.size()=0 and
                        c1.hasDescendentsWithMoreThan2Hyponyms())
    to c2:RD!Concept (value<-c1.value, hyponym<-c1.hyponym,
                      hypernym<-c1.hypernym)
    do
    {
        for(i in c1.hyponym)
        {
            if(i.hasDescendentsWithMoreThan2Hyponyms())
            {
                c2.hyponym<-thisModule.newConcept4EAT(i);
                thisModule.resolveTemp(c1.getRestrictedDomainModel(),
                                       'rm2').concepts<-c2.hyponym;
            }
        }
```

```
    }
}
```

Transformaciones modelo a código Las transformaciones definidas en mof2text (T8 y T9) se han implementado mediante la plataforma Acceleo[4]. Esta plataforma está integrada en Eclipse por lo que permite la implementación de transformaciones mof2text mediante un conjunto de etiquetas sobre metamodelos Ecore.

Cada transformación definida en Acceleo se llama módulo (*module*). Cada módulo puede contener una o varias plantillas (*template*) que se encargan de generar texto o consultas (*query*) que permiten extraer información del modelo de entrada.

Una transformación en Acceleo debe comenzar por la declaración del módulo:

```
[modul <module_name>('metamodelURI')]
```

Las plantillas representan un conjunto de sentencias que permiten emparejar elementos de un modelo de entrada y generar el texto deseado. Se delimitan por las etiquetas:

```
[template]
```

y

```
\verb[/template]
```


Y contienen un parámetro que indica la clase del metamodelo a partir de la cuál se comienza el emparejamiento en el modelo de entrada.

Dentro de una plantilla de Acceleo pueden existir varias etiquetas que determinan la funcionalidad de la plantilla. A continuación se describen aquellas usadas en este trabajo de tesis doctoral:

La etiqueta *file* se usa para indicar al motor de transformaciones de Acceleo que debe generar todo el contenido de la etiqueta en un fichero. La sintaxis es la siguiente:

```
[file (<uri_expression>, <append_mode>, '<output_encoding>')]
(...)
[/file]
```

[4] http://www.eclipse.org/acceleo/

```
<uri_expression> indica el nombre del fichero de salida;
<append_mode> (opcional) indica si el texto de salida debe añadirse
al fichero o se debe reemplazar su contenido completo;
<output_encoding> (opcional) indica la codificación a usar en el
fichero de salida.
```

La definición de bucles se hace mediante la etiqueta *for* de la siguiente manera:

```
[for (iterator : Type | expression)]
(...)
[/for]
```

También es posible indicar la generación de texto antes o después de cada iteración de un bucle, mediante los parámetros *before*, *separator* o *after*. Un ejemplo sería el siguiente:

```
[for (Sequence{1, 2, 3}) before ('sequence: ') separator (', ')
after (';')] [self/][/for]
```

Este ejemplo generaría el siguiente texto:

```
sequence: 1, 2, 3;
```

Otra etiqueta muy útil es *if*, que permite especificar condiciones:

```
[if (condition)] (...) [/if]
```


Existe además en Acceleo una biblioteca bastante amplia de funciones para el manejo de cadenas de caracteres. Por ejemplo:

```
toUpperFirst()
```

crea una copia de una cadena de caracteres reemplazando el primer carácter por su equivalente en mayúsculas. Por ejemplo el resultado de:

```
'toUpperFirstoperation'.toUpperFirst()
```

sería

```
'ToUpperFirstoperation'
```

.

Mediante el uso de estas etiquetas y funciones se han implementado las transformaciones T8 y T9 para la generación de código de patrones de preguntas y respuestas, respectivamente, para el SBR-DA AliQAn. Como ejemplo se describe a continuación la transformación T9:

```
[module generateAnswerPattern('http://answerpatternmetamodel/1.0')/]

[template public generateAnswerPattern(a : AnswerPatternModel)]

[file (a.name.concat('.cc'), false, 'UTF-8')]

map<string,TPatronResp> generarPatronesResp()
{
    map<string,TPatronResp> patronesResp;

    [for (p : Pattern | a.hasPatterns)]
    TPatronResp patronResp[p.answerType.name.toUpperFirst()/];
    patronResp[p.answerType.name.toUpperFirst()/].insertarTipo
       ("[for (t : ELexicalType | p.answerType.hasAnswerConcept->first().typeSet)
        separator(' ')][t.toString()/][/for]");
    patronResp[p.answerType.name.toUpperFirst()/].insertarTipoPregunta
       ("[p.answerType.name/]");
    patronResp[p.answerType.name.toUpperFirst()/].insertarBP(insertarSNS2
       ("[for (c : Concept | p.answerType.hasAnswerConcept) separator(' ')]
       [for (s : String | c.idSet) separator(' ')]H [s/][/for][/for]"));
    patronesResp.insert(make_pair("[p.answerType.name/]",
        patronResp[p.answerType.name.toUpperFirst()/]));
    [/for]
}

[/file]

[/template]
```

Este módulo espera como entrada un modelo cuyo metamodelo sea el de patrones de respuesta. La plantilla tienen de entrada un elemento *a* del tipo *AnswerPatternModel*. A continuación, mediante una etiqueta *file*, se indica que el fichero de salida tendrá el mismo nombre que *a* pero terminado en *".cc"* (ya que AliQAn está implementado en C++). A continuación se escriben una serie de líneas de código que se generarán en el fichero, para, a continuación definir un bucle que será el encargado de generar código para cada uno de los patrones del modelo de entrada. Dentro de este bucle principal se encuentra el código necesario para cada uno de los patrones de respuesta, el cuál también necesitará de la definición de más bucles que permitan navegar iterativamente

por el modelo de entrada (elementos de las clases *AnswerType* y *Concept*), generando el código necesario por AliQAn.

De la misma manera se ha definido la transformación para la generación de código en AliQAn de los nuevos patrones de preguntas adaptados al dominio restringido. La transfromación implementada en Acceleo es la siguiente:

```
[template public generateQuestionPattern(q : QuestionPatternModel)]

[for (a : AnswerType | q.questionTypology)]

[file (a.name.concat('.cc'), false, 'UTF-8')]

TGrupo generar[a.name.toUpperFirst()/]() {
    list<TBS> listaBloques;
    TGrupo [a.name/]("[a.name/]");
    string LEltos[a.name.toUpperFirst()/] =
          "[for (c : Concept | a.refersTo)
             separator(' ')]
           [for (s : String | c.idSet)
            separator(' ')]H [s/][/for][/for]";
    [for (p : Pattern | q.hasPatterns)]
    [if (p.answerType.name=a.name)]

    TPatron [p.name/];
    [for (e : Expression | p.hasExpressions->asSequence())]
    [if (e.name.startsWith('sp'))]
    [for (v : String | e.value)]
    [p.name/].insertarSp("[v/]");
    [/for]
    [/if]
    [if (e.name.startsWith('ptdt'))]
    [for (v : String | e.value)]
    [p.name/].insertarPtdt("[v/]");
    [/for]
    [/if]
    [if (e.name.startsWith('vbc'))]
    listaBloques.push_back(insertarVerbo
        ("[for (v : String | e.value)
        separator(' ')]L [v/][/for]
        [for (rc : Concept | e.relatedConcepts)
        separator(' ')][for (s : String | rc.idSet)
        separator(' ')]H [s/][/for][/for]",false));
    [/if]
    [if (e.name.startsWith('sns'))]
    [if (e=p.hasExpressions->asSequence()->last())]
    listaBloques.push_back(insertarSNS
        ("[for (v : String | e.value)
        separator(' ')]L [v/][/for]
        [for (rc : Concept | e.relatedConcepts)
        separator(' ')][for (s : String | rc.idSet)
        separator(' ')]H [s/][/for][/for]",true));
    [else]
    [if (p.hasExpressions->asSequence()->
```

```
            at(p.hasExpressions->asSequence()->
            indexOf(e)+1).name.startsWith('sps'))]
        listaBloques.push_back(insertarSNSconSPSS
           ("[for (v : String | e.value)
           separator(' ')]L [v/][/for]
           [for (rc : Concept | e.relatedConcepts)
           separator(' ')][for (s : String | rc.idSet)
           separator(' ')]H [s/][/for][/for]",false,"#",
           "[for (v : String | p.hasExpressions->asSequence()->
           at(p.hasExpressions->asSequence()->indexOf(e)+1).value)
           separator(' ')]L [v/][/for]
           [for (rc : Concept | p.hasExpressions->asSequence()->
           at(p.hasExpressions->asSequence()->indexOf(e)+1).relatedConcepts)
           separator(' ')][for (s : String | rc.idSet)
           separator(' ')]H [s/][/for][/for]",true));
        [else]
        listaBloques.push_back(insertarSNS("[for (v : String | e.value)
           separator(' ')]L [v/][/for] [for (rc : Concept | e.relatedConcepts)
           separator(' ')][for (s : String | rc.idSet)
           separator(' ')]H [s/][/for][/for]",true));
        [/if]
        [/if]
        [/if]
        [if (e.name.startsWith('sps') and not p.hasExpressions->asSequence()->
           at(p.hasExpressions->asSequence()->indexOf(e)-1).name.startsWith('sns'))]
        listaBloques.push_back(insertarSPS("#","[for (rc : Concept | e.relatedConcepts)
           separator(' ')][for (s : String | rc.idSet)
           separator(' ')]H [s/][/for][/for]",false));
        [/if]
        [/for]
        [p.name/].insertarBS(listaBloques);
        listaBloques.clear();
        [a.name/].insertarPatron([p.name/]);
        [/if]
        [/for]
    } [/file] [/for] [/template]
```

6.3. Conclusiones

En este capítulo se presenta cómo se ha implementado MARAQA, la herramienta basada en Eclipse que da soporte a la propuesta definida en esta tesis. MARAQA se compone de los siguientes módulos:

- Un módulo (módulo 1) que contiene los metamodelos descritos en esta tesis doctoral implementados mediante Ecore: el metamodelo para dominios restringidos, el metamodelo para patrones de preguntas y el metamodelo para patrones de respuestas.

- Un módulo para las transformaciones que permiten la obtención de los modelos (módulo 2).
- Un módulo para las transformaciones entre modelos (módulo 3).
- Un módulo para las transformaciones que permiten generar el código (módulo 4).
  Estos módulos dan soporte a cada una de las partes de la propuesta de adaptación de SBR-DA a dominios restringidos presentada en esta tesis.

Parte IV

Evaluación de la propuesta: aplicación al dominio agrícola

Capítulo 7

Evaluación

En este capítulo, se detallan los diferentes experimentos realizados sobre la propuesta de adaptación de un SBR-DA a un dominio restringido, específicamente, aplicados a un dominio agrícola. Los experimentos se han desarrollado en base a tres objetivos principales:

- Definir un marco comparativo para evaluar nuestra propuesta (ver sección 7.3), identificando los principales problemas en la utilización de un sistema de BR-DA concreto sobre un dominio restringido como el agrícola.
- Demostrar la idoneidad de nuestra aproximación para hacer tolerante al ruido textual a sistemas de recuperación de información en dominios restringidos (ver sección 7.4).
- Evaluar nuestro método de adaptación de sistemas de BR-DA a dominios restringidos (ver sección 7.5).

Para poder llevar a cabo los experimentos hemos empleado diferentes recursos existentes:

- *AliQAn*: sistema de BR-DA.
- *JIRS*: sistema de RI.
- *Agrovoc*: SOC disponible para el dominio agrícola, concretamente un tesauro.
- *Revista Cubana de Ciencia Agrícola (RCCA)*: dominio restringido agrícola.

Además se han creado otros propios para la aplicación de nuestra investigación como:

- *Colección de Preguntas RCCA*: un conjunto de 330 preguntas elaboradas por expertos del dominio.
- *Corpus textual RCCA*: conjunto de ficheros de texto plano (con o sin presencia de ruido textual) a partir de los archivos originales de la RCCA.

Primeramente haremos una descripción de las características y funcionamiento de los recursos (existentes y propios) empleados en todos los experimentos y ejemplos desarrollados a lo largo de esta tesis. Luego describiremos las medidas de evaluación que se emplearán. Finalmente, se detallarán cada uno de los experimentos y los resultados obtenidos.

7.1. Recursos empleados en los experimentos

Es indispensable realizar una breve descripción de los detalles fundamentales de los recursos y herramientas que se emplearán para probar la validez de nuestra propuesta. Además de describir el entorno de aplicación de la misma: una revista científica agrícola.

7.1.1. AliQAn: sistema de BR-DA inicial para la evaluación

En esta sección se detallan las características del sistema de BR-DA monolingüe para la lengua castellana, *AliQAn* (**Ali**cante ***Q****uestion* ***An****swering*).

El objetivo que perseguimos es presentar el punto de partida de nuestras evaluaciones. AliQAn es utilizado, dentro de nuestro trabajo, como *baseline* de nuestros experimentos para demostrar la efectividad de la propuesta de adaptación de sistemas de BR a dominios restringidos. De esta manera nos proporciona un marco comparativo y de evaluación que permite comprobar y valorar la valía de nuestra estrategia.

Un poco de historia... AliQAn ha sido diseñado y desarrollado por el Grupo de investigación en Procesamiento del Lenguaje

y Sistemas de Información (GPLSI[1]) dentro del Departamento de Lenguajes y Sistemas Informáticos (DLSI[2]) de la Universidad de Alicante (UA[3]).

En la edición del 2005 del CLEF (Peters *et al.*, 2006), se presentó el sistema AliQAn a la comunidad científica por primera vez (Roger *et al.*, 2005b). Un año más tarde en el CLEF 2006 (Peters *et al.*, 2007), desarrollando una serie de mejoras, modificaciones y ampliaciones (Ferrández *et al.*, 2006b), AliQAn tomó parte de nuevo en el certamen, quedando situado en ambas ediciones en los primeros puestos de la tarea de BR monolingüe en español (Vallin *et al.*, 2005; Magnini *et al.*, 2006). Por último en el 2008 (Roger *et al.*, 2008; Terol *et al.*, 2008) participó nuevamente enfrentando nuevos retos en función de solucionar preguntas con anáfora, interactuar con un corpus textual con características diferentes (i.e. Wikipedia) y disminuir la cantidad de respuestas inexactas.

Nuestra propuesta de adaptación de sistemas de BR-DA a nuevos dominios restringidos de manera automática, es un aporte importante para AliQAn. Ya que el desarrollo de una estrategia de adaptación de este tipo hasta la fecha no había sido planteada dentro de nuestro grupo de investigación, y que por otra parte, se escapa de las funcionalidades que AliQAn presenta.

Arquitectura del sistema AliQAn En esta sección se detalla la estructura, funcionalidad y fases del proceso de BR desarrolladas dentro del sistema AliQAn.

AliQAn, se basa fundamentalmente en el análisis sintáctico de las preguntas y documentos en donde se realizan las búsquedas, y en el emparejamiento complejo de patrones sintácticos para la clasificación de preguntas y detección de respuestas candidatas. Para llevar a cabo el análisis sintáctico de las preguntas y documentos se utiliza SUPAR (Ferrández *et al.*, 1998), analizador sintáctico desarrollado dentro del grupo, el cual trabaja con las preguntas y documentos etiquetados léxicamente por un PoS tagger como por ejemplo MACO (Acebo *et al.*, 1994b; Ferrández & Peral, 2005).

[1] `http://gplsi.dlsi.ua.es/`
[2] `http://www.dlsi.ua.es/`
[3] `http://www.ua.es/`

El sistema SUPAR realiza un análisis sintáctico parcial que permite identificar la estructura gramatical de las oraciones. Dichas estructuras están compuestas por Bloques Sintácticos (BS) que dan forma a los textos. A partir de la salida del sistema SUPAR se extraen los BS de cada una de las oraciones, formando las unidades sintácticas básicas necesarias para definir los patrones de clasificación de preguntas y los patrones de localización de respuestas.

El procesamiento realizado por SUPAR permite al sistema AliQAn identificar tres tipos de BS diferentes: núcleo verbal (VBC), sintagma nominal simple (SNS) y sintagma preposicional simple (SPS). Además SUPAR tiene otras etiquetas como: preposiciones (sp), pronombres o determinantes interrogativos (PtDt) y cláusulas que componen cada oración (CCC).

En el cuadro 7.1 se muestra un ejemplo donde se pueden observar los BS extraídos del análisis sintáctico de una oración ejemplo.

Oraciones[4]	**BS cada oración**
... El rumen es un ecosistema ...	[SNS, El rumen] [VBC, ser] [SNS, un ecosistema]
... la Heteropsylla cubana permaneció en el cultivo de Leucaena ...	[SNS, la Heteropsylla cubana] [VBC, permanecer] [SPS, en el cultivo de Leucaena]

Cuadro 7.1. Ejemplo de extracción de BS

La Figura 7.1 refleja la arquitectura global del sistema AliQAn, presentada en el trabajo (Ferrández *et al.*, 2006b) dentro del CLEF 2006. Como se puede observar en la figura, la arquitectura se divide en dos grandes fases: la fase de indexación y la fase de búsqueda. A continuación se detalla brevemente cada una de ellas.

Fase de indexación En esta fase, el sistema realiza una preparación de los datos en donde las respuestas van a ser buscadas, con el objetivo de acelerar el proceso de BR. AliQAn lleva a cabo

[4] Oraciones extraídas del conjunto de documentos (corpus de revista agrícola RCCA) propuesto para nuestros experimentos.

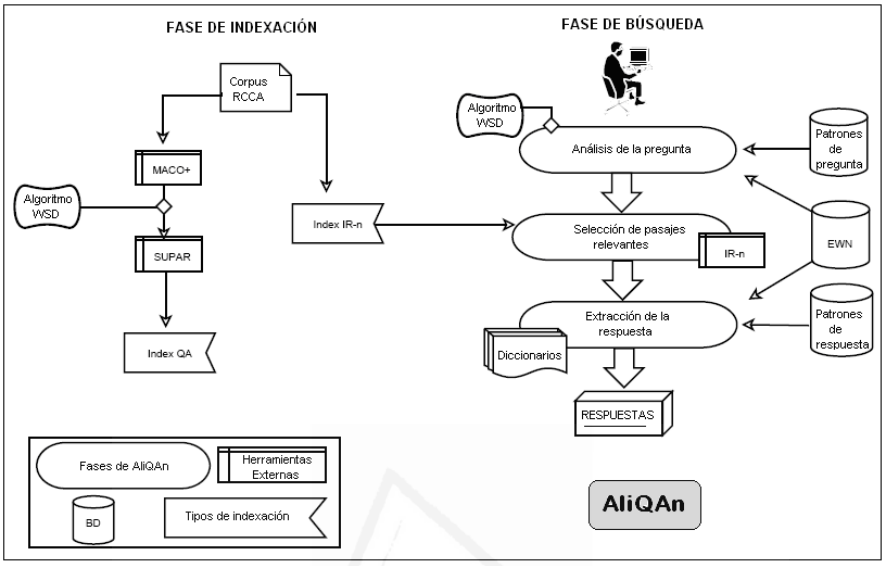

Figura 7.1. Arquitectura general del sistema de BR monolingüe español, AliQAn

dos tipos de indexaciones diferentes, la indexación procesada por un sistema de recuperación de pasajes relevantes (como JIRS (Soriano, 2007) o IR-n (Llopis & González, 2001)), y la indexación realizada para el proceso de BR.

La indexación realizada por el sistema de RI es totalmente independiente al sistema AliQAn, y por eso, no vamos a profundizar en su explicación en este sección.

Por otro lado, en la indexación realizada para el proceso de BR, se almacenada la información sintáctica y semántica de los documentos obtenida a partir de los análisis realizados. En este sentido, se almacenan los SN, SV y SP obtenidos por el sistema SUPAR y los resultados de aplicar un algoritmo de desambiguación semántica (que no detallaremos por no ser un objetivo de nuestra evaluación).

Fase de búsqueda La fase de búsqueda sigue los pasos típicos de todo proceso de BR: análisis de la pregunta, selección de los pasajes relevantes y extracción de la respuesta. En los siguientes puntos de la exposición se detallarán estas tres fases.

Análisis de la pregunta. En esta primera fase del proceso de BR llevado a cabo por AliQAn, se realizan dos tareas principales:

- Clasificar las preguntas en función del tipo de respuesta esperada: esto es, detectar el tipo de información que las posibles respuestas deben satisfacer para ser consideradas por el sistema como respuestas candidatas (nombres propios, cantidades, fechas, etcétera).
- Identificar los BS principales de la pregunta: procesar la pregunta para detectar y extraer los BS principales de la misma, los cuales serán utilizados por el proceso de búsqueda.

La taxonomía, en la que van a ser enmarcadas las preguntas que se le hacen a AliQAn, se ha desarrollado en base a *WordNet Based-Types* y *EuroWordNet Top-Concepts*, definiendo los siguientes tipos: persona, grupo, objeto, lugar (ciudad, capital, país), abreviatura, evento, numérico (cantidad, económico, edad, medida, periodo, porcentaje), temporal (año, mes, día, efeméride, fecha) y definición.

El tipo de respuesta esperada se detecta mediante la aplicación de un conjunto de patrones sintácticos. A la pregunta se le procesan todos los patrones de cada una de las categorías, asignándole a cada categoría una puntuación en función de su similitud con el patrón. Obviamente, una vez hecho esto, se elige la de mayor puntuación. El sistema AliQAn posee un conjunto de alrededor de 200 patrones sintácticos para determinar el tipo de respuesta esperada de cada pregunta, comparando los BS de la pregunta con los BS de los patrones.

En el cuadro 7.2 se puede observar un ejemplo, en base a una pregunta de nuestro corpus de prueba RCCA (ver apéndice A), donde se muestra la extracción de los BS principales de la pregunta y del patrón utilizado para la detección del tipo de respuesta esperada (en este caso es del tipo "persona"). Este es uno de los patrones más relajado de AliQAn para detectar el tipo de pregunta "persona", otros más estrictos son por ejemplo: [quién|quiénes] [VBC, ser] [SNS, hipónimo de persona]. Vale destacar que la última relación es obtenida a partir de las relaciones de hiponimia presentes en EuroWordNet.

Pregunta 162 Corpus RCCA	**BS**	**Patrón**
¿Quiénes han estudiado la miel como fuente energética en terneros destetados?	[Ptdt, Quiénes] [VBC, haber estudiar] [SNS, la miel] [SNS, fuente energética [SPS, en terneros destetados]]	[quién|quiénes] [...]

Cuadro 7.2. Ejemplo de análisis de pregunta por el sistema AliQAn

Selección de los pasajes relevantes. En el segundo paso del proceso de BR los pasajes relevantes, donde las respuestas van a ser localizadas, son creados y recuperados desde la colección de documentos. Este proceso es realizado por alguna herramienta de recuperación de información, como las anteriormente mencionadas en la fase de indexación (JIRS o IR-n).

Los dos sistemas devuelven una lista de pasajes relevantes en los cuales AliQAn va a desarrollar la búsqueda de soluciones. Pero se diferencian en los datos de entrada. Por un lado, el sistema IR-n toma como entrada el conjunto de palabras que forman los BS principales detectados y extraídos en la fase de análisis de la pregunta por AliQAn. Por otro lado, JIRS es un sistema totalmente independiente que toma por entrada la pregunta completa en lenguaje natural (para más detalle ver la sección 7.1.2).

El objetivo de este procedimiento de selección y extracción de información relevante es reducir la complejidad del proceso de búsqueda de la solución, reduciendo la cantidad de información textual a procesar.

Extracción de la respuesta. La extracción de la respuesta supone el último paso dentro del proceso global de BR. El tipo de respuesta esperada, los BS principales de la pregunta y el conjunto de patrones sintácticos para la localización de respuestas candidatas (con información léxica, sintáctica y semántica) son usados con el objetivo de encontrar las respuestas precisas.

La localización de las respuestas concretas en los pasajes, se realiza en dos pasos principales:

- Primero, haciendo uso de los BS de la pregunta y por medio de los patrones diseñados para la extracción de soluciones, un BS es marcado como candidato a solución.
- Segundo, una vez localizado el BS candidato, se le aplican las restricciones léxicas y semánticas (en función del tipo de respuesta esperada) para comprobar que es una posible solución.

El cuadro 7.3 muestra el proceso de localización de la respuesta correcta:

Pregunta 63 Corpus RCCA	¿En qué porciento la disminución en la restricción alimenticia provoca graves trastornos en el fisiologismo del animal y en el peso del feto?
Tipo	numerico_porcentaje
BS de la pregunta	[SP en [PtDt, qué] [SNS, porciento] [SNS, la disminución [SPS, en la restricción alimenticia]] [VBC, provocar] [...]
Frase de los documentos	[...] *una disminución del 30 % al 40 % en la restricción alimenticia ha provocado graves trastornos en el fisiologismo del animal y en el peso del feto* [...]
BS de la frase	[SNS, una disminución [SPS, del **30 % al 40 %** [SPS, en la restricción alimenticia]]] [VBC, provocar] [...]
Respuesta	30 % al 40 %

Cuadro 7.3. Ejemplo de extracción de la respuesta por el sistema AliQAn

Evaluación del sistema AliQAn En este punto, se van a presentar los diferentes experimentos realizados sobre el sistema AliQAn. La evaluación del sistema ha sido efectuada usando los conjuntos de preguntas oficiales del CLEF de las ediciones de 2003, 2004, 2005, 2006 y 2008. A su vez, la localización de las respuestas candidatas ha sido efectuada sobre la colección de documentos formada por el conjunto de noticias de la agencia EFE (de los años 1994 y 1995) que el CLEF propuso para los años del 2003–2006 y sobre los documentos de Wikipedia que fue propuesta a partir del año 2007.

En el cuadro 7.4 se muestran los resultados obtenidos de cada uno de los experimentos.

Edición CLEF	Tarea	Precisión[5] (%)
2003	español–español	54,17
2004	español–español	41,05
2005	español–español	51,5
2006	español–español	50,5
2008	español–español	19,5

Cuadro 7.4. Evaluación del sistema AliQAn con preguntas del CLEF

Si se analiza el Cuadro 7.4 se puede concluir que el sistema AliQAn a promediado una precisión de 49 % en sus primeras cuatro participaciones en el CLEF. Sin embargo, se aprecia un dramático descenso en el año 2008, obteniendo sólo un 19,5 % de precisión (para más detalles remitirse a nuestro trabajo en (Roger *et al.*, 2008)). A pesar de que en esta última participación el sistema mejoró en aspectos como el tratamiento de las preguntas inexactas (resultando sólo 4 preguntas de este tipo) y de las preguntas con anáfora o correferenciadas (obteniendo el segundo lugar en este sentido), se presentaron serios problemas en la adaptación de AliQAn a las características del nuevo corpus Wikipedia. Los problemas estaban relacionados fundamentalmente con la codificación de caracteres no latinos (caracteres no incluidos en la norma ISO 8859-1 o ISO Latín 1) y los textos semiestructurados en Wikipedia. Por lo que se puede concluir que AliQAn, como cualquier otro sistema de BR-DA, presenta dificultades en su adaptación a nuevos dominios de aplicación o cuando debe emplear recursos del conocimiento con diferentes formatos.

7.1.2. JIRS: sistema de RI empleado en la evaluación

El sistema de RI baseline usado para llevar a cabo los experimentos se denomina JIRS (JAVA *Information Retrieval System*) (Soriano, 2007; Buscaldi *et al.*, 2010). JIRS es un sistema de RI y Recuperación de Pasajes (RP) de alta modularidad, escalabilidad y configuración. A parte de realizar búsquedas por los

[5] Para calcular la precisión del sistema se consideran todas las primeras respuestas correctas e inexactas, cuando la inexactitud de éstas últimas es provocada por el exceso de información en la respuesta.

tradicionales métodos de búsqueda basados en términos, permite hacer búsquedas basadas en modelos de *n-gramas*. Este sistema de RI permite como entrada preguntas en lenguaje natural, en lugar de palabras claves o *keywords* y devuelve como salida una lista de pasajes candidatos a contener la respuesta, los cuales están ordenados de manera decreciente según la probabilidad calculada usando un modelo de densidad de distancias de *n-gramas*. En esta distancia todos los pasajes que contienen *n-gramas* con mayor número de términos relevantes tendrán mayor peso que el resto de pasajes.

Precisamente por estas características JIRS es un sistema orientado a la BR haciendo una búsqueda sistemática de todos los *n-gramas* de la pregunta con el fin de encontrar los pasajes con mayor probabilidad de contener la respuesta correcta.

Es válido señalar que JIRS ha sido usado por tres sistemas de Búsqueda de Respuestas (BR) que participaron en el CLEF[6] 2005. Estos sistemas de BR alcanzaron los mejores resultados en las tareas monolingües de español e italiano, y también en las tareas multilingües inglés-español y español-inglés (Soriano *et al.*, 2005).

7.1.3. Tesauro AGROVOC: SOC de dominio restringido

En nuestro caso de estudio de dominio agrícola, usamos el tesauro AGROVOC[7] como Sistema de Organización del Conocimiento (SOC). AGROVOC es un vocabulario multilingüe, estructurado y controlado, elaborado para abarcar la terminología de todos los ámbitos de la agricultura, la silvicultura, la pesca, y las esferas relacionadas con los alimentos (como el medio ambiente, desarrollo sustentable, nutrición, etc.). El tesauro AGROVOC consta de palabras o expresiones (términos), en diversos idiomas, organizados de manera relacional (por ejemplo: término genérico, término específico, y término análogo), para identificar o buscar recursos en el dominio agrícola. Entre estos términos tiene un

[6] CLEF: Cross-Language Evaluation Forum, http://clef-campaign.org/
[7] Agrovoc: `http://www.fao.org/agrovoc/`

número total de 16700 descriptores (i.e. términos genéricos, específicos y análogos), y 10758 no descriptores (i.e. términos que describen para qué se usa un descriptor). El objetivo principal del tesauro es la estandarización del proceso de indexación a fin de facilitar las búsquedas, hacer que resulten más eficaces, y entregar a los usuarios los recursos más pertinentes.

AGROVOC fue desarrollado por la Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO) y la Comisión de Comunidades Europeas a principios de los años 80. El mismo es actualizado por la FAO aproximadamente cada tres meses y las modificaciones específicas pueden ser vistas por los usuarios en el sitio de AGROVOC. Se encuentra disponible en los cinco idiomas oficiales de la FAO: inglés, francés, español, chino y árabe. Está disponible también en checo, alemán, japonés, eslovaco y tailandés, y se encuentra en etapa de traducción o revisión en hindú, húngaro, italiano, coreano y persa.

Profundizando un poco más en AGROVOC podemos señalar que consta de términos compuestos por una o más palabras que representan un único concepto. Para cada término se presenta un conjunto de palabras que muestra la relación jerárquica y no jerárquica que mantiene con otros términos: BT (término más amplio), NT (término más específico), RT (término análogo) y UF o USE (no descriptor). Estas relaciones representan el ámbito y la estructura para el tesauro AGROVOC. Por ejemplo, el término "contaminación" tiene los siguientes términos relacionados:

NT : Disposición de ácidos
NT : Contaminación del aire
NT : Contaminación difusa
NT : Contaminación de sedimentos
NT : Contaminación del agua
RT : Degradación ambiental
RT : Contaminantes
RT : Pesticidas

Otro ejemplo, es el término más específico "contaminación del aire" con las relaciones:

BT : Contaminación

RT : Atmósfera
RT : Efecto invernadero

Estas relaciones entregan el ámbito y la estructura para el tesauro AGROVOC. Por ejemplo, "contaminación" es el término más amplio para "contaminación del aire", cuyos términos análogos son "atmósfera" y "efecto invernadero" que, en su conjunto, definen el ámbito de la información que representan estos términos. Además, AGROVOC emplea anotaciones aclaratorias para explicar el significado y contexto de los términos, si así lo amerita. Los términos taxonómicos y geográficos aparecen señalados con etiquetas para facilitar la búsqueda, selección y descarga empleando filtros.

7.1.4. Revista Cubana de Ciencia Agrícola

La aplicación de nuestra propuesta es sobre un dominio agrícola como la Revista Cubana de Ciencia Agrícola[8] (RCCA). La revista es publicada en idioma inglés y español desde 1966. En este momento la revista RCCA tiene publicados 43 volúmenes, cada uno con un promedio de tres o cuatro números, lo que hacen un total de 140 números y 2000 artículos (28.65 MB como ficheros PDF). Esta revista comprende temáticas relacionadas con la Ciencia Agrícola, como son Pastos y Forrajes o Ciencia Animal. En este trabajo, usamos las ediciones de la revista en idioma español como corpus.

La razón por la que usamos la revista RCCA como corpus es que representa un buen ejemplo de corpus de dominio restringido dentro del contexto de aplicaciones de PLN, ya que cumple las tres condiciones establecidas por Minock en (Minock, 2005): (i) es circunscrito porque la pertinencia de los términos de búsqueda es fácilmente determinada (i.e. las preguntas de los usuarios sólo están relacionadas con la ciencia agrícola), (ii) es complejo ya que utiliza terminología específica (i.e. el corpus contiene una plétora de términos específicamente agrícolas, como "rasgos organolépticos" o "lepidoptera"), y (iii) es práctico porque un grupo

[8] http://www.ica.inf.cu/productos/rcca/

representativamente grande de personas está interesado en ese corpus (i.e. investigadores y estudiantes en la ciencias agrícolas).

Por otro lado, la RCCA publica artículos desde 1966 por lo que un número importante de artículos ha tenido que ser digitalizado: 1479 artículos publicados entre 1967-2000 fueron escaneados y almacenados como ficheros PDF. La obtención y creación de esos fichero PDF requirió la utilización de herramientas OCR para extraer el texto de los documentos en formato duro (i.e. papel) y representan un porcentaje significativo (73.95 % del total). Estos ficheros introducen muchísimo ruido cuando son convertidos a ficheros de texto plano. Con lo cual los experimentos que se describirán más adelante se realizaron usando datos con ruido *real*, en lugar de introducir datos con ruido simulados. Por consiguiente, podemos plantear que nuestro caso de estudio es altamente representativo para la evaluación de nuestra aproximación sobre la estrategia de tolerancia al ruido por sistemas de RI en dominios restringidos.

7.1.5. Corpus textual RCCA

Como algunos autores han manifestado (Hassan & Baumgartner, 2005), el procesamiento automático y la extracción de datos de ficheros PDF no es una tarea sencilla ya que la mayor parte de ellos tiene poca información estructural. Este detalle es aún peor en el dominio científico, donde a menudo se manifiestan problemas con la codificación de los caracteres o con la maquetación (p.e. formato de doble columnas, encabezados y pies de página, conversion de tablas y fórmulas, etc.); introduciendo así muchos errores sintácticos y semánticos en los documentos convertidos. Para crear el corpus textual que usamos en los experimentos, a partir del formato de texto enriquecido de los ficheros PDF de la RCCA, usamos la utilidad de Linux `pdftotext`. En función de tratar con los problemas anteriores se ejecutó dicho recurso con los siguientes parámetros: [-enc encoding] para obtener correctamente la codificación de los caracteres, [-nopgbrk] para evitar la mezcla del encabezado y pie de página con el texto principal, y [-layout]

para mantener la estructura original del texto (o sea, `pdftotext [-layout] [-enc encoding] [-nopgbrk] file.pdf file.txt`).

No obstante a estos intentos por obtener un corpus textual totalmente limpio a partir de los ficheros PDF, la mayoría de los errores que se introdujeron en el mismo fueron de tipo ortográfico o tipográfico, debido al uso de herramientas OCR (como se estudió previamente en la sección3.1). El 73.95 % de los artículos de la revista fueron creados a partir de herramientas OCR, como explicábamos en la sección anterior, lo cual constituye una cifra importante.

Otro objetivo que nos planteamos en la creación de nuestro corpus de prueba fue la eliminación del ruido en pequeñas partes del corpus. La finalidad era garantizar un marco de comparación en los experimentos para evaluar nuestra propuesta tolerante al ruido para el proceso de RI. Específicamente, estos textos preprocesados manualmente para eliminar el ruido, sirvieron para determinar los límites entre los que debe ubicarse el rendimiento de nuestra aproximación para ser viable (ver sección 7.4). En un total de 150 documentos del corpus fue eliminado manualmente el ruido, cumpliendo con la premisa de que cada uno de esos documentos debía contener al menos una respuesta correcta a las preguntas de prueba descritas en la sección siguiente 7.1.6.

Es importante destacar que a partir de la comprobación y rectificación de esos documentos de manera manual, detectamos algunos de los tipos de ruido presentes en los textos. Haciendo énfasis en los problemas que aparecen comúnmente en el corpus RCCA, se pueden clasificar basándonos en nuestra experiencia previa como: sustitución de caracteres (*n**r**cr**a**sis* en lugar de *n**e**cr**o**sis*), omisión de caracteres (*cocidiosis* en lugar de *coc**c**idiosis*), inserción de caracteres (*plásti**t**co* en lugar de *plástico* o *Cytophaga**a*** en lugar de *Cytophaga*), combinación de caracteres (*solub**k**s* en lugar de *solub**le**s* o ***k**idol**k**tico* en lugar de ***ác**idol**ác**tico*) o descomposición de caracteres (*fermentació**íí*** en lugar de *fermentació**n***). Además, existen otros tipos de errores más difíciles de clasificar debido a su naturaleza altamente aleatoria: (*crrhicrtas* en lugar de *cubiertas* o *utilizaci,$n* en lugar de *utilización*). Los errores tipográficos también pueden aparecer debido a los saltos de líneas, interrup-

ción de las oraciones, división en sílabas u omisión del espaciado; por ejemplo (*esponjasintravaginales* en lugar de *esponjas intravaginales*). Concluir que todos estos errores afectan la sintaxis de las oraciones, la morfología de las palabras y por tanto en muchas ocasiones el sentido semántico de los textos.

7.1.6. Colección de preguntas RCCA

Para la realización de los experimentos que en este capítulo se exponen, se crearon un conjunto de preguntas en español sobre el corpus de la RCCA. Estas preguntas fueron creadas a partir de entrevistas a expertos en el dominio agrícola en Cuba: comité editorial de la revista, trabajadores e investigadores del Instituto de Ciencia Animal y profesores de Ingeniería Agrónoma en la Universidad de Matanzas. El número de preguntas asciende a 330, de las cuales se emplearon 150 (ver las primeras 150 preguntas del Apéndice A) para los experimentos sobre la estrategia tolerante al ruido. Esto se debe a que estas 150 preguntas tenían su respuesta en los textos que se preprocesaron manualmente para eliminar el ruido. De esta forma se pueden establecer los límites para medir la efectividad de nuestra aproximación (ver la sección 7.4). La respuesta de 56 de esas preguntas aparecen libres de ruido en el corpus, mientras que la respuesta de 94 de ellas están afectadas por el ruido. Algunos ejemplos de preguntas son: *¿Qué es la necrosis cerebrocortical?* o *¿Qué produce la cytophaga?*. Aclaremos que el total de las 330 preguntas es usado en la prueba de nuestro método de adaptación de SBR-DA a dominios restringidos.

Las preguntas son fundamentalmente de tipo: factual, definición, causa-efecto, razón-motivo, modo, descripción, numérica, con restricciones temporales y preguntas de tipo lista. Las preguntas de tipo factual son la inmensa mayoría y están basadas en hechos, preguntando por nombres de plantas, microorganismos, sustancias, fármacos, compuestos químicos, comidas, fauna, persona; por una localización, por el día en el que sucedió algún hecho, etcétera. Por otro lado, las preguntas de tipo lista esperan como respuesta una lista de personas, objetos, cantidades numéricas o fechas.

7.2. Medida de evaluación

Siguiendo por analogía la estrategia de evaluación del CLEF, catalogaremos cada solución devuelta en nuestros experimentos entre los siguientes tipos:

- **Correcta:** la respuesta devuelta por el sistema es exacta y no contiene elementos o componentes extra.
- **Incorrecta:** pueden ocurrir dos situaciones:
  1. Las respuestas devueltas por el sistema no responden a la pregunta formulada.
  2. Las preguntas no han sido contestadas aunque exista solución en los documentos.
- **Inexacta:** la respuesta devuelta por el sistema:
  1. Incluye algún elemento o componente extra de lo que sería la solución correcta.
  2. Le falta algún elemento con respecto a lo que debería ser la solución correcta.
- **No soportada:** son soluciones correctas descubiertas al azar, es decir, el sistema responde una solución correcta que ha sido extraída de documentos de los que no se puede inferir la respuesta.

En nuestro caso, el cálculo de la precisión de las diferentes ejecuciones que veremos en las secciones siguientes se ha efectuado teniendo también en cuenta como soluciones correctas las soluciones inexactas cuando éstas generan una respuesta correcta con información extra.

7.3. Marco comparativo para evaluar nuestra propuesta: experimentos previos

La propuesta de evaluación que en este capítulo se presenta, surge con diferentes objetivos: (i) identificar los principales problemas en la utilización de un sistema de BR-DA concreto sobre un dominio restringido dado, para así (ii) confirmar los problemas pendientes, sobre adaptación de sistemas de BR, planteados

en el análisis del estado de la cuestión en el capítulo 3, y (iii) definir un punto de comparación con el resto de evaluaciones que se harán del método de adaptación propuesto en este trabajo de investigación.

Para realizar este experimento se usaron las 330 preguntas y el corpus de la RCCA explicados anteriormente; y como sistemas *baseline* o de partida empleamos: JIRS como sistema de RI y AliQAn como sistema de BR-DA. Seguidamente, resumiremos los problemas principales detectados en este experimento, definiremos una tipología de errores para alcanzar un mayor grado de comprensión, y por último discutiremos los resultados y conclusiones alcanzadas de mayor importancia para esta investigación.

7.3.1. Problemas en la aplicación de JIRS y AliQAn al dominio agrícola

En esta sección describiremos los problemas fundamentales detectados en la ejecución de este experimento haciendo énfasis en aquellos que conciernen a los objetivos de este trabajo.

Errores generados por el ruido textual en el sistema de RI El formato de los documentos con los cuales se elaboró el corpus y el ruido textual introducido en los mismos genera uno de los problemas más representativos de todos los que analizaremos, ya que se origina en las fases iniciales y sus efectos son trasladados al resto de las fases.

La mayoría de los sistemas de RI están diseñado para interactuar con colecciones de documentos que asumen tendrán unas frases con una estructura correctamente definida en cuanto a signos de puntuación, finales de oración, saltos de líneas bien marcados y sin ningún ruido textual. Se basan en la asunción de que la redundancia de información, en los corpus grandes, evita los malos resultados en las respuestas que brinda el sistema sobre corpus ruidosos. Por ende, los errores detectados que se describirán seguidamente tienen consecuencias funestas, principalmente sobre dominios restringidos, en los resultados del proceso de RI: (i) pasajes de texto devueltos de gran extensión, debido a la interrupción de determinadas frases por la presencia de caracteres

erróneos y a la imposibilidad de detectar el final de línea, y (ii) pérdida de la respuesta debido a que el sistema de RI no puede hallar los términos claves de la consulta realizada por el usuario, ya que presentan ruido en el corpus. Este último caso es el que peores consecuencias acarrea y sobre el cual se centra nuestra propuesta de tolerancia al ruido.

Errores por insuficientes Tipos de Respuesta Esperada Con anterioridad se analizó qué tipos constituían la taxonomía de Tipos de Respuesta Esperada (TRE) de AliQAn, a partir de los cuales sería capaz de clasificar la pregunta y encontrar su respuesta. En el sistema *baseline* de BR esta tipología estaba agrupada de la siguiente forma, para interactuar con el dominio de noticias periodísticas: persona, grupo, objeto, profesión, lugar (ciudad, capital, país), abreviatura, evento, numérico (cantidad, económico, edad, medida, periodo, porcentaje), temporal (año, mes, día, efeméride, fecha), definición y tipos especiales (email, teléfono, fax). Esa clasificación era apropiada en los inicios de su implementación y para dominios abiertos, pero se ha demostrado que es insuficiente para la aplicación de AliQAn al entorno agrícola.

Para fundamentar lo dicho se citan algunos ejemplos de tipos de respuestas esperada que no son consideradas por AliQAn, a partir del análisis de algunas preguntas (ver todas las preguntas en el anexo A).

- *Pregunta 1* "¿Cuáles son los metabolitos principales que vienen del tracto digestivo?", actualmente clasificada como *profesión* en lugar de *compuestos bioquímicos*.
- *Pregunta 2* "¿Cuáles son los tejidos principales involucrados en la lipogénesis?", actualmente clasificada como *evento* en lugar de *partes del cuerpo*.
- *Pregunta 41* "¿Cuál fue el factor limitante en la disminución del rendimiento de la leguminosa alfalfa?", actualmente clasificada como *persona* en lugar de *factores climáticos*.

Errores en los patrones de pregunta En las pruebas realizadas algunas preguntas han sido clasificadas de forma incorrecta, a pesar de que su tipo se encuentra dentro de la taxonomía de TRE inicial de AliQAn. Eso nos confirma su ineficiencia en el

entorno agrícola. Citaremos algunas situaciones que ejemplifican esta afirmación.

- *Pregunta 42* "¿en qué estación en Cuba se desarrollaron las pruebas de campo para el control de las plagas?" se refiere claramente a un período del año, asociado al término estación, pero el sistema en lugar de clasificarlo como *numerico_periodo*, lo clasifica como *lugar*. Esta situación en concreto ha sido provocada por la falta de información en la formulación propia de la pregunta y porque consideramos que el contexto puede ayudar a identificar el sentido práctico y significado de ella. En contraste ha sido correcta la clasificación según la programación en el sistema.
- *Pregunta 16* "¿A partir de qué edad se cebaron toros Brahman, Charolais, Criollos y Santa Gertrudis?", está asociada al concepto numérico edad y ha sido clasificada como un *objeto* (entidad_objeto) y no al tipo correcto *edad* (numerico_edad). Este es un caso de las preguntas que han tenido una asignación errónea del tipo, estando este codificado correctamente en el *baseline*.
- *Pregunta 15* "¿Cuáles fueron las plagas de artrópodas de principal importancia?", el término plagas que debe estar asociado directamente a *organismos*, aparece unido a la categoría *persona*. Este problema en concreto, es provocado por la terminología propia del dominio agrícola. Precisamente este tipo de problema es el que influirá en cualquier sistema de BR–DA que se aplique este dominio. Por ello, enfrentar este tipo de situación fue uno de los objetivos de este trabajo y demostraremos la efectividad de nuestra propuesta en este mismo capítulo.

Errores en los patrones de extracción de la respuesta Otro de los problemas graves generados por el sistema *baseline* es lo concerniente a la definición de sus patrones de búsqueda de respuesta o a la ausencia de los mismos para satisfacer las necesidades de información en un dominio agrícola como la RCCA. Por tanto, es importante solucionar la situación de que algunos patrones existentes y codificados para la clasificación de las preguntas no se ajusten de manera correcta a las diferentes opciones que tienen

los tipos de preguntas de AliQAn, en el momento de realizar la búsqueda de la respuesta. Por un lado, los patrones de preguntas logran clasificar correctamente la pregunta gracias a los pronombres interrogativos no ambiguos (dónde, quién, cuándo, cuánto) pero en realidad no capturan de forma amplia todas las opciones que incluye el nuevo dominio. Por otro lado, en los patrones de extracción de la respuesta si se aprecian totalmente los errores generados, ya que es imposible para el sistema devolver la respuesta correcta. Ejemplo de esto son las preguntas:

- *Pregunta 40* "¿Dónde fueron introducidas las heces de ganado como suplemento vitamínico?"
- *Pregunta 46* "¿Dónde depositan los huevos las hembras del Trichogramma?"

Las dos preguntas fueron clasificadas como *entidad lugar*, sin embargo ese patrón de AliQAn sólo enmarca los siguientes casos: metrópolis, ciudad, centro urbano, capital, ubicación, situación, sitio, radicación, posición, lugar, localización, emplazamiento, colocación, puesto, terreno, tierra, agua, formación geológica, cuerpo celeste, construcción, y túnel. Pero no incluye otros conceptos que son más propios del dominio dentro de este patrón, por ejemplo: *dietas*, *especies*, *partes del cuerpo*, *procesos*, *zonas climáticas*, etc. Por estas razones no se encuentra finalmente la respuesta, ya que esas preguntas si que se refieren a lugares, pero el patrón de AliQAn no abarca los nuevos conceptos del dominio.

7.3.2. Tipología de errores

En esta sección mostraremos una tabla que representa la tipología de los errores detectados en el proceso de aplicación del sistema de BR-DA de partida, AliQAn, al dominio restringido agrícola de la RCCA. Lo hacemos con el objetivo de alcanzar un mayor grado de comprensión, y para ofrecer datos como la cantidad de preguntas que fueron afectadas por cada uno de los problemas detectados y el por ciento que representan sobre el total de 330 preguntas (ver las preguntas en el apéndice A). De esta forma queda evidenciada la necesidad de aplicar un método de

adaptación al sistema AliQAn para que funcione correctamente en el dominio agrícola.

En el Cuadro 7.5 se encuentra la descripción del problema generado por cada uno de los errores detectados, junto al por ciento que representa del total de las preguntas. Es válido aclarar que hay errores que son generados a partir de errores más generales. Este es el caso de la incorrecta asignación del tipo de respuesta esperada, ya que se debe al fallo de los patrones de preguntas y a la insuficiencia de la taxonomía de los tipos de respuesta esperada del SBR-DA *baseline*.

Cuadro 7.5. Problemas generados en la aplicación de AliQAn al dominio agrícola.

Nº Error	**Motivo del problema**	**%**
1	Fallos en la RI (por el ruido textual)	28,5
2	Insuficientes TRE	36
3	Fallos en los patrones de pregunta	16
4	Incorrecta asignación del TRE	52
5	Fallos en los patrones de respuesta	15

7.3.3. Discusión

Como resultado de este experimento y de su preparación, quedaron las bases sentadas para el resto de experimentos, tanto por: la elaboración de los recursos que se emplearán, la determinación del dominio de aplicación, la detección de los problemas fundamentales que surgen de la aplicación de un sistema de BR-DA a dominios restringido, y los valores de precisión alcanzados por el sistema AliQAn en el dominio que se estudia, para ser usado como marco de evaluación.

Precisamente, la precisión del sistema AliQAn sobre el corpus y la colección de preguntas creadas fue de un 12,8 %. Una cifra muy baja comparada con el valor promedio de 43 % que siempre alcanzó en los certámenes del CLEF.

Por último, como resultado también de este experimento podemos reafirmar los problemas que son necesarios dar solución para obtener un SBR-DA adaptado al dominio agrícola: la taxonomía de TRE del SBR-DA, (ii) tratamiento de la terminología

propia del dominio, (iii) ineficacia de los patrones de preguntas y respuestas del SBR-DA y (iv) el ruido textual en el corpus. Por tanto, AliQAn es un buen ejemplo de SBR-DA para evaluar nuestra propuesta de adaptación a dominios restringidos.

7.4. Evaluación de la estrategia de tolerancia al ruido textual para Sistemas de Recuperación de Información en Dominios Restringidos

En el presente capítulo se describen los experimentos realizados para mostrar la idoneidad de nuestra distancia DM para comparar multipalabras y de nuestra aproximación para la recuperación de la información correcta desde un corpus ruidoso de dominio restringido como el agrícola. La finalidad del primer experimento es ilustrar el comportamiento de la distancia DM a la hora de comparar palabras simples y multipalabras, y así confirmar su utilidad dentro de nuestra propuesta de tolerancia a ruido de un sistema de RI. El segundo experimento tiene como objetivo la determinación de los valores de varias medidas para medir el desempeño del sistema de RI *baseline* cuando el corpus está limpio (máximo valor) y cuando el corpus presenta ruido (mínimo valor). Así se fijaron los valores mínimos y máximos del umbral en el cual deben encontrarse los resultados de nuestra propuesta para ser válida. Se muestra con este experimento que el valor mínimo decrece drásticamente en comparación con el máximo valor. Luego, el tercer experimento tiene como finalidad mostrar que nuestra aproximación es útil para incrementar el valor mínimo del sistema de RI *baseline* hasta obtener valores similares al valor máximo al usar un corpus ruidoso.

Para llevar a cabo estos experimentos se utilizaron las 150 preguntas (ver las 150 primeras preguntas del apéndice A) que tenían la respuesta dentro de los documentos que fueron manualmente preprocesados para eliminar el ruido (ver secciones 7.1.5 y 7.1.6). Se tomaron sólo estas preguntas, ya que era la forma más fiel

de comprobar la efectividad de nuestra propuesta calculando los valores mínimos y máximos explicados con anterioridad.

7.4.1. Experimento para medir la efectividad de la distancia *DM*

En este primer experimento el objetivo fundamental es demostrar la efectividad de nuestra distancia DM para realizar comparaciones entre palabras simples y multipalabras. Se realiza una comparación con otras de las distancias que se analizaron en el capítulo 4. Para llevar a cabo el experimento se tomaron 66 palabras relevantes a una palabra simple "bacteria". Del total de preguntas 47 eran multipalabras.

En el Cuadro 7.6 se muestran las primeras 15 palabras recuperadas por cada distancia empleada en la evaluación (DM, Levenshtein, Jaro-Winkler, Needleman-Wunsch). Pudiéndose apreciar los buenos resultados de nuestra distancia DM.

Cuadro 7.6. Palabras recuperadas por las diferentes distancias con el pivote "bacteria".

Pos.	DM	Levenshtein	Jaro-Winkler	Needleman-Wunsch
1	Bacteria	Bacteria	Bacteria	Bacteria
2	Bactericidas	Vateria	Vateria	Suctoria
3	Bacterinas	Bacterinas	Rizobacterias	Listeria
4	Bacteriosis	Bacteremia	Bacteremia	Cacoecia
5	Bacteremia	Pasteuria	Bacterinas	Wisteria
6	Bacteria butírica	Pantera	Nectarina	Victoria
7	Bacteria acética	Wisteria	Micobacterias	Laccaria
8	Bacteria anaerobia	Victoria	Aberia	Pouteria
9	Bacteria aerobia	Listeria	Bactra	Barleria
10	Bacteria metanógena	Suctoria	Mactra	Vateria
11	Bacterias acidopropiónicas	Laccaria	Aceria	Pasteuria
12	Bacterias antagonistas	Bactra	Bactericidas	Cyanobacteria
13	Bacterias nitrificantes	Bactris	Abamectina	Koeleria
14	Bacterias metanotróficas	Mactra	Lacertidae	Glyceria
15	Bacteria coliforme	Barleria	Ictericia	Blumeria

Por otro lado, al analizar los gráficos 7.2 y 7.3 se ratifica lo que veníamos diciendo de los buenos resultados. Específicamente, recuperar 52 de las palabras relevantes seleccionadas por un experto del dominio para el experimento. En cuanto a la cobertura,

lo mismo, DM alcanza resultados muy superiores al resto de las distancias evaluadas.

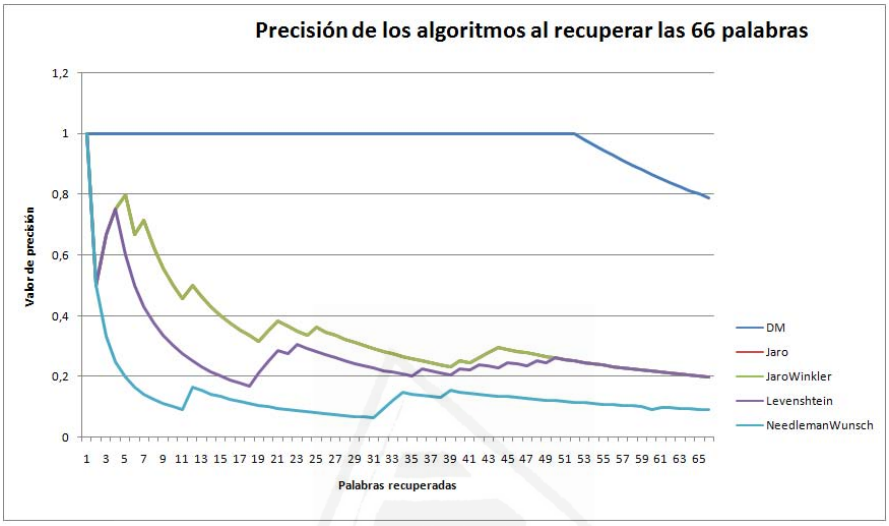

Figura 7.2. Comparación entre DM y otros algoritmos de distancia entre palabras (precisión a las n palabras recuperadas).

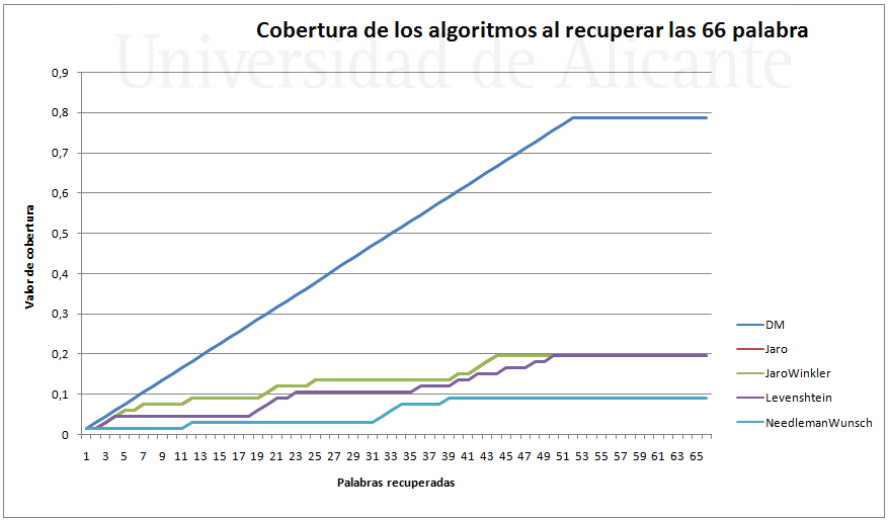

Figura 7.3. Comparación entre DM y otros algoritmos de distancia entre palabras (cobertura a las n palabras recuperadas).

Después de realizar este pequeño experimento con el objetivo de ilustrar la efectividad de DM con las comparaciones entre palabras, da igual si son o no simples, continuaremos con la descripción de los experimentos realizados para medir nuestra aproximación para incluir tolerancia a ruido en sistemas de RI en dominios restringidos. Vale destacar que los valores de DM que se definieron empíricamente, después de realizar varios estudios preliminares, para considerar a dos palabras como correlacionadas y sustituir a una por otra fueron una mínima de distancia 0 y una máxima de 0,37

7.4.2. Experimento para determinar los valores límites

Este primer experimento tiene como objetivo obtener los valores máximos y mínimos de diferentes medidas de evaluación del rendimiento al usar JIRS (nuestro sistema *baseline* de RI). Estos valores serán usados para mostrar la idoneidad de nuestra aproximación con posterioridad en el segundo experimento.

En el Cuadro 7.7 se muestran algunos resultados de este experimento. Se obtuvieron un total de 432,997 pasajes y 180,460 términos del dominio como resultado de aplicar el proceso de indexación de los documentos en todo el corpus. Por otro lado, aplicando este proceso a los 150 documentos preprocesados manualmente se obtuvieron 6,795 pasajes y 10,437 términos de dominio (1,894 de esos términos contienen ruido). Por lo tanto, alrededor del 18 % de los términos presentan ruido.

El experimento fue llevado a cabo usando 56 preguntas que no estaban afectadas por el ruido y 94 preguntas cuyas respuestas presentaban ruido. La cantidad total de pasajes recuperados y de pasajes relevantes recuperados para todas las preguntas se muestra en el Cuadro 7.7. Se hace necesario aclarar que se decidió devolver 25 pasajes por pregunta con la finalidad de analizar apropiadamente los resultados y la posición asignada a las respuestas correctas.

Este experimento se desarrolla en dos partes: primeramente se obtiene el mejor desempeño de JIRS usando el corpus que ha sido preprocesado para remover parcialmente el ruido. Explicar

que haremos referencia a esta fase con las siglas CPB, expresando así que se realiza el experimento sobre el **C**orpus **P**reprocesado y usando el sistema **B**aseline de RI (en nuestro caso JIRS). En segundo lugar, se obtiene el peor desempeño que puede tener el sistema baseline de RI usando el **C**orpus **R**uidoso, al cual llamaremos CRB de manera análoga al anterior. Por consiguiente, los resultados de CPB y CRB fijan los valores máximos y mínimos en el umbral que deben encontrarse los resultados de nuestra propuesta para ser apropiada. Por ejemplo, analizando los resultados de este experimento, se puede concluir que los documentos relevantes recuperados para todas las preguntas usando nuestra aproximación deben estar entre 79 y 130.

El resto de los resultados obtenidos en este experimento se muestran en el Cuadro 7.8. Específicamente, fueron calculadas la siguientes medidas: precisión (en inglés *precision*), cobertura (*recall*), F1[9], Mean Average Precision (MAP) y Mean Reciprocal Rank (MRR). Los valores en el Cuadro 7.8 muestran que el ruido afecta mucho a los resultados devueltos por el sistema de RI (p.e. $MRR(CPB) = 0{,}77$ vs. $MRR(CRB) = 0{,}22$, apreciándose una disminución de 0,55 en el valor de MRR).

Cuadro 7.7. Resumen estadístico de la evaluación de la estrategia de tolerancia al ruido para la RI en dominios restringidos.

Características de la colección	
Número de Documentos en la Colección:	2000
Número de Pasajes en la Colección:	432997
Número de Preguntas:	150
Número total de pasajes sobre todas las preguntas	
Recuperados (25 por pregunta):	3750
Relevantes:	150
Relevantes Recuperados (RR)	
Corpus Preprocesado + Baseline (CPB)	130
Corpus Ruidoso + Baseline (CRB)	79
Corpus Ruidoso + Propuesta (CRP)	110

[9] $F1 = 2 * \frac{precision * recall}{precision + recall}$

Cuadro 7.8. Resultados de la evaluación de la estrategia de tolerancia al ruido para la RI en dominios restringidos.

Medias a Nivel de Pasajes						
RR (# pasajes)				**Precisión**		
CPB	**CRB**	**CRP**		**CPB**	**CRB**	**CRP**
110	20	105	En 1 pasaje	0.73	0.13	0.70
118	25	108	En 2 pasajes	0.39	0.08	0.36
122	28	110	En 3 pasajes	0.27	0.06	0.24
125	58	110	En 5 pasajes	0.16	0.07	0.14
130	71	110	En 10 pasajes	0.08	0.04	0.07
130	75	110	En 15 pasajes	0.05	0.03	0.04
130	79	110	En 20 pasajes	0.04	0.02	0.03
MAP no-interpolada				0.77	0.22	0.71
Otras Medidas						
Cobertura Global				0.86	0.52	0.73
F1				0.06	0.04	0.06
MRR				0.77	0.22	0.71

7.4.3. Experimento para evaluar nuestra propuesta

El objetivo principal de este experimento es la evaluación de la efectividad de nuestra propuesta a través de la comparación de sus resultados con los resultados previamente obtenidos de manera explícita y exhaustiva. Los resultados esperados deben encontrarse entre los valores obtenidos en el experimento previo, específicamente, mientras más cerca estén del valor obtenido en la primera parte del experimento previo mejor será la funcionalidad de nuestra propuesta.

Los resultados de este experimento ($MRR(CRP) = 0{,}71$) se pueden apreciar en la columna CRP (empleando **C**orpus **R**uidoso y nuestra **P**ropuesta) del Cuadro 7.8. Analizando los resultados podemos decir que mejoran considerablemente los valores obtenidos por el sistema de RI *baseline* en el experimento previo usando el corpus ruidoso ($MRR(CRB) = 0{,}22$), además de encontrarse cerca de los resultados óptimos retornados por el sistema *baseline* de RI utilizando un corpus limpio ($MRR(CPB) = 0{,}77$).

7.4.4. Discusión de los resultados

Nuestro segundo experimento muestra que el ruido afecta al sistema de RI *baseline* a través de los valores de los límites mínimo

y máximo calculados. Por otro lado, el tercer experimento muestra que nuestra aproximación tiene un correcto funcionamiento en dominios restringidos cuando el corpus presenta ruido, ya que obtiene resultados muy cercanos al límite superior calculado en el segundo experimento.

Se hace importante resaltar que los valores de MAP y F1 mostrados en el Cuadro 7.8 son similares para nuestra propuesta y para el sistema de RI *baseline* sobre el corpus limpio, mientras que la diferencia en la cobertura global es sólo de 0,13 y en el MRR es de 0,06. Haciendo un análisis profundo de todas las medidas evaluadas, se puede detallar que en nuestra aproximación ("CRP") la cobertura se ve afectada porque fueron recuperados 20 pasajes relevantes menos. La principal razón es que algunos términos con ruido no tienen sus homólogos en AGROVOC debido al hecho que están demasiado deformados o no están en el tesauro. No obstante, la media armónica ponderada (F1, en inglés *weighted harmonic mean*) de la precisión y la cobertura alcanzan el mismo resultado en ambos experimentos.

La efectividad de nuestro algoritmo DM para llevar a cabo el emparejamiento (en inglés *mapping*) del tesauro fue también evaluada. Un estudio previo, similar al realizado aquí en el primer experimento, a partir de 500 términos del corpus y de la colección de preguntas, nos mostró que alrededor del 70 % de los términos son ruidosos y el 45 % multi-palabras. Además sólo 200 de los 500 términos no encontraron un término correspondiente para emparejar en AGROVOC: 80 debido a un emparejamiento incorrecto y 120 debido a no encontrarse en AGROVOC. Finalmente, señalemos que los resultados son similares tanto en palabras simples como en multi-palabras, lo cual muestra la viabilidad de nuestro algoritmo DM para tratar con corpus de dominios restringidos caracterizados por la presencia de multipalabras.

7.5. Evaluación del Método de Adaptación de SBR-DA a dominios restringidos

En el presente capítulo se describen los experimentos realizados para mostrar la utilidad de nuestro método de adaptación de sistemas de BR a dominios restringidos, empleando como caso de estudio el dominio agrícola de la RCCA. Específicamente se realizarán dos experimentos en función de comprobar nuestra aproximación para: (i) generar una taxonomía de Tipos de Respuestas Esperadas ajustada al dominio restringido, y (ii) adaptar los patrones de pregunta y respuesta de un SBR-DA a un dominio restringido.

7.5.1. Experimentos sobre la generación de taxonomías de TRE para dominios restringidos

El corpus usado para este experimento está compuesto por los textos del corpus RCCA creado para realizar las evaluaciones de nuestra propuesta (ver sección previa 7.1.5). El primer paso de nuestro experimento consiste en el procesamiento de este corpus con un etiquetador gramatical como MACO y un etiquetador sintáctico como SUPAR. Luego se realiza el indexado y se calculan las frecuencias de cada término indexado. Para ello nosotros consideramos los términos relevantes como todos aquellos que tienen *fr>25* de frecuencia relativa y *tf-idf>0.01*, así obtuvimos y especificamos en un modelo de dominio restringido 8696 términos relevantes mediante la transformación T1 (ver la sección 5.2.1). Seguidamente, los términos que son sustantivos son usados en la transformación T2 para enriquecer semánticamente el modelo de dominio restringido usando Agrovoc y WordNet.

En la tabla 7.9 se muestra un resumen de nuestro resultados: el modelo de dominio restringido tiene 9022 conceptos de los cuales 3029 son multi-palaras. La mayoría de los conceptos (8530) proviene del SOC de dominio (Agrovoc), 3473 conceptos son extraídos del SOC genérico (Wordnet), y obtuvimos 2981 conceptos que se encuentran representados en los dos SOCs. Todos estos conceptos son representados en el modelo de dominio restringido

enriquecido. Después obtuvimos la taxonomía de TRE desde el modelo de dominio restringido enriquecido, a través de la aplicación de la transformación T3, siguiendo el criterio de selección de todos aquellos conceptos con un número de hipónimos mayor que 2. En la tabla 7.9 se puede apreciar que la taxonomía de TRE contiene alrededor del 10 % de los conceptos del modelo de dominio restringido. Además la taxonomía de TRE obtenida se puede apreciar en el apéndice C.

Cuadro 7.9. Resumen estadístico del Modelo de Dominio Restringido y la taxonomía de TRE creados (# clases semánticas).

	Modelo DR				**Taxonomía TRE**			
Nivel	Agrovoc	WordNet	Multi-palabras	Total	Agrovoc	WordNet	Multi-palabras	Total
0	438	174	212	462	149	77	69	161
1	1382	479	812	1429	133	83	67	149
2	1565	551	568	1627	98	70	40	121
3	1144	486	334	1233	79	77	24	111
4	839	362	250	935	70	68	19	105
5	896	375	261	979	76	53	24	94
6	1002	433	255	1053	66	52	17	82
7	562	291	140	587	37	18	12	43
8	284	156	61	289	32	19	4	32
9	291	95	84	295	11	9	3	13
10	72	41	28	77	5	2	1	5
11	30	17	12	30	4	2	1	4
12	20	11	8	20	1	1	0	1
13	3	1	2	3	0	0	0	0
Total	8530	3473	3029	9022	761	531	281	921

Un estudio previo llevado a cabo con nuestro sistema de BR-DA *baseline* (AliQAn) con 180 preguntas de entrenamiento elaboradas por expertos en el dominio agrícola sobre el corpus RCCA (ver las pregunta entre la número 150 y 330 en los anexos A). En este caso de estudio nosotros analizamos como AliQAn clasifica estas preguntas sobre el dominio restringido agrícola usando su propia taxonomía de TRE (explicada en la sección 7.1.1) siendo 148 preguntas incorrectamente clasificadas (82 %). Los errores en la clasificación de las preguntas fueron causados debido al hecho de que AliQAn tiene una taxonomía de TRE muy pobre para dominios restringidos, como el dominio agrícola en cuestión. El 64 % de estos errores ocurría porque no podía ser reconocido el tipo de respuesta esperada (i.e. 95 preguntas) y el 36 % de los errores se debía a una incorrecta clasificación (53 preguntas son clasificadas como *objeto*, siendo esta clasificación demasiado genérica

para que sea útil en un dominio restringido como el agrícola, permitiendo así que se den respuestas incorrectas). Algunos ejemplos de preguntas incorrectamente clasificadas fueron anteriormente explicados en la sección 5.2.3 y con posterioridad en este capítulo se verán otros ejemplos. Por otro lado, 32 preguntas que representan el 18 % fueron correctamente clasificadas por AliQAn al usar su taxonomía de TRE, específicamente preguntas con tipo de respuesta esperada como: *persona*, *lugar*, *porcentaje numérico*, *cantidad numérica*, *temporal fecha*, y *abreviatura*. Este estudio confirma la idea acerca de que es necesario manejar mayor información semántica y más específica, en los tipos de preguntas más complejos y ambiguos (como las preguntas que comienzan por *Qué* y *Cuál*) en función de incrementar la precisión en su clasificación dentro del dominio restringido.

Finalmente, la taxonomía de TRE obtenida usando nuestra aproximación es incorporada en la taxonomía original del sistema AliQAn y se comprueba que 165 preguntas, que representan el 91.6 %, son correctamente clasificadas a partir de la misma colección de documentos. Quedaron 15 preguntas sin clasificar porque el sistema AliQAn no es capaz de lidiar con preguntas del tipo causa-efecto. Por tanto, el rendimiento del proceso de clasificación de la pregunta de AliQAn en nuestro dominio agrícola se incrementó en un 73.6 %. Específicamente, las preguntas que previamente fueron clasificadas de forma muy genérica como tipo *objeto*, ahora fueron clasificadas de forma más precisa usando la nueva taxonomía de TRE creada para el dominio agrícola por medio de nuestra aproximación: en lugar de clasificarlas como *objeto* fueron clasificadas como algunos de los hipónimos de objeto incluidos en la nueva taxonomía de TRE más refinada.

7.5.2. Experimentos sobre adaptación de patrones a dominios restringidos

Un estudio previo sobre nuestro sistema *baseline* de BR-DA AliQAn (Roger *et al.*, 2008) fue llevado a cabo con 180 preguntas de entrenamiento sobre el corpus de RCCA, como resultado del mismo se detectó que el 52 % de los errores en la adaptación del

sistema estaban causados por la pobre e incorrecta taxonomía de tipos de respuesta esperadas y, por tanto, debido a la insuficiencia de sus patrones de preguntas y respuestas en el dominio de aplicación. Ejemplos de preguntas que estaban afectadas: "*¿cuál fue el factor limitante en la disminución del rendimiento de la leguminosa alfalfa?*" clasificada como "entidad_persona" en lugar de "factores_climaticos". Esta pregunta falla aunque el concepto "factor" puede ser encontrado en el SOC genérico usado por AliQAn (i.e., WordNet), porque no existe ningún patrón asociado a ese concepto. Por consiguiente, desde el modelo de dominio restringido son detectadas palabras altamente frecuentes y con cierta importancia en el corpus (como "viento", "lluvia", "niebla", etc.) sobre el concepto "precipitacion_atmosferica" (desde nuestro SOC de dominio Agrovoc) y también el hiperónimo de este concepto "factores_climaticos" como concepto tope, determinando de esta manera un patrón en el modelo de patrones de preguntas para él.

Otro ejemplo acerca de los problemas de adaptación resueltos por nuestra aproximación se puede ver en la pregunta "*¿Dónde depositan los huevos las hembras del Trichogramma?*" la cual es correctamente clasificada como "entidad_lugar" por medio del pronombre interrogativo *donde*. Sin embargo, los patrones asociados a este tipo de pregunta no están correctamente adaptados al nuevo dominio ya que están sólo relacionados con conceptos como *pais*, *agua*, etc. (por tanto se espera detectar sus hipónimos), pero no con otros conceptos como *partes del cuerpo*, *zonas climaticas*, etc. tan frecuentes en el nuevo dominio.

7.6. Conclusiones

En este capítulo ha quedado demostrada la utilidad de nuestra aproximación para definir una taxonomía de TRE ajustada al dominio, a partir del corpus de documentos y de los SOC disponibles, y cumpliendo que:

- Sea capaz de interactuar con cualquier esquema de representación del SOC realizando la menor cantidad de transformaciones.

- Controle la sobrecarga de la taxonomía con conceptos sin interés para el dominio restringido donde se aplica.
- No sea necesario el estudio de colecciones de preguntas para desarrollar la nueva taxonomía, quitándole a los diseñadores y desarrolladores de sistemas de BR-DR la ardua tarea de realizar este proceso a mano.
- La cobertura de la taxonomía sea realista y elevada ya que el sistema de BR-DR podrá emplear, para realizar la tarea de clasificación de la pregunta, el modelo del dominio restringido a partir del cual se generó la taxonomía de TRE. Con esto queremos decir, que el sistema de BR-DR será capaz de clasificar todos los TRE que contiene la nueva taxonomía.

Finalmente, vale destacar que usando nuestra aproximación, 165 preguntas son correctamente clasificadas, de las cuales 133 daban inicialmente problemas en la clasificación al emplear la taxonomía de TRE original del sistema de BR-DA AliQAn. Luego de ser correctamente clasificadas estas preguntas pueden ser respondidas de manera correcta por el sistema, elevando la precisión del mismo en un 73.6 %.

En el futuro será necesaria la evaluación de nuestro método de adaptación en otros dominios y empleando otros sistemas de BR.

Parte V

Conclusiones finales y trabajos futuros

Capítulo 8

Conclusiones

El principal objetivo de esta tesis es el desarrollo de un método que facilite la adaptación de SBR-DA a dominios restringidos, empleando de manera transparente los recursos de conocimiento del dominio disponibles. Para conseguir este objetivo, una vez se realizó una profunda revisión del estado de la cuestión en los campos relacionados con la investigación, establecimos tres premisas esenciales:

La adaptación de SBR-DA a dominios restringidos debe superar problemas de portabilidad, reusabilidad e integración. Hemos basado nuestra aproximación en la técnica de ingeniería de software de desarrollo dirigido por modelos. Esto implica que nuestro método está diseñado de tal forma que supera los problemas de portabilidad, reusabilidad e integración en el desarrollo de sistemas de Búsqueda de Respuestas en Dominios Restringidos.

Los SBR-DR necesitan de taxonomías de TRE refinadas para aumentar la precisión de sus respuestas. Nuestra aproximación para la creación de taxonomías de TRE para dominios restringidos está basada en la utilización del corpus y de los recursos de conocimiento disponibles en el dominio. Por tanto, se podrá generar una taxonomía refinada de manera automática cada vez que se emplee nuestro método en un nuevo dominio.

Los SBR-DR deben ser tolerantes a la presencia de ruido en sus datos. Definimos una estrategia para adicionar tolerancia a fallos provocados por el ruido textual al proceso de recuperación de información, dentro de la arquitectura de la BR, basándonos en la utilización de una distancia de edición extendida y de un recurso del conocimiento que sirva como vocabulario controlado

e intermedio entre las palabras con ruido y las originales. Esta propuesta es independiente del tipo de ruido presente y del tipo de sistema RI o BR que la utilice.

Hemos definido dos aproximaciones principales para cumplir este objetivo. Primero, desarrollamos una estrategia de tolerancia a ruido para sistemas de recuperación de información en dominios restringidos. Segundo, creamos un método de adaptación de los patrones de preguntas y respuestas y generación de taxonomías de TRE para dominios restringidos. Estas dos aproximaciones pueden considerarse como procesos semi-automáticos donde únicamente es necesaria la supervisión final del desarrollador del sistema de BR-DR que las utilice. En la primera aproximación, el sistema sólo requiere algún SOC o recurso del conocimiento del dominio; no es necesario tener conocimiento de los tipos de ruidos introducidos en el corpus. En la segunda aproximación, el sistema necesita el corpus de documentos sobre el que actuará el sistema de BR, los SOCs disponibles en el dominio y el código fuente de los patrones que emplea el sistema de BR *baseline*, a partir del cual se creará el sistema de BR adaptado al dominio.

En el resto de este capítulo presentamos las conclusiones alcanzadas para cada una de los métodos y discutimos diversas direcciones futuras de investigación. Además presentamos los trabajos de investigación actualmente en progreso. Finalmente presentamos las principales publicaciones derivadas de este trabajo de investigación.

8.1. Principales aportaciones y conclusiones

Con el objetivo de facilitar y clarificar la lectura de las principales aportaciones y conclusiones de nuestro trabajo de investigación, éstas se listan y clasifican por grupos.

- Estrategia de tolerancia al ruido textual en dominios restringidos.
  - Algoritmo de Distancia de Edición eXtendida (DEx) capaz de considerar también las multi-palabras de manera eficiente. Siendo este detalle fundamental en los dominios restringidos.

  - Algoritmo para la incorporación de la estrategia tolerante a ruido propuesta al proceso de Recuperación de Información de manera independiente a su arquitectura empleando algún Sistema de Organización del Conocimiento (SOC) disponible en el dominio.
  - Creación de un corpus con ruido real para evaluación de otros investigadores.
  - Evaluación exhaustiva que demuestra los beneficios de la propuesta.
- Método de adaptación automático de SBR-DA a diferentes dominios restringidos
  - Definición de los metamodelos necesarios para generar modelos de dominio restringido enriquecido semánticamente de manera automática a partir del corpus y empleando los SOCs disponibles. Se emplearon para ello técnicas de desarrollo dirigido por modelos.
  - Creación de taxonomías de Tipo de Respuesta Esperada ajustadas a un dominio restringido de manera automática y sin necesidad de disponer de corpus de preguntas de entrenamiento. El proceso tiene su base en los términos relevantes del corpus y sus conceptos asociados en los SOCs disponibles usando el modelo de dominio restringido enriquecido creado con anterioridad.
  - Adaptación de patrones de preguntas y respuestas del SBR-DA para dominios restringidos de manera automática y semi-supervisada manualmente por el desarrollador del sistema de BR al final del proceso. El método de adaptación tiene su base en el análisis del corpus, de los SOCs disponibles y en la obtención de la codificación de los patrones del sistema. Por tanto, emplea tanto el modelo de dominio restringido enriquecido como la taxonomía de TRE para ese dominio previamente obtenidos. De esta forma se garantizan unos patrones ajustados al dominio, a la terminología que se emplea y a los recursos de conocimiento disponibles.

8.2. Producción científica

El desarrollo del trabajo presentado en esta tesis ha sido publicado en diferentes congresos y foros. Seguidamente enumeramos las principales contribuciones obtenidas.

8.2.1. Búsqueda de Respuestas en Dominios Abiertos

1. **Sandra Roger, Katia Vila, Antonio Ferrández, María Pardiño, Jose Manuel Gomez, Marcel Puchol-Blasco, Jesús Peral: AliQAn, Spanish QA System at CLEF-2008.** ***9th Workshop of the Cross-Language Evaluation Forum, CLEF 2008.*** Este trabajo describe nuestra participación en el foro QA@CLEF 2008, específicamente en la tarea monolingüe de BR español – español. Los principales retos trazados estaban encaminados a lograr una mejora de nuestro sistema de BR (esto es, AliQAn) con respecto a: (i) la capacidad de tratar preguntas relacionadas por una misma temática (preguntas correferenciadas), obteniendo el segundo mejor resultado en este sentido; y (ii) minimizar el número de respuestas inexactas, resultando sólo 4 respuestas inexactas de un total de 200 preguntas.
2. **Sandra Roger, Katia Vila, Antonio Ferrández, María Pardiño, Jose Manuel Gomez, Marcel Puchol-Blasco, Jesús Peral: Using AliQAn in Monolingual QA@CLEF 2008.** ***9th Workshop of the Cross-Language Evaluation Forum, CLEF 2008. Lecture Notes in Computer Science Vol. 5706 , pp. 333-336***
3. **Rafael M. Terol, Marcel Puchol-Blasco, María Pardiño, José Manuel Gómez, Sandra Roger, Katia Vila, Antonio Ferrández, Jesús Peral, Patricio Martínez-Barco: Integrating Logic Forms and Anaphora Resolution in the AliQAn System.** ***CLEF 2008: Lecture Notes in Computer Science, Vol. 5706, pp. 438-441***
4. **Rafael M. Terol, Marcel Puchol-Blasco, María Pardiño, José Manuel Gómez, Sandra Roger, Katia Vila, Antonio Ferrández, Jesús Peral, Patricio Martínez-Barco: AliQAn, Spanish QA System at Multilingual**

QA@CLEF-2008. Integrating Logic Forms and Anaphora Resolution in the AliQAn System. *CLEF 2008.* Este trabajo describe nuestra participación en la tarea multilingüe inglés – español del foro QA@CLEF 2008. Se tuvieron en consideración dos mecanismos posibles en el módulo de traducción: (i) basado en formas lógicas y (ii) basado en técnicas de traducción automática. Los resultados fueron ligeramente mejores al emplear el mecanismo *ii*, pero el mecanismo *i* es un buen método para obtener representaciones de conocimiento independientes del idioma.

8.2.2. Búsqueda de Respuestas en Dominios Restringidos y Recursos de Conocimiento

1. **Katia Vila, Antonio Ferrández: Developing an Ontology for Improving Question Answering in the Agricultural Domain. *Metadata and Semantic Research Third International Conference, MTSR 2009*** En este trabajo presentamos una propuesta para mejorar los resultados de un SBR-DA empleando recursos específicos del dominio, y así obtener un SBR-DR. Para ello se llevaron a cabo los siguientes pasos: (i) crear una ontología que cubra los conceptos del dominio, (ii) enriquecer dicha ontología con recursos de datos públicos, por ejemplo, el tesauro Agrovoc o la base de datos léxica WordNet, y (iii) alinear la ontología semánticamente enriquecida con los artículos del corpus de dominio.
2. **Katia Vila, Antonio Ferrández: Integrating knowledge resources in restricted-domain question answering. *International Journal of Metadata, Semantics and Ontologies (IJMSO)*.** Este trabajo es una extensión del anterior (Vila & Ferrández, 2009), por haber sido seleccionado entre los mejores artículos del congreso. *ESTADO: ENVIADO*

8.2.3. Propuesta de tolerancia al ruido textual en el proceso de RI

1. **Katia Vila, Josval Díaz, Antonio Fernández y Antonio Ferrández: An Approach for Adding Noise-Tolerance**

to Restricted-Domain c Information Retrieval. ***15th International Conference on Applications of Natural Language to Information Systems, NLDB 2010. Lecture Notes in Computer Science Vol. 6177, pp. 1-12.***. Este trabajo has sido seleccionado entre los mejores artículos del evento y se trabaja en su extensión para un número especial de la revista Data & Knowledge Engineering (DKE). En este artículo presentamos una aproximación para añadir a cualquier sistema de RI existente tolerancia a ruidos textuales, empleando recursos de conocimiento de dominios restringidos. La importancia del resultado obtenido en este trabajo recae sobre el hecho de que el ruido textual afecta más los resultados de los sistemas de RI, y por tanto de la BR, sobre corpus de dominios restringidos, ya que esos corpus suelen ser pequeños y sin redundancia.

2. **Noise-tolerance feasibility for restricted-domain Information Retrieval systems.** ***Data & Knowledge Engineering (DKE)***. Este artículo es una extensión del trabajo (Vila *et al.*, 2010). Los aportes fundamentales en esta ocasión son evaluaciones más exhaustivas: aplicando las propuesta a un corpus mayor de preguntas, a dos sistemas de RI para demostrar que es independiente de la arquitectura del sistema que se emplee, explicaciones más detalladas de la distancia propuesta para lidiar con las multipalabras. *ESTADO: ENVIADO.*

8.2.4. Propuesta de adaptación de un SBR-DA a dominios restringidos

1. **Katia Vila, Jose-Noberto Mazón, Antonio Ferrández: Using QVT for adapting question analysis to restricted domain QA systems.** ***Twenty-Second International Conference on Software Engineering & Knowledge Engineering, SEKE 2010***. En este trabajo se presenta una propuesta dirigida por modelos para adaptar a los sistemas de BR a nuevos dominios de manera automática y sistemática, por medio de un conjunto de transformaciones de

modelos establecidas usando el lenguaje QVT (Query/View/-Transformation). Su importancia radica en que la mayoría de los sistemas de BR actuales están orientados a dar respuesta a preguntas de dominio abierto pero, para sean aplicados de forma satisfactoria a escenarios del mundo real, se hace necesaria una adaptación del análisis de la pregunta a las características del dominio de aplicación en concreto.

2. **Katia Vila, Jose-Noberto Mazón, Antonio Ferrández: Model-driven knowledge-based development of expected answer type taxonomies for restricted domain question answering.** ***Fourth Metadata and Semantics Research Conference, MTSR 2010.*** En este trabajo presentamos una aproximación basada en técnicas de desarrollo de software dirigido por modelos, en función de aminorar la tarea de diseñar una taxonomía de Tipos de Respuesta Esperada (TRE) para dominios restringidos, usando recursos de conocimiento heterogéneos. En los dominios restringidos es aún más crucial el diseño de una taxonomía TRE apropiada, ya que los expertos el dominio usarán terminología muy específica del dominio y así las preguntas y respuestas esperadas serán más precisas. La propuesta permite definir nuevas taxonomías de TRE de forma automática y sin necesidad de tener un corpus de preguntas de entrenamiento.
3. **Katia Vila, Jose-Noberto Mazón, Antonio Ferrández: Towards a model-driven approach for using restricted-domain knowledge to adapt question answering systems.** ***Knowledge and Information Systems. An International Journal.*** Este artículo presenta una novedosa aproximación, basada en desarrollo dirigido por modelos, para adaptar automáticamente y con el menor esfuerzo un sistema de BR a nuevos dominios restringidos, integrando la información textual y lo recursos de conocimiento existentes. De esta forma se obtiene un SBR-DR como una interfaz intuitiva que incluye todo tipo de objetos para su funcionamiento, facilitando la interacción del humano con su entorno. *ESTADO: ENVIADO.*

4. **Katia Vila, Antonio Ferrández: Obtaining Precise Information in the Agricultural Domain by using QA systems. *Revista Cubana de Ciencia Agrícola (RCCA)*.**

8.3. Trabajos futuros

En la actualidad se están realizando diferentes trabajos, que toman como punto de partida la investigación que se ha realizado en esta tesis doctoral.

1. Realizar una experimentación más exhaustiva empleando otros sistemas de RI para demostrar que los resultados son independientes del motor de búsqueda, como sucedía al emplear JIRS en las evaluaciones realizadas en este trabajo. Además de aplicarla a otros dominios e idiomas.

Estos trabajos se están realizando actualmente y serán presentados como la extensión del artículo "*An Approach for Adding Noise-Tolerance to Restricted-Domain Information Retrieval*" (Vila *et al.*, 2010) seleccionado para presentar en un número especial de la revista Data & Knowledge Engineering (DKE).

Se han planteado otros trabajos más a largo plazo:

1. Aplicar nuestro método de adaptación de sistemas de Búsqueda de Respuestas a otros dominios restringidos diferentes al agrícola y empleando otros SBR-DA diferentes a AliQAn; para seguir demostrando y evaluando las ventajas de nuestra aproximación. Estas evaluaciones no fueron realizadas para este trabajo ya que no se contaba con los recursos necesarios: otro corpus de dominio restringido y otro SBR-DA.

Parte VI

Bibliografía consultada

1998. Summarization of Imaged Documents without OCR. *Computer Vision and Image Understanding*, **70**(3), 307 – 320.

2006. *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006*. The Association for Computer Linguistics.

Acebo, S.; Ageno, A.; Climet-S.; et al. 1994. MACO: Morphological Analyzer Corpus-Oriented. *In: ESPRIT BRA-7315 Aquilex II Working Paper*.

Acebo, S., Ageno, Alicia, Climent, Salvador, Farreres, Javier, Padró, Lluís, Ribas, Francesc, Rodríguez, Horacio, & Soler, O. 1994a. *MACO: Morphological Analyzer Corpus-Oriented*. Tech. rept. Dept. LSI - Universitat Politécnica de Catalunya.

Acebo, S., Ageno, A., Climent, S., Farreres, J., Padró, L., Placer, R., Rodriguez, H., Taulé, M., & Turno, J. 1994b. MACO: Morphological Analyzer Corpus-Oriented. *ESPRIT BRA-7315 Aquilex II, Working Paper 31*.

Agarwal, Sumeet, Godbole, Shantanu, Punjani, Diwakar, & Roy, Shourya. 2007. How Much Noise Is Too Much: A Study in Automatic Text Classification. *Pages 3–12 of: ICDM*. IEEE Computer Society.

Atkinson, Colin, & Kühne, Thomas. 2003. Model-Driven Development: A Metamodeling Foundation. *IEEE Software*, **20**(5), 36–41.

Aunimo, Lili, Heinonen, Oskari, Kuuskoski, Reeta, Makkonen, Juha, Petit, Renaud, & Virtanen, Otso. 2003. Question Ans-

wering System for Incomplete and Noisy Data. *Pages 193–206 of:* Sebastiani, Fabrizio (ed), *ECIR*. Lecture Notes in Computer Science, vol. 2633. Springer.

Baeza-Yates, Ricardo A., & Ribeiro-Neto, Berthier A. 1999. *Modern Information Retrieval*. ACM Press / Addison-Wesley.

Benamara, F. 2004. *Cooperative Question Answering in Restricted Domains: the WEBCOOP Experiment.* Tech. rept. Institut de Recherches en Informatique de Toulouse, IRIT.

Beydeda, Sami, Book, Matthias, & Gruhn, Volker. 2005. *Model-Driven Software Development*. Springer.

Bézivin, Jean. 2005. On the unification power of models. *Software and System Modeling*, **4**(2), 171–188.

Bobrow, Daniel G. 1964. *Natural Language Input for a Computer Problem Solving System*. Tech. rept.

Brill, E., & R., P. 1994. A rule-based approach to prepositional phrase attachment disambiguation. *Pages 998–1004 of: Proc. 15th International Conference of Computational Linguistics*.

Brill, Eric, Lin, Jimmy J., Banko, Michele, Dumais, Susan T., & Ng, Andrew Y. 2001. Data-Intensive Question Answering. *In: TREC*.

Buscaldi, Davide, Rosso, Paolo, Gómez-Soriano, José Manuel, & Sanchis, Emilio. 2010. Answering questions with an n-gram based passage retrieval engine. *Journal of Intelligent Information Systems*, **34**(2), 113–134.

Chen, Qing, Li, Mu, & Zhou, Ming. 2007. Improving Query Spelling Correction Using Web Search Results. *Pages 181–189 of: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics.

Chung, H. Y.-I. S., Han, K.-S., Yoon, D.-S., Lee, J.-Y., Kim, S.-H., & Rim, H.-C. 2004a. *A Practical QA System in Restricted Domains*. Tech. rept. Dept. of Comp. Science and Engineering, Korea University and Dept. of Comp. Software Engineering, Sangmyung University, Korea.

Chung, H. Y.-I. S., Han, K.-S., Yoon, D.-S., Lee, J.-Y., Kim, S.-H., & Rim, H.-C. 2004b. A Practical QA System in Restricted

Domains. *Pages 39–45 of: Proceedings of the ACL Workshop*. ACL04.

Cucerzan, Silviu, & Brill, Eric. 2004. Spelling Correction as an Iterative Process that Exploits the Collective Knowledge of Web Users. *Pages 293–300 of: EMNLP*. ACL.

Czarnecki, Krzysztof, & Helsen, Simon. 2003 (October). Classification of Model Transformation Approaches. *In: Proceedings of the 2nd OOPSLA Workshop on Generative Technique in the Context of the Model Driven Architecture*.

Damerau, Fred. 1964. A technique for computer detection and correction of spelling errors. *Commun. ACM*, **7**(3), 171–176.

De Pablo, C.; Martínez-Fernández, J.L. ; Martínez P. ; Villena J. ; García-Serrano A.M. ; Goñi J.M.; González J.C. 2004. miraQA: Inicial Experiments in Question Answering. *Pages 371–376 of: Proceedings of Cross Language Evaluation Forum*. CLEF 2004 Workshop.

Doan-Nguyen, H., & K., L. 2005. *Using Terminology and a Concept Hierarchy for Restricted-Domain Question-Answering*. Tech. rept. Department of Computer Science and Software Engineering, Concordia University.

Doan-Nguyen, H., & Keila, L. 2004. The problem of Precision in Restricted-Domain Question-Answering . Some Proposed methods of Improvement. *Pages 8–15 of: Proceedings of the ACL Workshop*.

Ely, John W, Osheroff, Jerome A, Gorman, PaulÑ, Ebell, Mark H, Chambliss, M Lee, Pifer, Eric A, & Stavri, P Zoe. 2000. A taxonomy of generic clinical questions: classification study. *BMJ*, **321**(7258), 429–432.

Esser, Wolfram M. 2004a. Fault-Tolerant Fulltext Information Retrieval in Digital Multilingual Encyclopedias with Weighted Pattern Morphing. *Pages 338–352 of:* McDonald, Sharon, & Tait, John (eds), *ECIR*. Lecture Notes in Computer Science, vol. 2997. Springer.

Esser, Wolfram M. 2004b. Fault-Tolerant Fulltext Search for Large Multilingual Scientific Text Corpora. *Journal of Digital Information*, **6**(1), 1368–7506.

Fernández, Antonio C., Díaz, Josval, Fundora, Alfredo, & Muñoz, Rafael. 2009. Un algoritmo para la extracción de características lexicográficas en la comparación de palabras. *In: IV Convención Científica Internacional de La Universidad De Matanzas CIUM'09.*

Ferrández, A., Palomar, M., & Moreno, L. 1998. Anaphor Resolution in Unrestricted Texts with Partial Parsing. *Pages 385–391 of: COLING-ACL.*

Ferrández, Antonio, Palomar, Manuel, & Moreno, Lidia. 1999. An Empirical Approach to Spanish Anaphora Resolution. *Machine Translation*, **14**(3-4), 191–216.

Ferrández, S., & Peral, J. 2005. Investigating the Best Configuration of HMM Spanish PoS Tagger when Minimum Amount of Training Data Is Available. *In:* (Montoyo *et al.*, 2005).

Ferrández, S., López-Moreno, P., Roger, S., Ferrández, A., Peral, J., Alvarado, X., Noguera, E., & Llopis, F. 2006a. Monolingual and Cross-Lingual QA Using AliQAn and BRILI Systems for CLEF 2006. *In:* (Peters *et al.*, 2007).

Ferrández, S., López-Moreno, P., Roger, S., Ferrández, A., Peral, J., Alvarado, X., Noguera, E., & Llopis, F. 2006b. Monolingual and Cross-Lingual QA Using AliQAn and BRILI Systems for CLEF 2006. *In:* (Peters *et al.*, 2007).

Ferrés, Daniel, & Rodríguez, Horacio. 2006. Experiments adapting an open-domain question answering system to the geographical domain using scope-based resources. *Pages 69–76 of: MLQA '06: Proceedings of the Workshop on Multilingual Question Answering.* Morristown, NJ, USA: Association for Computational Linguistics.

Ferrés, D., Samir Kanaan-Edgar González Alicia Ageno Horacio Rodríguez Mihai Surdeanu Jordi Turmo. 2004. TALP-QA System at TREC 2004: Structural and Hierarchical Relaxing of Semantic Constraints. *In: Proceedings of the Text Retrieval Conference.* TREC 2004 Main QA task.

France, Robert, & Rumpe, Bernhard. 2007. Model-driven Development of Complex Software: A Research Roadmap. *Pages 37–54 of: FOSE '07: 2007 Future of Software Engineering.* Washington, DC, USA: IEEE Computer Society.

Frank, A. H.-U. K., Xu, Feiyu, Uszkoreit, Hans, Crysmann, Berthold, Jörg, Brigitte, & Schäfer, Ulrich. 2005. *Querying Structured Knowledge Sources*. Tech. rept. German Research Center for Artificial Intelligence, DFKI.

Gerber, Anna, Lawley, Michael, Raymond, Kerry, Steel, Jim, & Wood, Andrew. 2002. Transformation: The Missing Link of MDA. *Pages 90–105 of:* Corradini, Andrea, Ehrig, Hartmut, Kreowski, Hans-Jörg, & Rozenberg, Grzegorz (eds), *ICGT*. Lecture Notes in Computer Science, vol. 2505. Springer.

Golder, Carolina, & Gaonach, Daniel. 2003. *Leer y comprender. Psicología de la lectura*. 1 edn. Mexico: Siglo XXI editores.

Green, Bert F., Wolf, Alice K., Chomsky, Carol, & Laughery, Kenneth. 1961. Baseball: an automatic question-answerer. *Pages 219–224 of: IRE-AIEE-ACM '61 (Western): Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*.

H. Doan-Nguyen and L. Keila. 2004. The problem of Precision in Restricted-Domain Question-Answering . Some Proposed methods of Improvement. *Pages 8–15 of: Proceedings of the ACL Workshop*.

Harabagiu, S. G. A. M., & Moldovan, D.I. 1999. WordNet 2 - A Morphologically and Semantically Enhanced Resource. *Pages 1–8 of: Proceedings of ACL-SIGLEX99: Standardizing Lexical Resources*.

Harabagiu, Sanda M., Moldovan, Dan I., Pasca, Marius, Mihalcea, Rada, Surdeanu, Mihai, Bunescu, Razvan C., Girju, Roxana, Rus, Vasile, & Morarescu, Paul. 2000. FALCON: Boosting Knowledge for Answer Engines. *In: TREC*.

Hassan, Tamir, & Baumgartner, Robert. 2005. Intelligent Text Extraction from PDF Documents. *Pages 2–6 of: CIMCA/IAWTIC*. IEEE Computer Society.

Hawking, David, Thistlewaite, Paul B., & Bailey, Peter. 1996. ANU/ACSys TREC-5 Experiments. *In: TREC*.

Hermjakob, Ulf. 2001. Parsing and question classification for question answering. *Pages 1–6 of: Proceedings of the workshop on Open-domain question answering*. Morristown, NJ, USA: Association for Computational Linguistics.

Herzog, Otthein, & Rollinger, Claus-Rainer (eds). 1991. *Text Understanding in LILOG, Integrating Computational Linguistics and Artificial Intelligence, Final Report on the IBM Germany LILOG-Project*. Lecture Notes in Computer Science, vol. 546. Springer.

Hess, M. M., Schwitter, R., Rinaldi, F., & Dowdall, J. 2002. Towards Answer Extraction: an application to Technical Domain. *Pages 460–464 of: ECAI2002, European Conference on Artificial Intelligence*.

Hirschberg, Daniel S. 1977. Algorithms for the Longest Common Subsequence Problem. *J. ACM*, **24**(4), 664–675.

Hodge, Gail. 2000. *Systems of Knowledge Organization for Digital Libraries: Beyond Traditional Authority Files*. The Digital Library Federation Council on Library and Information Resources.

Hopfe, Christina J., Rezgui, Yacine, Métais, Elisabeth, Preece, Alun D., & Li, Haijiang (eds). 2010. *Natural Language Processing and Information Systems, 15th International Conference on Applications of Natural Language to Information Systems, NLDB 2010, Cardiff, UK, June 23-25, 2010. Proceedings*. Lecture Notes in Computer Science, vol. 6177. Springer.

Hovy, Eduard, Hermjakob, Ulf, & Ravichandran, Deepak. 2002. A question/answer typology with surface text patterns. *Pages 247–251 of: Proceedings of the second international conference on Human Language Technology Research*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Hovy, E., Gerber L. Hermajakob U. Junk M. Lin C. 2000. Question answering in Webclopedia. *In: Proceedings of the Ninth Text Retrieval Conference*.

Imamura, Kenji, & Sumita, Eiichiro. 2002. Bilingual corpus cleaning focusing on translation literality. *Pages 1713–1716 of: In: 7th International Conference on Spoken Language Processing (ICSLP-2002*.

J-Y.Nie, & Cai, Jian. 2001 (oct). Filtering noisy Parallel Corpora of Web Pages. *Pages 453–458 of: IEEE symposium on NLP and Knowledge Engineering*.

Jaro, M. A. 1989. Advances in record-linkage methodology as applied to matching the 1985 Census of Tampa, Florida. *Journal of the American Statistical Association*, **84**(406), 414–420.

Jijkoun, V. M. G. a. d. R. M. 2004. The University of Amsterdam at QA@CLEF2004. *In: Proceedings of Cross Language Evaluation Forum*. CLEF 2004 Workshop.

Jing, Hongyan, Lopresti, Daniel, & Shih, Chilin. 2003. Summarization of Noisy Documents: A Pilot Study. *Pages 25–32 of:* Radev, Dragomir, & Teufel, Simone (eds), *Proceedings of the HLT-NAACL 03 Text Summarization Workshop*.

Jones, Karen Spärck. 1999. Automatic Summarizing: Factors and Directions. *Pages 1–14 of:* Mani, Inderjeet, & Maybury, Mark (eds), *Advances in Automatic Text Summarization*. MIT Press.

Jones, Karen Spärck. 2007. Automatic summarising: The state of the art. *Inf. Process. Manage.*, **43**(6), 1449–1481.

Kantor, Paul B., & Voorhees, Ellen M. 1996. Report on the TREC-5 Confusion Track. *In: TREC*.

Kantor, Paul B., & Voorhees, Ellen M. 2000. The TREC-5 Confusion Track: Comparing Retrieval Methods for Scanned Text. *Inf. Retr.*, **2**(2/3), 165–176.

Khadivi, Shahram, & Ney, Hermann. 2005. Automatic Filtering of Bilingual Corpora for Statistical Machine Translation. *In:* (Montoyo *et al.*, 2005).

Kleppe, A., Warmer, J., & Bast, W. 2003. *MDA Explained. The Practice and Promise of The Model Driven Architecture*. Addison Wesley.

Knoblock, Craig A., Lopresti, Daniel P., Roy, Shourya, & Subramaniam, L. Venkata. 2007. Special issue on noisy text analytics. *IJDAR*, **10**(3-4), 127–128.

Knuth, Donald E. 1964. Backus normal form vs. Backus Naur form. *Commun. ACM*, **7**(12), 735–736.

Kosseim, Leila, & Yousefi, Jamileh. 2008. Improving the performance of question answering with semantically equivalent answer patterns. *Data Knowl. Eng.*, **66**(1), 53–67.

Kukich, Karen. 1992. Techniques for Automatically Correcting Words in Text. *ACM Comput. Surv.*, **24**(4), 377–439.

Levenshtein, Vladimir I. 1966. *Binary codes capable of correcting deletions, insertions, and reversals.* Tech. rept. 8.

Levine, J. M., & Fedder, L. 1989. *The theory and implementation of a bidirectional question answerin system.* Tech. rept. 182. University of Cambridge.

Li, Mu, Zhu, Muhua, Zhang, Yang, & Zhou, Ming. 2006. Exploring Distributional Similarity Based Models for Query Spelling Correction. *In:* (DBL, 2006).

Li, Xin, & Roth, Dan. 2006. Learning question classifiers: the role of semantic information. *Nat. Lang. Eng.*, **12**(3), 229–249.

Llopis, Fernando, & González, José Luis Vicedo. 2001. IR-n: A Passage Retrieval System at CLEF-2001. *Pages 244–252 of:* Peters, Carol, Braschler, Martin, Gonzalo, Julio, & Kluck, Michael (eds), *CLEF*. Lecture Notes in Computer Science, vol. 2406. Springer.

Lopresti, Daniel P., Roy, Shourya, Schulz, Klaus, & Subramaniam, L. Venkata. 2009. Special issue on noisy text analytics. *IJDAR*, **12**, 139–140.

López-Cózar, Ramón, & Rubio, Antonio J. 1997. SAPLEN : un sistema de diálogo en lenguaje natural para una aplicación comercial. *Procesamiento del Lenguaje Natural, Revista SEPLN*, 65–81.

Magnini, Bernardo, Giampiccolo, Danilo, Forner, Pamela, Ayache, Christelle, Jijkoun, Valentin, Osenova, Petya, Peñas, Anselmo, Rocha, Paulo, Sacaleanu, Bogdan, & Sutcliffe, Richard F. E. 2006. Overview of the CLEF 2006 Multilingual Question Answering Track. *In:* (Peters *et al.*, 2007).

Manthey, Bodo, & Reischuk, Rüdiger. 2003. The Intractability of Computing the Hamming Distance. *Pages 88–97 of:* Ibaraki, Toshihide, Katoh, Naoki, & Ono, Hirotaka (eds), *ISAAC*. Lecture Notes in Computer Science, vol. 2906. Springer.

Martínez-Santiago, F., & Ureña-López, L.A. 2002 (mayo). Propuesta para un sistema CLIR independiente del lenguaje. *Pages 141–148 of: I Jornadas de Tratamiento y Recuperación de Información (JOTRI-02).*

Martínez-Santiago, F., Diaz-Galiano, M.C., Martín-Valdivia, M.T., Rivas-Santos, V.M., & Ureña-López, L.A. 2002 (mayo). Aplicación de redes neuronales y bayesianas en la detección de multipalabras para tareas IR. *Pages 88–95 of: I Jornadas de Tratamiento y Recuperación de Información (JOTRI-02).*

MDA. 2003. *MDA Guide 1.0.1.* `http://www.omg.org/cgi-bin/doc?omg/03-06-01`.

Mellor, Stephen J., Clark, AnthonyÑ., & Futagami, Takao. 2003. Guest Editors' Introduction: Model-Driven Development. *IEEE Software*, **20**(5), 14–18.

Metzler, Donald, & Croft, W. Bruce. 2005. Analysis of Statistical Question Classification for Fact-Based Questions. *Inf. Retr.*, **8**(3), 481–504.

Miller, David R. H., Boisen, Sean, Schwartz, Richard M., Stone, Rebecca, & Weischedel, Ralph M. 2000. Named Entity Extraction from Noisy Input: Speech and OCR. *Pages 316–324 of: ANLP.*

Miller, G. A. 1990. WordNet: An on-line lexical database. *International Journal of Lexicography*, **3**(4), 235–312.

Minock, Michael. 2005. Where are the "killer applications" of restricted domain question answering? *Page 4 of: Proceedings of the IJCAI Workshop on Knowledge Reasoning in Question Answering.*

Méndez-Díaz, E., Vilares-Ferro J. Cabrero-Souto D. 2004. COLE at CLEF-2004. *Pages 413–418 of: Workshop of Cross-Language Evaluation Forum.* CLEF.

MOF. 2006. *Meta Object Facility (MOF), 2.0* `http://www.omg.org/spec/MOF/2.0/PDF/`.

MOFM2T. 2008. *MOF Model to Text Transformation Language (MOFM2T), 1.0* `http://www.omg.org/spec/MOFM2T/1.0/PDF/`.

Moldovan, Dan I., Pasca, Marius, Harabagiu, Sanda M., & Surdeanu, Mihai. 2003. Performance issues and error analysis in an open-domain question answering system. *ACM Trans. Inf. Syst.*, **21**(2), 133–154.

Mollá, D. R., Schwitter, R., Dowdall, J., & Hess, M. 2003. Answer Extraction from Technical Texts. *IEEE Intelligent Systems*, 12–17.

Mollá, D. y. M. H. 1999. On the Escalability of the Answer Extraction System: ExtrAns. *Pages 219–224 of: Application of Natural Language to Information Systems (NLDB'99).*

Mollá, Diego, & Vicedo, José Luis. 2007. Question Answering in Restricted Domains: An Overview. *Association for Computational Linguistics*, **Special Section on Restricted-Domain Question Answering**, 41–61.

Mollá, Diego, & González, José Luis Vicedo. 2007. Question Answering in Restricted Domains: An Overview. *Computational Linguistics*, **33**(1), 41–61.

Montoyo, Andrés, Muñoz, Rafael, & Métais, Elisabeth (eds). 2005. *Natural Language Processing and Information Systems, 10th International Conference on Applications of Natural Language to Information Systems, NLDB 2005, Alicante, Spain, June 15-17, 2005, Proceedings.* Lecture Notes in Computer Science, vol. 3513. Springer.

Moreno, Lidia, Palomar, Manuel, & Pastor, Moisés. 1993. Interpretación de la comparación en consultas a una base de datos geográfica a través de la lógica. *Procesamiento del Lenguaje Natural, Revista SEPLN*, 259–277.

Needleman, Saul B., & Wunsch, C. D. 1970. A general method applicable to the search for similarities in the amino-acid sequence of two proteins. *Journal of Molecular Biology*, **48**(3), 443–453.

Neumann, G. a. S. B. 2004. Experiments on Robust NL Question Interpretation and Multi-layered Document Annotation for a Cross-Language Question/Answering-System. *Pages 15–17 of: Proceedings of Cross Language Evaluation Forum.* CLEF 2004 Workshop.

Ng, Kwong Bor, Loewenstern, David, Basu, Chumki, Hirsh, Haym, & Kantor, Paul B. 1996. Data Fusion of Machine-Learning Methods for the TREC5 Routing Task (and other work). *In: TREC.*

Niu, Y. ; H., G. 2004. Analysis of Semantic Classes in Medical Text for Question Answering. *Pages 54–61 of: Proceedings of the ACL Workshop.*

Norvig, Peter. 1992. *Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Nyberg, Eric, Mitamura, Teruko, Frederking, Robert, Pedro, Vasco, Bilotti, Matthew W., Schlaikjer, Andrew, & Hannan, Kerry. 2005. Extending the JAVELIN QA system with domain semantics.

OCL. 2010. *Object Constraint Language (OCL), 2.2* `http://www.omg.org/spec/OCL/2.2/PDF/`.

Palmer, David D., & Ostendorf, Mari. 2001. Improving information extraction by modeling errors in speech recognizer output. *Pages 1–5 of: HLT '01: Proceedings of the first international conference on Human language technology research.* Morristown, NJ, USA: Association for Computational Linguistics.

Peñas, Anselmo, Forner, Pamela, Sutcliffe, Richard, Rodrigo, Álvaro, Forascu, Corina, Alegria, Iñaki, Giampiccolo, Danilo, Moreau, Nicolas, & Osenova, Petya. 2009. Overview of ResPubliQA 2009: Question Answering Evaluation over European Legislation. *In: Working Notes of Cross Language Evaluation Forum (CLEF).*

Peters, Carol, Gey, Fredric C., Gonzalo, Julio, Müller, Henning, Jones, Gareth J. F., Kluck, Michael, Magnini, Bernardo, & de Rijke, Maarten (eds). 2006. *Accessing Multilingual Information Repositories, 6th Workshop of the Cross-Language Evalution Forum, CLEF 2005, Vienna, Austria, 21-23 September, 2005, Revised Selected Papers.* Lecture Notes in Computer Science, vol. 4022. Springer.

Peters, Carol, Clough, Paul, Gey, Fredric C., Karlgren, Jussi, Magnini, Bernardo, Oard, Douglas W., de Rijke, Maarten, & Stempfhuber, Maximilian (eds). 2007. *Evaluation of Multilingual and Multi-modal Information Retrieval, 7th Workshop of the Cross-Language Evaluation Forum, CLEF 2006,*

Alicante, Spain, September 20-22, 2006, Revised Selected Papers. Lecture Notes in Computer Science, vol. 4730. Springer.

Peters, Carol, Deselaers, Thomas, Ferro, Nicola, Gonzalo, Julio, Jones, Gareth J. F., Kurimo, Mikko, Mandl, Thomas, Peñas, Anselmo, & Petras, Vivien (eds). 2009. *Evaluating Systems for Multilingual and Multimodal Information Access, 9th Workshop of the Cross-Language Evaluation Forum, CLEF 2008, Aarhus, Denmark, September 17-19, 2008, Revised Selected Papers*. Lecture Notes in Computer Science, vol. 5706. Springer.

Pinchak, Christopher, & Lin, Dekang. 2006. A Probabilistic Answer Type Model. *In: EACL*.

Pérez-Coutiño, M.; Solorio, T.; Montes-y-Gómez Manuel; López-López Aurelio; Villaseñor-Pineda-Luis. 2004. The Use of Lexical Context in Question Answering for Spanish. *In: Proceedings of Cross Language Evaluation Forum*.

QVT. 2005. *MOF 2.0 Query/View/Transformation*. `http://www.omg.org/cgi-bin/doc?ptc/2005-11-01`.

Raphael, Bertram. 1964. *SIR: A computer program for semantic information retrieval*. Tech. rept.

Ravichandran, Deepak, & Hovy, Eduard H. 2002. Learning surface text patterns for a Question Answering System. *Pages 41–47 of: ACL*.

Ráez, Arturo Montejo, Valdivia, María Teresa Martín, Ortega, José Manuel Perea, & López, L. Alfonso Ureña. 2010. Using bigrams detection for text categorization in scientific domain. *Procesamiento del Lenguaje Natural, Revista SEPLN*, marzo, 91–98.

Rice, Stephen V., Nagy, George L., & Nartker, Thomas A. 1999. *Optical Character Recognition: An Illustrated Guide to the Frontier*. Norwell, MA, USA: Kluwer Academic Publishers.

Rinaldi, F. D., Hess, M., Mollá, D., & Schwitter, R. 2004. *Question answering in terminology-rich technical domains*. Tech. rept. New Directions in Question Answering. AAAI Press, Mark Maybury ed.

Rinaldi, F. J. D., M. Hess-K. Kaljurand M. Koit-K. Vider Kahusk. 2002. Terminology as Knowledge in Answer Extraction. *Pa-*

ges 107–113 of: Proceedings of the 6th International Conference on Terminology and Knowledge Engineering. TKE02.

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M.M., & Gatford, M. 1995. OKAPI at TREC-3. *Pages 109–130 of:* D.K.Hartman (ed), *Overview of the Third TExt Retrieval Conference*. TREC 3.

Roger, S., Ferrández, S., Ferrández, A., Peral, J., Llopis, F., Aguilar, A., & Tomás, D. 2005a. AliQAn, Spanish QA System at CLEF 2005. *In:* (Peters *et al.*, 2006).

Roger, S. F., S. ;Ferrández A.-Aguilar A. Peral J.; Llopis F. Tomás-D. 2005. AliQAn, Spanish QA System at CLEF 2005. *In: WorkShop of Cross-Language Evaluate Forum*. CLEF 2005.

Roger, Sandra, Ferrández, Sergio, Ferrández, Antonio, Peral, Jesús, Llopis, Fernando, Aguilar, A., & Tomás, David. 2005b. AliQAn, Spanish QA System at CLEF-2005. *Pages 457–466 of: CLEF*.

Roger, Sandra, Vila, Katia, Ferrández, Antonio, Pardiño, María, Gómez, José Manuel, Puchol-Blasco, Marcel, & Peral, Jesús. 2008. Using AliQAn in Monolingual QA@CLEF 2008. *In:* (Peters *et al.*, 2009).

Roy, Shourya, & Subramaniam, L. Venkata. 2006. Automatic Generation of Domain Models for Call-Centers from Noisy Transcriptions. *In:* (DBL, 2006).

Russell, Stuart J., & Norvig, Peter. 2003. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition.

Sang, Erik Tjong Kim, Bouma, Gosse, & de Rijke, Maarten. 2005. Developing Offline Strategies for Answering Medical Questions. *Pages 41–45 of: Proceedings of the AAAI-05 workshop on Question Answering in restricted domains*.

Sartori, Fabio, Sicilia, Miguel-Ángel, & Manouselis, Nikos (eds). 2009. *Metadata and Semantic Research Third International Conference, MTSR 2009, Milan, Italy, October 1-2, 2009*. Vol. 46. Communications in Computer and Information Science (CCIS), Springer.

Sebastiani, Fabrizio. 2002. Machine learning in automated text categorization. *ACM Comput. Surv.*, **34**(1), 1–47.

Sekine, S., Sudo, K., & Nobata, C. 2002 (May). Extended Named Entity Hierarchy. *Pages 1818–1824 of:* Rodríguez, M. Gonzáles, & Araujo, C. Paz Suárez (eds), *Proceedings of* 3^{rd} *International Conference on Language Resources and Evaluation (LREC'02).*

Sendall, Shane, & Kozaczynski, Wojtek. 2003. Model Transformation: The Heart and Soul of Model-Driven Software Development. *IEEE Software*, **20**(5), 42–45.

Shi, Lixin, & Nie, Jian-Yun. 2006. Filtering or adapting: two strategies to exploit noisy parallel corpora for cross-language information retrieval. *Pages 814–815 of: CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management.* New York, NY, USA: ACM.

Sleator, D., & T., D. 1993. Parsing English with a link grammar. *Pages 277–292 of: Third International Workshop on Parsing Technologies.*

Sobrino, Alejandro. 2004. Interrogación, en lenguaje natural, de una base de datos lógica. *Pages 921–931 of: Actas del VI Congreso de Lingüística General.* Santiago de Compostela: Métodos y aplicaciones de la lingüística, Vol. 1, 2007, ISBN 84-7635-670-8.

Solorio, T. a. L. L. A. 2004. Learning Named Entity Classifiers using Support Vector Machines. *Lecture Notes in Computer Science, Springer-Verlag*, 158–166.

Sopeña, Luis. 1983. USL: Un Sistema para interrogar en castellano a bases de datos relacionales. *Procesamiento del Lenguaje Natural, Revista SEPLN*, 38–41.

Soriano, José Manuel Gómez. 2007. *Recuperación de Pasajes Multilingües para la Búsqueda de Respuestas.* Phd. Thesis, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, Valencia, Spain.

Soriano, José Manuel Gómez, Buscaldi, Davide, Asensi, Empar Bisbal, Rosso, Paolo, & Arnal, Emilio Sanchis. 2005. QUASAR: The Question Answering System of the Universidad Politécnica de Valencia. *In:* (Peters *et al.*, 2006).

Spasic, I. G. N.; Sophia, Ananiadou. 2003. Using Domain-specifics verbs for term classification. *Pages 17–24 of: Proceedings of the ACL Workshop on Natural Language Processing in Biomedicine*.

Subramaniam, L. Venkata, Roy, Shourya, Faruquie, Tanveer A., & Negi, Sumit. 2009. A survey of types of text noise and techniques to handle noisy text. *Pages 115–122 of: AND '09: Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data*. New York, NY, USA: ACM.

Taghva, Kazem, & Stofsky, Eric. 2001. OCRSpell: An Interactive Spelling Correction System for OCR Errors in Text. *Intl. Journal on Document Analysis and Recognition*, **3**(3), 125–137.

Taghva, Kazem, Borsack, Julie, & Condit, Allen. 1996. Effects of OCR Errors on Ranking and Feedback Using the Vector Space Model. *Inf. Proc. and Management*, **32**(3), 317–327.

Terol, R., P., M., M.B., & Palomar, Manuel. 2006. Aplicación de técnicas basadas en PLN al tratamiento de preguntas médicas en Búsqueda de Respuestas. *Revista de la Sociedad Española para el Procesamiento del Lenguaje Natural*, 17–24.

Terol, R. M. P. M.-B. a. M. P. 2005. Applying Logic Forms to Biomedical Q-A. *In: Proceedings of the International Symposium on Innovations in Intelligent Systems and Applications*.

Terol, Rafael M., Puchol-Blasco, Marcel, Pardiño, María, Gómez, José Manuel, Roger, Sandra, Vila, Katia, Ferrández, Antonio, Peral, Jesús, & Martínez-Barco, Patricio. 2008. Integrating Logic Forms and Anaphora Resolution in the AliQAn System. *In:* (Peters *et al.*, 2009).

Thomas, Dave A. 2004. MDA: Revenge of the Modelers or UML Utopia? *IEEE Software*, **21**(3), 15–17.

Tomás, David, & González, José Luis Vicedo. 2007. Multiple-Taxonomy Question Classification for Category Search on Faceted Information. *Pages 653–660 of:* Matousek, Václav, & Mautner, Pavel (eds), *TSD*. Lecture Notes in Computer Science, vol. 4629. Springer.

Tong, Xiang, Zhai, ChengXiang, Milic-Frayling, Natasa, & Evans, David A. 1996. OCR Correction and Query Expansion for

Retrieval on OCR Data – CLARIT TREC-5 Confusion Track Report. *In: TREC*.

UML. 2010. *Unified Modeling Language (UML), 2.3* `http://www.omg.org/spec/UML/2.3/`.

Vallin, Alessandro, Magnini, Bernardo, Giampiccolo, Danilo, Aunimo, Lili, Ayache, Christelle, Osenova, Petya, Peñas, Anselmo, de Rijke, Maarten, Sacaleanu, Bogdan, Santos, Diana, & Sutcliffe, Richard F. E. 2005. Overview of the CLEF 2005 Multilingual Question Answering Track. *In:* (Peters *et al.*, 2006).

Vicedo, J. L., Saiz M.-Izquierdo R. 2004. Does English help Question Answering in Spanish? . *In: Proceedings of Cross Language Evaluation Forum*.

Vila, Katia, & Ferrández, Antonio. 2009. Developing an Ontology for Improving Question Answering in the Agricultural Domain. *In:* (Sartori *et al.*, 2009).

Vila, Katia, Díaz, Josval, Fernández, Antonio, & Ferrández, Antonio. 2010. An Approach for Adding Noise-Tolerance to Restricted-Domain Information Retrieval. *In:* (Hopfe *et al.*, 2010).

Vinciarelli, Alessandro. 2005. Noisy Text Categorization. *IEEE Trans. Pattern Anal. Mach. Intell.*, **27**(12), 1882–1895.

Voorhees, Ellen M. 1999. The TREC-8 Question Answering Track Report. *In: TREC*.

Wagner, Robert A., & Fischer, Michael J. 1974. The String-to-String Correction Problem. *J. ACM*, **21**(1), 168–173.

Weizenbaum, Joseph. 1966. ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the Association for Computing Machinery, Commun. ACM*, **9**(1), 36–45.

Weston, Jason, Mukherjee, Sayan, Chapelle, Olivier, Pontil, Massimiliano, Poggio, Tomaso, & Vapnik, Vladimir. 2000. Feature Selection for SVMs. *Pages 668–674 of:* Leen, Todd K., Dietterich, Thomas G., & Tresp, Volker (eds), *NIPS*. MIT Press.

Wilensky, Robert, Chin, DavidÑ., Luria, Marc, Martin, James H., Mayfield, James, & Wu, Dekai. 1988. The Berkeley UNIX

Consultant Project. *Computational Linguistics*, **14**(3), 35–84.

Winkler, William E. 1999. *The State of Record Linkage and Current Research Problems*. Tech. rept. Statistical Research Division, U.S. Census Bureau, Washington, DC.

Winograd, Terry. 1972. *Understanding Natural Language*. Orlando, FL, USA: Academic Press, Inc.

Wojnar, Ales, Mlýnková, Irena, & Dokulil, Jirí. 2010. Structural and semantic aspects of similarity of Document Type Definitions and XML schemas. *Inf. Sci.*, **180**(10), 1817–1836.

Woods, W. A. 1973. Progress in natural language understanding: an application to LUNAR geology. *Pages 441–450 of: AFIPS '73: Proceedings of the June 4-8, 1973, national computer conference and exposition*, vol. 42. New York, NY, USA: ACM.

Parte VII

Anexos

Apéndice A

Corpus de Preguntas RCCA

A.1. Preguntas sobre dominio agrícola para evaluar BR-DR español–español

1. ¿Cuáles son los metabolitos principales que vienen del tracto digestivo?
2. ¿Cuáles son los tejidos principales involucrados en la lipogénesis?
3. ¿Cuáles son los problemas fundamentales de la nutrición animal en países económicamente subdesarrollados?
4. ¿Cuál es la cantidad mínima de proteína cruda para un aumento de 1 kg diario de peso vivo?
5. ¿A qué se debieron los niveles más elevados de amoníaco en dietas con concentrados?
6. ¿Cuál fue el efecto más importante en el control del Rhopalosiphum?
7. ¿Cómo son los estimados de heredabilidad referentes al peso final?
8. ¿Cuál sería un indicador económico de la calidad de la carcasa?
9. ¿Qué forma tomó la curva de rendimiento contra el peso vivo vacío en carneros?
10. ¿Quién mostró que la curva de rendimiento contra el peso vivo vacío en carneros es una hipérbola?
11. ¿En qué año Tulloh mostró que la curva de rendimiento contra el peso vivo vacío en carneros es una hipérbola?
12. ¿Como qué el girasol es uno de los cultivos más prometedores en las condiciones tropicales de Cuba?
13. ¿Qué produjo el sorgo?
14. ¿Dónde fueron introducidas primeramente las heces de ganado?
15. ¿Cuáles fueron las plagas de artrópodas de principal importancia?
16. ¿A partir de qué edad se cebaron toros Brahman, Charolais, Criollos y Santa Gertrudis?
17. ¿Qué debe comprender un agricultor para que tenga buen éxito?
18. ¿En qué la genética es la ciencia más importante?
19. ¿A qué se debe el valor que tienen los rumiantes para la raza humana?
20. ¿En cuántas categorías se puede dividir la producción de carne?
21. ¿En qué porciento fue inferior la producción de leche con miel que con maíz?
22. ¿Cuántos toros Holstein con cánulas colocadas en el rúmen se alimentaron en jaulas individuales?
23. ¿En qué juega un papel importante el estado de madurez de los forrajes?
24. ¿Qué condiciones climáticas en Cuba hacen imposible que el grano de sorgo tenga menos de un 20 % de humedad en el momento de cosecharse?
25. ¿Qué disminuyó significativamente al aumentar el número de aves por jaula?

26. ¿En qué año Hopkins sugirió que el valor nutritivo en las proteínas radicaba enteramente en los aminoácidos que producían su hidrólisis?
27. ¿Quién sugirió en 1916 que el valor nutritivo en las proteínas radicaba enteramente en los aminoácidos que producían su hidrólisis?
28. ¿Qué se ha colocado a lo largo del tracto digestivo de los rumiantes para estudiar el curso de su digestión?
29. ¿Cuántas hembras fueron utilizadas para calcular la heredabilidad del peso vivo en broilers Barred Plymouth Rock?
30. ¿Cuántos machos fueron utilizados para calcular la heredabilidad del peso vivo en broilers Barred Plymouth Rock?
31. ¿Cuál fue el método de correlación que se utilizó para determinar el coeficiente de heredabilidad para cada sexo?
32. ¿cómo se determinó el coeficiente de heredabilidad?
33. ¿cuántos toros Cebú se utilizaron para estudiar el efecto de cuatro niveles de forraje y de una inoculación ruminal?
34. ¿A base de qué estaban alimentados los 48 toros Cebú para estudiar cuatro niveles de forraje?
35. ¿Qué se reducía al aumentar la proporción de fibra en una dieta básica de miel/urea?
36. ¿qué se utilizó para determinar la razón proteica neta (RPN) de las heces de ganado?
37. ¿qué diseño se usó para determinar la RPN de las heces de ganado utilizando ratas blancas?
38. ¿Qué valor se fijó usando la razón proteica neta (RPN) de heces de ganado?
39. ¿De quién es el método de la razón proteica neta usado para fijar el valor biológico de la proteína?
40. ¿Dónde fueron introducidas las heces de ganado como suplemento vitamínico?
41. ¿cuál fue el factor limitante en la disminución del rendimiento de la leguminosa alfalfa?
42. ¿en qué estación en Cuba se desarrollaron las pruebas de campo para el control de las plagas?
43. ¿Qué es lo que han conservado las semillas de insecticidas de hidrocarburo clorado bajo una variedad de condiciones climáticas y de suelos?
44. ¿qué disminuyó significativamente al aumentar el número de aves por jaula?
45. ¿qué no se afectó al aumentar el número de aves por jaula?
46. ¿Dónde depositan los huevos las hembras del Trichogramma?
47. ¿Dónde se está usando ampliamente la inseminación artificial en rebaños de carne?
48. ¿A cuántos animales se le realizaron autopsias y examinaron macro y microscópicamente?
49. ¿Cuántos corazones de las víctimas de "muerte súbita"mostraron cambios degenerativos en el examen microscópico?
50. ¿Cuántos corazones de las víctimas de "muerte súbita"mostraron signos de pericarditis en el examen microscópico?
51. ¿Qué porciento de harina de pescado tenía cada fuente energética en la dieta?
52. ¿Qué se puede emplear para acortar el ciclo de vida de la alfalfa para semillas?
53. ¿En qué raza fue más alto el porcentaje de rendimiento?
54. ¿En qué porcentaje fueron incrementadas las ganancias de peso vivo por animal, por obrero y por tractor?
55. ¿En cuántas formas diferentes ha sido aplicado el sistema de cebar toros con miel/urea ad libitum y forraje desarrollado en el Instituto de Ciencia Animal?
56. ¿Cuál es la técnica más apropiada para el estudio de la digestión de los azúcares en el rumen?

57. ¿Por qué está determinada la calidad de la carne y grasa de cerdos según Nambela?
58. ¿Qué aparece en nuestros pastizales en determinadas épocas del año?
59. ¿Cuál es el factor fundamental que influye en el desarrollo ruminal anatómico y fisiológico?
60. ¿Quienes ejercen un efecto abrasivo en la pared del rumen?
61. ¿Por quienes fue demostrado el efecto positivo de la suplementación con harina de girasol sobre el consumo de forraje y las ganancias de peso vivo?
62. ¿Dónde ocurre la hidrólisis del ácido fítico?
63. ¿En qué porciento la disminución en la restricción alimenticia provoca graves trastornos en el fisiologismo del animal y en el peso del feto?
64. ¿A cuántos días ocurre la mayor tasa de crecimiento según Rathray, McKrown y Eley?
65. ¿Qué implican los esquemas conocidos a través de pruebas en jaulas de metabolismo o cámaras calorimétricas?
66. ¿Qué criterio se sugiere incorporar en los centros genéticos porcinos como indicador global de la eficiencia económica?
67. ¿Qué mide la productividad numérica?
68. ¿Desde qué punto de vista la productividad por tiempo de presencia es la más real?
69. ¿Quienes hallaron que los terneros que recibieron alimento seco tuvieron mayor peso del contenido del tracto gastrointestinal en 1973?
70. ¿Qué aplicación requiere que el ternero joven consuma temprano el alimento sólido?
71. ¿Qué influencia provoca pérdidas y gastos de energía como producción de calor?
72. ¿Qué medición constituye una premisa básica para estudios de requerimientos nutricionales?
73. ¿Quiénes estudiaron el efecto de la caseína infundida en el rumen en la excreción urinaria de alantoína?
74. ¿Cuándo Blaxter y Martin estudiaron el efecto de la caseína infundida en el rumen en la excreción urinaria de alantoína?
75. ¿Cuántas horas de ayuno nocturno mantuvieron a las aves para determinar la actividad enzimática de las proteasas?
76. ¿Qué valor puede variar en animales con distintas demandas energéticas y alimentados con similares raciones?
77. ¿Quién ha hecho posible la producción de un bioconcentrado de lisina a partir de la fermentación directa de las mieles finales de caña?
78. ¿Quién indicó que al disminuir el contenido de lignina aumentó la digestibilidad del pasto en 1978?
79. ¿Qué aumentó al disminuir el contenido de lignina?
80. ¿Sobre qué fue desarrollada la levadura torula en Cuba?
81. ¿qué incrementó el empleo de altos niveles de levadura torula en dietas para pollos de ceba?
82. ¿En cuáles zonas no existen diferencias en la producción animal a base de asociaciones gramíneas/leguminosas?
83. ¿Quién plantea que no existen diferencias en la producción animal a base de asociaciones gramíneas/leguminosas entre las zonas templadas y los trópicos?
84. ¿Dónde ha tenido éxito la respuesta a inyecciones intramusculares de insulina?
85. ¿Con qué han sido alimentados los cerdos que han tenido éxito en la respuesta a inyecciones intramusculares de insulina?
86. ¿qué indican los resultados que se obtienen en el área tropical?
87. ¿a qué correspondió la caída ocurrida en el primer trimestre de 1981?
88. ¿qué provocó la caída ocurrida en el primer trimestre de 1981?

89. ¿cómo fue el llenado de los silos?
90. ¿qué son los silos de laboratorio?
91. ¿Por qué vía se han desarrollado tecnologías en Cuba para el secado de las mieles finales de caña?
92. ¿Por qué vía se han desarrollado tecnologías en nuestro país para el secado de las mieles finales de caña?
93. ¿A qué se le denomina miel final deshidratada?
94. ¿qué es MFD?
95. ¿En qué podría usarse la miel final deshidratada?
96. ¿Por qué no fue afectado el crecimiento corporal de las novillas?
97. ¿Por qué técnica fue estimada la digestibilidad in vitro?
98. ¿Qué se ha usado para alimentar a terneros en Cuba en los últimos años?
99. ¿En dónde tiene ventajas el uso de leche fermentada en la alimentación de terneros en Cuba?
100. ¿Cómo se realizó el análisis químico del heno y del concentrado?
101. ¿Cuándo se realizó el análisis químico del heno y del concentrado?
102. ¿Por qué método se efectuó la alantoína en 1942?
103. ¿Cómo se realizó la digestibilidad de la materia orgánica?
104. ¿Qué pueden provocar los piensos con alto contenido de ácido linoleico?
105. ¿Qué pueden provocar los metabolitos tóxicos según Budowski?
106. ¿Qué provoca la harina de trigo al ponerse en contacto con los líquidos?
107. ¿Qué origina el empastamiento del pico y la mucosa bucal?
108. ¿En qué proceso se le incorporan cantidades considerables de harina de trigo al afrechillo de trigo?
109. Diga alguna propiedad de la harina de trigo.
110. ¿Qué es EN?
111. ¿Qué es PC?
112. ¿Qué complica los resultados de los experimentos para explorar las necesidades minerales de los cerdos?
113. ¿En qué se basan las normas de Ca y P para cerdos en crecimiento del NRC?
114. ¿En qué son útiles las técnicas electroforéticas?
115. ¿Cuál gen tuvo un efecto favorable en la respuesta inmune a la cepa LA SOTA del virus de Newcastle?
116. ¿Para qué se utilizó la dócima de Duncan?
117. ¿Con qué se relaciona el peso de los huevos?
118. ¿Qué aves presentan un mayor grosor de la cáscara?
119. ¿Por quienes ha sido estudiada la cáscara de las aves cuello desnudo Nana?
120. ¿Qué ha ocurrido con el uso de la zeolita en la alimentación animal?
121. ¿Qué puede ocasionar la persistencia de los piretroides en los cultivos?
122. ¿Desde qué año se estudian en Cuba las aves cuello desnudo para el engorde?
123. ¿Qué porciento de materia seca produjo el sistema girasol-guinea?
124. ¿Cuándo fue mayor la prolificidad de cabras Nubia?
125. ¿Dónde fue menor el consumo de materia seca?
126. ¿En qué pudo influir el bajo contenido de MS de la Saccharína?
127. ¿Qué resulta de suministrar altas proporciones de leucaena en la dieta?
128. ¿Qué puede hacer la glicine?
129. ¿Qué habilidad tienen los microorganismos ruminales ?
130. ¿Qué no posee la proteína microbiana?
131. ¿De qué depende el crecimiento de los animales?
132. ¿Cómo fue el promedio de peso corporal de la gallina criolla cuello desnudo al poner el primer huevo?
133. ¿a qué se asocia el mejor estado de inmunocompetencia según Fraga?
134. ¿en qué son superiores las gallinas cuello desnudo de acuerdo con Nordskog?

135. ¿qué pudiera aumentar la supervivencia de las estacas en siembras en surcos?
136. ¿qué contribuyó al establecimiento de las estacas y a la producción de biomasa?
137. ¿quién halló una relación directa entre el diámetro de las etacas, el porcentaje de germinación y su capacidad de rebrote?
138. ¿Cuáles son las causas de mayores daños económicos en los rebaños lecheros?
139. ¿Cuáles son las pérdidas más costosas en materia de rebaños lecheros?
140. ¿Qué porciento de terneros Holstein mueren cada año en Estados Unidos?
141. ¿Qué manifiesta alta variabilidad genético aditivo?
142. ¿Por qué las varianzas genéticas aditivas asociadas a las pérdidas totales fueron superiores?
143. ¿Para quienes fueron altos los coeficientes de variación genética?
144. ¿Por qué los sementales se hallan evaluados para pérdidas totales?
145. ¿Cuáles son los sementales que ocasionan más del 10 % de las pérdidas?
146. ¿A qué provocan daños sensibles los abortos y las crías muertas?
147. ¿Cómo es posible elevar la producción de leche?
148. ¿Qué medida de interés no se ha estudiado ampliamente en el búfalo de río?
149. ¿Cuál es la raza deseada para generalizar por absorción en las condiciones de Cuba?
150. ¿Cuál es la media general del peso al nacimiento de los bucerros?
151. ¿En qué porciento son las pérdidas debidas a la cosecha mecanizada?
152. ¿Qué patrón siguieron los primeros síntomas de toxicidad de miel?
153. ¿Para qué se inserta plástico en el rumen?
154. ¿Qué es la necrosis cerebrocortical?
155. ¿En qué proporciones están la glucosa, fructosa y sacarosa en las mieles?
156. ¿Cómo es la conversión de alimentos en la ceba de cerdos con dietas basadas en miel?
157. ¿Qué produce la cytophaga johnsonii?
158. ¿Qué densidad tienen los pellets?
159. ¿A qué conducieron los experimentos de Preston en 1958?
160. ¿Qué publicaciones existen sobre el desarrollo de la fermentación ruminal en terneros alimentados con concentrados?
161. ¿Qué se busca con un sistema de esponjas intravaginales de polyuretano?
162. ¿Quiénes han estudiado la miel como fuente energética en terneros destetados?
163. ¿Qué afecta el desarrollo anatómico del rumen?
164. ¿Qué se ha señalado sobre el ambiente interior del rumen?
165. ¿Qué son los tratamientos diéticos?
166. ¿Qué resultado se obtiene al aumentar la concentración de urea en la miel en la dieta del ganado?
167. ¿Con qué aumentó el crecimiento de ratas albinas alimentadas con dietas que contenían solamente amino ácidos esenciales?
168. ¿Qué disminuyó el nivel más alto de metionina en las ponedoras?
169. ¿Qué produce la sustitución de los granos por azúcar y suplementos proteicos?
170. ¿Qué mostró el epitelio ruminal de todos los animales que recibieron miel?
171. ¿En qué ambiente se sugiere la selección de animales bajo condiciones tropicales?
172. ¿Qué producen las inoculaciones bacterianas en leguminosas de regiones tropicales y templadas?
173. ¿Cuáles fueron los tratamientos para romper la dormancia?
174. ¿Con cuántas horas la inoculación mostró una tendecia favorable no significativa?
175. ¿Qué provocan las cubiertas duras e impermeables de las semillas de hierba de guinea común?
176. ¿Cómo se fabrica la miel rica?

177. ¿Desde cuando se conoce el rápido metabolismo de los carbohidratos solubles en el rumen?
178. ¿Qué introdujeron para elevar la producción de ácido lático en ovejas?
179. ¿Qué produce el Streptococcus agalactiae?
180. ¿En qué términos es costoso secar granos artificialmente?
181. ¿Qué es la Pangola?
182. ¿Qué es el paraná?
183. ¿Qué es la faragua?
184. ¿Qué provoca el streptococcus en las vacas?
185. ¿Qué valor aumentan los insecticidas organofosfóricos sistémicos?
186. ¿Cuáles son los insecticidas organofosfóricos sistémicos?
187. ¿Qué levadura se hace de la fermentación directa de la miel final?
188. ¿Dónde se usa la atrazina?
189. ¿Cuál es el factor limitante en la utilización de urea?
190. ¿A quién afecta la toxicidad de la miel?
191. ¿Qué es la toxicidad de la miel?
192. ¿A qué se debe la poliencefalomalasia?
193. ¿Qué es NCC?
194. ¿Qué es la canulación?
195. ¿Qué son la glucosa, fructosa y sacarosa?
196. ¿Para qué fue empleada la técnica de la cánula re-entrante en el íleo?
197. ¿Qué papel juega la sacarasa en la digestión de las mieles?
198. ¿Qué demostró Sugimoto sobre la célula de Saccharomyces cerevhiaa?
199. ¿Cuál es la principal población de ganado en Cuba?
200. ¿A qué conduce la sustitución de cereales por azúcar en dietas para pollos de ceba?
201. ¿Cuál es el motivo de la tasa alta de mortalidad en los patos?
202. ¿En qué se basaron los sistemas de cría de ganado de carne y puercos en crecimiento en Cuba?
203. ¿En qué consiste el síndrome conocido como toxicidad o borrachera de miel en ganado de carne?
204. ¿Qué es palatabilidad en la nutrición?
205. ¿Cuál es el sistema actual para la cría de conejos en Cuba?
206. ¿Quién sugirió por primera vez usar carbohidratos solubles de la caña de azúcar como base para la producción animal intensiva en los trópicos?
207. ¿Cuando se sugirió por primera vez usar carbohidratos solubles de la caña de azúcar como base para la producción animal intensiva en los trópicos?
208. ¿Cuáles son las gramíneas que sustentan las mayor parte de la ganadería en Cuba?
209. ¿Qué es el guarapo?
210. ¿Qué es el AGV?
211. ¿Qué es el REP?
212. ¿Por qué se produce la mastitis en vacas de leche?
213. ¿Qué es la atrazina?
214. ¿Qué es Sitophilus teamais?
215. ¿Qué valor nutritivo puede tener el grano de sorgo cosechado temprano con alta humedad?
216. ¿Cómo es el clima en Arizona?
217. ¿Qué enfermedad provocan las leguminosas?
218. ¿En qué animales se presenta con frecuencia el timpanismo?
219. ¿Qué gusanos se responsabilizan por la creación de condiciones adecuadas de aereación del suelo?

220. ¿Qué animales nocivos pulverizan el material orgánico haciéndolo susceptible a la actividad microbiana?
221. ¿Cuál es la alimentación de los anécicos?
222. ¿Qué son las lombrices de tierra?
223. ¿Qué insectos son eficientes en la aceleración del movimiento del suelo?
224. ¿Qué insecto muestra tolerancia a los insecticidas?
225. ¿Cuál es la primera vacuna natural que recibe el ternero recién nacido?
226. Buscar el nombre de la primera vacuna no creada por el hombre que toma el ternero cuando acaba de nacer
227. ¿Qué tipo de terneras se utilizaron para determinar el efecto de la forma física del heno en el comportamiento de las terneras de reposición?
228. Devolver la clase de terneras que se usaron para determinar el efecto de la forma física del heno en el comportamiento de las terneras de reposición
229. ¿Cuál es el acrónimo del "intervalo parto-primera inseminación¿
230. ¿Cuál es la abreviatura de la "ganancia media diaria ¿
231. Nombra un sistema de crianza del ternero
232. ¿Qué sistemas de crianza del ternero hay?
233. ¿Qué es el Pennisetum purpureum?
234. ¿Qué tipo de sustancia es el Pennisetum purpureum?
235. ¿Qué carneros se usaron para evaluar los efectos de agregar cantidades crecientes de bicarbonato de sodio al concentrado?
236. ¿Qué clase de carneros se emplearon para valorar los efectos de agregar cantidades crecientes de bicarbonato de sodio al concentrado?
237. ¿Cuántos carneros se emplearon para valorar los efectos de agregar cantidades crecientes de bicarbonato de sodio al concentrado?
238. ¿Qué sustancia puede actuar como tampón de los ácidos orgánicos que se producen en el rumen a partir de alimentos que fermentan rápidamente?
239. ¿Cuál es la Directiva del Consejo Europeo para la protección de los animales usados para propósitos experimentales?
240. Buscar la Directiva del Consejo Europeo para la protección de los animales usados para propósitos científicos
241. ¿Cuántos habitantes habrán para el 2030 según informes de la FAO?
242. ¿Cuántas personas habrán en la población mundial para el 2030 según informes de la Organización para la Agricultura y Alimentación de las Naciones Unidas?
243. ¿Qué significa GEI?
244. ¿Qué porcentaje de la producción de metano generado por animales domésticos es producido por rumiantes?
245. ¿Qué producto energético-proteico se creó a partir de la caña de azúcar?
246. ¿Qué es la Saccharina?
247. ¿A qué grupo pertenecen la mayoría de las arcillas y zeolitas?
248. ¿De qué están compuestas la mayoría de las arcillas y zeolitas?
249. ¿Cómo se les conoce también a las arcillas?
250. ¿Cuál es el principal constituyente del filosilicato bentonita?
251. ¿Qué sustancia reduce la absorción de radionucleótidos del alimento?
252. ¿Qué sustancia es efectiva como portadora de la liberación lenta de medicamentos?
253. ¿Qué sustancia contiene el ATN?
254. ¿Cuál es el principal constituyente del ATN que tiene una alta selectividad por cationes de potasio y una relativamente baja selectividad por los cationes de calcio y magnesio?
255. ¿Qué especie se usa en Cuba para la producción de forrajes?
256. ¿Qué planta se emplea en Cuba para la producción de forrajes?
257. ¿Cuál es el pasto que más se propaga en Cuba?

258. Nombra un tipo de pasto cubano
259. ¿Qué planta herbácea de la familia de las compuestas Asteraceac es originaria de Centro América?
260. Buscar plantas herbáceas de la familia de las compuestas Asteraceac y originaria de Centro América
261. ¿Qué planta tiene el nombre vulgar de margaritona?
262. ¿Qué planta tiene el nombre vulgar de margarita gigante?
263. ¿A qué altitud se encuentra la Tithonia diversifolia?
264. ¿Qué planta se conoce como "botón de oro¿
265. ¿Qué planta se usa como cerca viva, como flora para apicultura, como medicina, en silvopastoreo bovino y como forraje de corte en la alimentación de cerdos, ovejas, conejos, bovinos y búfalos?
266. ¿Qué género de búfalo de agua tiene habilidades para adaptarse al calor y a las áreas húmedas?
267. ¿Qué búfalo de agua puede adaptarse al calor y a las áreas húmedas?
268. ¿En qué fecha se introdujo el búfalo de agua en Cuba?
269. ¿Qué sustancia producen los rumiantes en el rumen?
270. ¿Qué hidrocarburo producen los rumiantes en el rumen?
271. ¿Qué es la metanogénesis?
272. ¿A qué hidrocarburo se reduce el metilo en la metanogénesis?
273. ¿Qué tipo de ácido graso es el acético?
274. ¿Qué mineral necesita de enzimas para aumentar su digestibilidad por parte de los animales?
275. ¿Qué enzima aumenta aumenta la digestibilidad del fósforo orgánico por parte de los animales?
276. ¿Cuál es el animal doméstico menos estudiado del mundo?
277. ¿Qué animal consume pasto y rumia en el período diurno?
278. ¿Qué animales se dedican en el período diurno al consumo de pasto y rumia?
279. ¿Qué plantas usan los criadores en el aprovechamiento de becerros provenientes de hatos lecheros?
280. ¿Qué compuesto químico hace un medio más alcalino?
281. ¿Qué producto de origen vegetal posee especies en su microbiota que hidrolizan la urea?
282. ¿Qué ácido graso tiene un efecto inhibitorio en el crecimiento microbiano?
283. ¿Qué ácido orgánico produce el catabolismo de los carbohidratos?
284. Buscar el ácido orgánico que produce el catabolismo de los carbohidratos
285. ¿Qué cereal influye positivamente en el control de las malezas?
286. Nombrar un cereal que actue de forma positiva en el control de malezas
287. ¿Qué cereal aporta mayor cantidad de biomasa a los sistemas de intercalamiento?
288. Buscar el cereal que mayor aporte de biomasa realiza a los sistemas de intercalamiento
289. ¿Qué es el nitrógeno?
290. ¿En qué estación tuvo un comportamiento atípico el tratamiento con nitrógeno?
291. Buscar la estación donde el tratamiento con nitrógeno mostró un comportamiento atípico
292. ¿Qué elemento químico aumentó la duración del área foliar?
293. Buscar el elemento químico que con su adición aumentó la duración del área foliar
294. ¿Qué suelo ejerce quimio atracción hacia determinados grupos microbianos?
295. ¿Qué virus pueden afectar la actividad microbiana en algunos suelos?
296. Nombrar los microorganismos que pueden haber en algunos suelos afectando la actividad microbiana

297. ¿Qué tipo de rhizobiaceae es sensible al ataque de bacteriófagos?
298. ¿En qué microhabitat se ha encontrado la presencia de microorganismos nitrofijadores?
299. ¿Qué producto de origen vegetal tienen la capacidad para nodular en los tallos?
300. ¿Qué mineral es usado en las dietas para incrementar los componentes de la leche?
301. ¿Qué glucosidos son relacionados con la baja aceptabilidad y consumo de algunos forrajes de árboles?
302. ¿Cuál es la leguminosa que presenta un alto rendimiento de materia seca?
303. ¿Qué planta tiene mayor valor nutritivo?
304. ¿Qué glicosidos tienen un efecto defaunante en el rumen?
305. ¿Qué papilionoideae se degrada mucho más en el rumen que otros árboles?
306. ¿Qué tipo de fermentación ha sido utilizada para el reciclaje de materiales voluminosos?
307. ¿Qué contenido de nitrógeno necesita incorporar el proceso de síntesis proteica para la formación de los grupos aminos?
308. ¿Qué producto de origen vegetal es el mayor contribuidor de energía en las dietas de monogástricos?
309. ¿Qué caesalpinioideae no tiene influencia en las poblaciones de bacterias viables totales y hongos elulolíticos ruminales?
310. ¿Qué es la spirulina?
311. ¿Cuál es la mimosoideae más empleada en Cuba?
312. ¿Qué compuesto orgánico elaborado en las hojas de las plantas perennes interviene en el enraizamiento de los árboles?
313. Buscar los compuestos orgánicos que intervienen en el enraizamiento de los árboles
314. ¿Qué es la disposición espacial?
315. ¿Qué psyllidae permanece en el cultivo de leucaena?
316. ¿Qué rodentia tiene los mejores indicadores de crecimiento posdestete?
317. ¿Qué tipo de ganado se adapta fácilmente a las condiciones ambientales en las diferentes latitudes?
318. ¿Qué es el rumen?
319. ¿Qué mimosoideae incrementa la población microbiana y el número de organismos celulolíticos?
320. ¿Qué leguminosa incrementa la población microbiana total así como el número de organismos celulolíticos?
321. ¿Qué cereal es intercalado como planta protectora de otros cultivos?
322. Buscar el producto de origen vegetal que es intercalado como planta protectora de otros cultivos
323. ¿Qué esteroide modulado en el cerebro por los receptores de vasopresina favorece la agresión?
324. ¿Qué endomycetales es un producto del proceso de producción de alcohol y constituye una fuente de proteínas?
325. ¿Qué hongo se obtiene del proceso de producción de alcohol?
326. ¿Qué compuesto orgánico se utiliza para incrementar la densidad energética de la dieta de vacas lecheras?
327. ¿Qué ácidos orgánicos son utilizados para incrementar la grasa láctea?
328. Nombrar el compuesto químico que incrementa la grasa láctea
329. ¿Cuáles son los compuestos orgánicos que pueden modificar la intensidad y la orientación de las fermentaciones ruminales?
330. ¿Qué productos vegetales procesados se pueden suministrar para disminuir la grasa de la leche?

Apéndice B

Acrónimos

B.1. Glosario de acrónimos usados en la memoria

Acrónimo	Descripción
AliQAn	Alicante *Question Answering*
ASR	*Automatic Speech Recognition*
BR	Búsqueda de Respuestas
BR-DA	Búsqueda de Respuestas de Dominio Abierto
BR-DR	Búsqueda de Respuestas de Dominio Restringido
BS	Bloques Sintácticos
CLEF	*Cross-Language Evaluation Forum*
DLSI	Departamento de Lenguajes y Sistemas Informáticos
EWN	EuroWordNet
GPLSI	Grupo de investigación en Procesamiento del Lenguaje y Sistemas de Información
IA	Inteligencia Artificial
ILNBD	Interfaces en Lenguaje Natural de acceso a Bases de Datos
LC	Lingüística Computacional
MISC	Miscelánea
MT	*Machine Translation*
NTCIR	*NII Test Collection for IR Systems*
OCR	*Optical Character Recognition*
ORG	Organización
PLN	Procesamiento del Lenguaje Natural
RI	Recuperación de Información
SBR-DA	Sistemas de Búsqueda de Respuesta en Dominios Abiertos
SBR-DR	Sistemas de Búsqueda de Respuesta en Dominios Restringidos
SN	Sintagma Nominal
SOC	Sistemas de Organización del Conocimiento
SP	Sintagma Preposicional
SQL	*Structured Query Language*
SRI-DR	*Sistemas de Recuperación de Información en Dominios Restringidos*
SV	Sintagma Verbal
TEXT-MESS	Minería de Textos Inteligente, Interactiva y Multilingüe basada en Tecnología del Lenguaje Humano
TRE	Tipo de Respuesta Esperada
TREC	*Text Retrieval Conference*
UA	Universidad de Alicante
URL	*Uniform Resource Locator*
W3C	*World Wide Web Consortium*

Cuadro B.1. Acrónimos usados en la memoria

Apéndice C

Taxonomía de TRE para el dominio agrícola

C.1. Taxonomía de tipo de respuestas esperadas generada por nuestra propuesta para el dominio agrícola

```
-------------------------------------------
Taxonomía de tipo de respuesta esperada
a partir del modelo de dominio restringido.
-------------------------------------------

0 acuerdos

0 agentes
1 ---catalizador
2 ------enzimas
3 ---------hidrolasas
4 ------------proteasas
4 ------------glicosidasas
4 ------------esterasas

0 agricultura
1 ---practicas_agricolas
2 ------ganaderia
3 ---------metodos_de_crianza
3 ---------apicultura
4 ------------manejo_del_apiario

0 agua

0 alimentos
1 ---bebidas
2 ------jugo_de_frutas

0 almacenamiento

0 alojamiento_de_animales

0 biota
1 ---fauna
2 ------cordado
3 ---------vertebrados
4 ------------pajaros
5 ---------------coraciiformes
5 ---------------galliformes
6 ------------------pollo
3 ---------vertebrado
4 ------------mamifero
5 ---------------rumiante
6 ------------------cervidae
6 ------------------bovidos
7 ----------------------caprinae
8 ------------------------caprinos
8 ------------------------ovinos
7 ----------------------bovina
8 ------------------------ganado_bovino
9 --------------------------vaca
6 ------------------venado
6 ------------------bovido
7 ----------------------ovinos
5 ---------------lagomorfos
5 ---------------rodentia

0 colonia_de_abejas

0 compuesto_organico_halogeno
1 ---compuesto_organico_del_cloro

0 cosecha

0 costos

0 daños

0 derecho
1 ---derechos_humanos
2 ------derechos_de_propiedad
3 ---------propiedad
1 ---jurisprudencia

0 desechos
1 ---desechos_agricolas
2 ------residuos_de_cosechas
3 ---------paja

0 deterioro

0 empresas

0 enfermedad
1 ---enfermedades_organicas
2 ------enfermedades_de_la_piel

0 equipo
1 ---recipientes
1 ---maquinaria_de_labranza
```

```
2 ------arados
2 ------cultivadores
1 ---maquinaria_de_manutencion
1 ---cosechadoras

0 estructura_agricola
1 ---estructura_de_la_explotacion

0 extractos

0 fibras
1 ---fibras_vegetales
2 ------fibras_blandas

0 fuente_de_energia

0 grasas

0 grupos

0 indicadores

0 industria

0 leche

0 limpieza

0 mantenimiento

0 materiales

1 ---materiales_de_construccion

0 medicina

0 mercados

0 metodos

0 modelos

0 organismos
1 ---animales
2 ------animales_jovenes
2 ------suidae
3 ---------cerdo
2 ------animales_utiles
3 ---------animales_domesticos
4 ------------ganado
5 ---------------cerdo
5 ---------------caprinos
5 ---------------ovinos
5 ---------------ganado_bovino
6 ------------------vaca
5 ---------------aves_de_corral
6 ------------------pollo
2 ------mamifero
3 ---------rumiante
4 ------------cervidae
4 ------------bovidos
5 ---------------caprinae
6 ------------------caprinos
6 ------------------ovinos
5 ---------------bovina
6 ------------------ganado_bovino
7 --------------------vaca
4 ------------venado
4 ------------bovido
5 ---------------ovinos
3 ---------lagomorfos
3 ---------rodentia
2 ------insecta
3 ---------hymenoptera
4 ------------aphelinidae
3 ---------coleoptera
4 ------------scolytidae
5 ---------------ips
2 ------chordata
3 ---------vertebrados
4 ------------pajaros
5 ---------------coraciiformes
5 ---------------galliformes
6 ------------------pollo
1 ---microorganismos
2 ------bacteria
3 ---------bacillaceae
4 ------------bacillus
3 ---------rhizobiaceae
4 ------------rhizobium
1 ---plagas
1 ---plantas
2 ------poaceae
3 ---------avena
3 ---------brachiaria
3 ---------sorghum
2 ------leguminosae
3 ---------mimosoideae
3 ---------papilionoideae
4 ------------arachis
4 ------------canavalia
4 ------------centrosema
4 ------------desmodium
4 ------------medicago
4 ------------mucuna
4 ------------vigna
2 ------rutaceae
2 ------caryophyllaceae
2 ------cyanobacteria
2 ------pteridophyta
3 ---------helechos

0 pesca

0 piensos

0 precios

0 productividad

0 productos
1 ---subproductos
2 ------subproductos_de_cereales
3 ---------subproductos_de_la_molineria
1 ---productos_de_origen_animal
2 ------carne
3 ---------piezas_de_carne
2 ------canal_animal
2 ------productos_de_la_colmena

0 progenie

0 rendimiento

0 residuos
1 ---contaminantes

0 roca

0 seguridad

0 servicios

0 sexo

0 tejido
1 ---membrana

0 tierra

0 trabajo
```

C.1 Taxonomía de tipo de respuestas esperadas generada por nuestra propuesta para el dominio agrícola

```
0 transporte

0 trastornos
1 ---trastornos_metabolicos
1 ---trastornos_funcionales
2 ------trastornos_digestivos
2 ------trastornos_de_la_reproduccion
3 ---------complicaciones_del_embarazo

0 usos

0 reproduccion_dirigida

0 propiedades_fisicoquimicas
1 ---propiedades_opticas
1 ---propiedades_mecanicas
1 ---propiedades_termicas
2 ------temperatura
1 ---presion

0 propiedades_biologicas
1 ---tolerancia

0 fenomenos_biologicos
1 ---ritmos_biologicos

0 factores_inmunologicos

0 operaciones_forestales
1 ---aprovechamiento_de_la_madera

0 servicios_sociales

0 construcciones_hidraulicas

0 finanza

0 manejo_del_ganado
1 ---eliminacion
2 ------vaciado
3 ---------drenaje

0 instituciones
1 ---centros
1 ---instituciones_financieras
2 ------bancos
1 ---instituciones_de_educacion

0 ocupaciones
1 ---cientificos
1 ---obreros

0 servicios_de_alimentacion

0 biologia
1 ---comportamiento
2 ------aprendizaje
2 ------comportamiento_humano
3 ---------comportamiento_economico
4 ------------consumo
2 ------habitos_alimentarios
1 ---ecologia
2 ------ecosistema
1 ---taxonomia
2 ------taxa
3 ---------especies
3 ---------variedades
1 ---citologia
2 ------estructura_celular
3 ---------nucleo
4 ------------cromosomas
5 ---------------genes

0 polucion
1 ---contaminacion

0 composicion_aproximada

0 metodos_de_control

0 aspectos_fisiograficos

0 parametros_geneticos

0 desarrollo_biologico
1 ---crecimiento

0 metodos_de_mejoramiento_genetico
1 ---cruzamiento
1 ---selección

0 oferta_y_demanda
1 ---demanda

0 desempeño_animal

0 ordenacion_de_recursos

0 alimentacion
1 ---alimentacion_de_los_animales

0 manejo_del_cultivo

0 metodologia
1 ---ensayo

0 educacion
1 ---enseñanza

0 sistema_de_organizacion_del_conocimiento
1 ---esquema_de_relacion
2 ------taxonomia_(gestion_de_la_informacion)
2 ------taxonomia
3 ---------taxa
4 ------------especies
4 ------------variedades

0 estaciones_del_año

0 productos_de_origen_vegetal
1 ---estimulantes
1 ---frutas
1 ---pasta
1 ---cereales
1 ---frutos_secos
1 ---hortalizas
1 ---seudocereales

0 patogenesis

0 eleccion_de_la_epoca

0 organizaciones
1 ---fondo
1 ---organizaciones_internacionales

0 habitos_de_crecimiento

0 compuestos_inorganicos

0 herencia_genetica

0 dendrometria

0 metodos_de_aplicacion

0 factores_climaticos
1 ---condiciones_atmosfericas
2 ------precipitacion_atmosferica

0 productos_forestales
1 ---madera
```

```
2 ------madera_en_rollo

0 etapas_de_desarrollo
1 ---etapas_de_desarrollo_de_la_planta
1 ---etapas_del_desarrollo_animal

0 plantas_nocivas
1 ---malezas

0 enfermedades_de_las_plantas

0 estructuras_administrativas
1 ---oficina
2 ------agenciadelaonu

0 fabricas

0 fisiologia
1 ---metabolismo
2 ------via_bioquimica_del_metabolismo
1 ---reproduccion
2 ------reproduccion_sexual
3 ---------parto

0 compuestos_bioquimicos
1 ---vitaminas
2 ------vitaminas_b

0 sistemas_de_cultivo

0 estadisticas_vitales

0 funcion_fisiologica
1 ---movimiento
1 ---sentidos
1 ---fisiologia_de_la_nutricion
1 ---circulacion_sanguinea
1 ---secrecion
2 ------hormonas
3 ---------hormonas_sexuales
3 ---------esteroides
4 ------------esteroles

0 cercado

0 propiedades_plaguicidas

0 productos_pesqueros

0 compuestos_heterociclicos
1 ---purinas
1 ---triacina
1 ---azoles
2 ------imidazoles

0 factores_ambientales

0 necesidades_fisiologicas

0 inmunidad

0 renta

0 instalaciones_de_almacenamiento
1 ---silos

0 cobertura_de_suelos
1 ---suelo
1 ---vegetacion
2 ------bosques
2 ------praderas

0 nutricion

0 elementos_constitutivos

0 gestion
1 ---organizacion_del_trabajo

0 sistemas_silviculturales

0 amidas

0 anatomia_de_la_planta
1 ---organos_vegetativos_de_las_plantas
2 ------hojas
2 ------tallos
3 ---------tronco
2 ------raices
1 ---tejidos_vegetales
1 ---organos_reproductores_vegetales
2 ------inflorescencias
3 ---------flores

0 minerales

0 zonas_climaticas

0 ciencia
1 ---ciencia_de_informacion
2 ------clasificacion
3 ---------taxonomia
4 ------------taxa
5 ---------------especies
5 ---------------variedades
1 ---tecnologia
2 ------biotecnologia
1 ---ciencias_sociales
2 ------psicologia
3 ---------comportamiento
4 ------------aprendizaje
4 ------------comportamiento_humano
5 ---------------comportamiento_economico
6 -----------------consumo
4 ------------habitos_alimentarios
2 ------economia
3 ---------comercio
1 ---ingenieria
2 ------ingenieria_hidraulica
3 ---------drenaje
3 ---------riego
1 ---geografia
1 ---ciencias_de_la_tierra
2 ------geologia
3 ---------geofisica

0 medicion
1 ---dimension

0 partes_del_cuerpo
1 ---sistema_digestivo
2 ------estomago
1 ---regiones_del_cuerpo
1 ---tegumento
1 ---sistema_oseomuscular
2 ------huesos
1 ---organos_sensoriales
1 ---liquidos_corporales
2 ------sangre
3 ---------composicion_de_la_sangre
4 ------------proteinas_sanguineas
1 ---sistema_nervioso
2 ------sistema_nervioso_central
1 ---sistema_cardiovascular
1 ---glandulas_animales
2 ------glandulas_endocrinas
1 ---sistema_urogenital
2 ------genitalia
3 ---------aparato_femenino

0 produccion
1 ---produccion_animal
```

C.1 Taxonomía de tipo de respuestas esperadas generada por nuestra propuesta para el dominio agrícola

```
1 ---produccion_vegetal
2 ------cultivo
3 ---------labranza
3 ---------poda
3 ---------siembra
3 ---------apicultura
4 ------------manejo_del_apiario

0 americas
1 ---america_del_norte
2 ------eua
3 ---------estados_del_centro_norte_(eua)
4 ------------corn_belt_(eua)
1 ---america_del_sur
2 ------venezuela_(republica_bolivariana_de)

0 enmiendas_del_suelo
1 ---material_organico_de_cobertura

0 analisis
1 ---analisis_biologico

0 peces
1 ---peces_de_agua_dulce
1 ---peces_marinos

0 desarrollo_economico_y_social
1 ---desarrollo_economico

0 propagacion_de_plantas
1 ---propagacion_vegetativa

0 productos_procesados
1 ---productos_fermentados
2 ------leche_fermentada
1 ---productos_vegetales_procesados
2 ------gomas
2 ------productos_derivados_de_las_frutas
3 ---------jugo_de_frutas
2 ------fibras_vegetales
3 ---------fibras_blandas
1 ---productos_animales_procesados
2 ------productos_lacteos
3 ---------leche_fermentada

0 proteccion_legal
1 ---derechos_humanos
2 ------derechos_de_propiedad
3 ---------propiedad

0 europa
1 ---europa_occidental
2 ------francia

0 materiales_de_propagacion

0 formaciones_atmosfericas
1 ---fenomenos_atmosfericos

0 manejo_de_desechos

0 aromatizantes
1 ---condimentos
2 ------sal
3 ---------nitratos
3 ---------sulfatos
3 ---------haluro
4 ------------cloruros

0 tipos_de_suelos
1 ---tipos_geneticos_de_suelos

0 estructura_de_la_poblacion
1 ---estructura_por_edades
2 ------grupos_de_edad

0 politicas
1 ---politica_economica
2 ------politica_de_produccion
3 ---------ajuste_agrario
2 ------politica_de_comercio_exterior
3 ---------barreras_comerciales

0 compuestos_del_nitrogeno
1 ---compuesto_organico_del_nitrogeno
2 ------aminoacidos
3 ---------aminoacidos_sulfurados
2 ------compuestos_de_amino
3 ---------aminas
4 ------------aminas_biogenicas
2 ------peptidos
1 ---compuestos_de_amonio
2 ------compuestos_amonicos_cuaternarios

0 africa
1 ---africa_al_sur_del_sahara
2 ------africa_occidental
2 ------africa_oriental

0 reacciones_quimicas
1 ---degradacion
2 ------biodegradacion

0 ciencias_fisicas
1 ---fisica
2 ------radiacion
3 ---------luz
2 ------mecanica
1 ---quimica
2 ------quimicaorganica
3 ---------bioquimica
4 ------------metabolismo
5 ---------------via_bioquimica_del_metabolismo

0 asia_y_el_pacifico
1 ---oceania
2 ------australia

0 organovegetal
1 ---tallos
2 ------tronco

0 entidad
1 ---celulas
1 ---servivo
2 ------microorganismo
3 ---------bacteria
4 ------------bacillaceae
5 ---------------bacillus
4 ------------rhizobiaceae
5 ---------------rhizobium
2 ------planta
3 ---------hongos
4 ------------levadura
4 ------------deuteromycotina
4 ------------ascomycotina
5 ---------------endomycetales
6 ------------------saccharomyces
3 ---------plantavascular
4 ------------plantale
4 ------------pteridofita
5 ---------------helechos
4 ------------plantaherbacea
2 ------serhumano
3 ---------intelectual
4 ------------cientifico
3 ---------trabajador
4 ------------empleado
5 ---------------obreros
3 ---------malapersona
4 ------------infractor
5 ---------------impostor
```

```
1 ---objetoinanimado
2 ------sustancia
3 ---------levadura
3 ---------compuestos_quimicos
4 ------------acidos
5 ---------------acidos_organicos
6 ------------------aminoacidos
7 --------------------aminoacidos_sulfurados
6 ------------------acidos_fenolicos
6 ------------------acidos_grasos
4 ------------sales
5 ---------------sales_de_acidos_inorganicos
6 ------------------cloruros
6 ------------------nitratos
6 ------------------sulfatos
5 ---------------sales_de_acidos_organicos
4 ------------compuestos_organicos
5 ---------------alcaloides
5 ---------------alcoholes
6 ------------------polialcoholes
7 --------------------azucares_alcoholes
5 ---------------carbohidratos
6 ------------------azucares
7 --------------------azucares_reductores
8 ----------------------aldosas
6 ------------------azucares_alcoholes
6 ------------------polisacaridos
7 --------------------glucanos
6 ------------------oligosacaridos
6 ------------------glicosidos
6 ------------------monosacaridos
7 --------------------aldosas
5 ---------------lipidos
6 ------------------acilgliceroles
5 ---------------pigmentos
6 ------------------carotenoides
5 ---------------proteinas
6 ------------------enzimas
7 --------------------hidrolasas
8 ----------------------proteasas
8 ----------------------glicosidasas
8 ----------------------esterasas
6 ------------------globulinas
6 ------------------metalproteinas
5 ---------------cetonas
5 ---------------aldehidos
5 ---------------polimeros
5 ---------------compuesto_organico_del_nitrogeno
6 ------------------aminoacidos
7 --------------------aminoacidos_sulfurados
6 ------------------compuestos_de_amino
7 --------------------aminas
8 ----------------------aminas_biogenicas
6 ------------------peptidos
5 ---------------isoprenoides
6 ------------------terpenoidos
7 --------------------carotenoides
6 ------------------esteroides
7 --------------------esteroles
5 ---------------compuestos_aromaticos
6 ------------------compuestos_fenolicos
7 --------------------acidos_fenolicos
5 ---------------esteres
6 ------------------acilgliceroles
4 ------------aldehidos
4 ------------polimeros
4 ------------aminas
5 ---------------aminas_biogenicas
3 ---------elementos_quimicos
4 ------------no_metales
5 ---------------halogenos
4 ------------semimetales
4 ------------elementos_metalicos
5 ---------------metales_alcalinoterreos
5 ---------------metales_pesados
5 ---------------elementos_de_transicion
5 ---------------metales_alcalinos
4 ------------metal
3 ---------materia
4 ------------sustancias_quimicas
4 ------------productos_quimicos
5 ---------------agroquimicos
4 ------------colorante
5 ---------------pigmento
6 ------------------carotenoides
4 ------------piedra
4 ------------mineral
4 ------------fibra
5 ---------------fibranatural
3 ---------nutriente
4 ------------provisiones
4 ------------comida
5 ---------------productoagricola
6 ------------------frutas
6 ------------------verdura
4 ------------sustento
5 ---------------vitaminas
6 ------------------vitaminas_b
5 ---------------productos_lacteos
6 ------------------leche_fermentada
5 ---------------productolacteo
5 ---------------vitamina
6 ------------------vitaminaliposoluble
6 ------------------vitaminahidrosoluble
7 --------------------vitaminab
4 ------------bebida
3 ---------sustanciacorporal
4 ------------fluidocorporal
5 ---------------sangre
6 ------------------composicion_de_la_sangre
7 --------------------proteinas_sanguineas
2 ------artefacto
3 ---------via
3 ---------farmaco
4 ------------estimulantes
4 ------------medicamentos
5 ---------------vacuna
3 ---------estructura
4 ------------infraestructura
4 ------------inmueble
3 ---------creacion
4 ------------producto
5 ---------------subproductos
6 ------------------subproductos_de_cereales
7 --------------------subproductos_de_la_molineria
3 ---------utillaje
4 ------------mecanismo
4 ------------herramienta
5 ---------------utensilio
2 ------tierrafirme
3 ---------isla
2 ------porcion
3 ---------elemento
4 ------------suma
5 ---------------aditivos
1 ---celula
1 ---trozo
2 ------partedelcuerpo
3 ---------estructuraanatomica
4 ------------estructuraneurologica
5 ---------------sistema_nervioso_central
3 ---------organo
4 ------------organossexuales
4 ------------organosegregatorio
5 ---------------glandula
6 ------------------glandulas_endocrinas
6 ------------------glandulaendocrina

0 cuerpo
1 ---cromosomas
2 ------genes
```

C.1 Taxonomía de tipo de respuestas esperadas generada por nuestra propuesta para el dominio agrícola

```
0 rasgopsicologico
1 ---conocimiento
2 ------informacion
3 ---------datos
3 ---------fundamento
4 ------------sintoma
2 ------contenidomental
3 ---------areadeconocimiento
4 ------------area
5 ---------------disciplinacientifica
6 -----------------psicologia
7 ---------------------comportamiento
8 ------------------------aprendizaje
8 ------------------------comportamiento_humano
9 ---------------------------comportamiento_economico
10 ----------------------------consumo
8 ------------------------habitos_alimentarios
6 -----------------ciencias_sociales
7 ---------------------psicologia
8 ------------------------comportamiento
9 ---------------------------aprendizaje
9 ---------------------------comportamiento_humano
10 --------------------------comportamiento_economico
11 ---------------------------consumo
9 ---------------------------habitos_alimentarios
7 ---------------------economia
8 ------------------------comercio
6 -----------------cienciasnaturales
7 ---------------------fisica
8 ------------------------radiacion
9 ---------------------------luz
8 ------------------------mecanica
7 ---------------------ciencias_de_la_tierra
8 ------------------------geologia
9 ---------------------------geofisica
8 ------------------------geofisica
7 ---------------------cienciasdelatierra
8 ------------------------geologia
9 ---------------------------geofisica
8 ------------------------geografia
6 -----------------cienciassociales
5 ---------------areageografica
6 -----------------terreno

0 proteccion

0 globulina

0 teleosteo
1 ---malacopterigio

0 acto
1 ---hecho
1 ---actividad
1 ---acciondegrupo
2 ------mercantilizacion
3 ---------cambio
4 ------------cambiodeintegridad
3 ---------negocios
4 ------------finanzas
5 ---------------financiamiento

0 grupo
1 ---sistema
2 ------ecosistema
2 ------sistema_nervioso
3 ---------sistema_nervioso_central
2 ------estructura_social
3 ---------clases_sociales
2 ------sistemavascular
3 ---------sistema_cardiovascular

0 abstraccion
1 ---relacion
2 ------relacionsocial
3 ---------comunicación
4 ------------comunicacionoral
5 ---------------lenguaje
6 -----------------discusion
7 ---------------------tramitacion
8 ------------------------mercadeo
1 ---atributo
2 ------cualidad
3 ---------regularidad
4 ------------periodicidad
3 ---------morbosidad
4 ------------toxicidad
1 ---quantum
2 ------periodo
3 ---------estacion

0 fenomeno
1 ---proceso
2 ------procesamiento
3 ---------preservacion
3 ---------secado
3 ---------separacion
2 ------procesonatural
3 ---------procesobiologico
4 ------------cruzamiento
4 ------------metabolismo
5 ---------------via_bioquimica_del_metabolismo
3 ---------procesoquimico
1 ---naturaleza
2 ------fenomeno_fisico
3 ---------sorcion
2 ------fenomenofisico
3 ---------fenomenos_atmosfericos

0 posesion
1 ---bien
2 ------recurso
3 ---------recursos_naturales
4 ------------recursos_geneticos
4 ------------recursos_biologicos
5 ---------------recursos_geneticos
4 ------------recursos_no_renovables
5 ---------------recursos_minerales
5 ---------------recursos_de_la_tierra
6 -----------------tierras_de_pastoreo
7 ---------------------pastizales
6 -----------------tierras_agricolas
7 ---------------------tierras_cultivadas
8 ------------------------campo
9 ---------------------------praderas
3 ---------recursos_humanos
4 ------------mano_de_obra
5 ---------------obreros
```

Reunido el Tribunal que suscribe en el día de la fecha acordó otorgar, por a la Tesis Doctoral de Don/Dña. Katia Vila Rodríguez la calificación de .

Alicante de de

El Secretario,

El Presidente,

UNIVERSIDAD DE ALICANTE
CEDIP

La presente Tesis de D. _____ ha sido registrada con el nº _____ del registro de entrada correspondiente.

Alicante ___ de _____ de _____

El Encargado del Registro,