ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
Departamento de Tecnología Electrónica

Tesis doctoral

Método general de Extracción de Información basado en el uso de Lógica Borrosa. Aplicación en portales web.

Realizada por

Jorge Ropero Rodríguez.
Ingeniero de Telecomunicaciones

Dirigida por

Carlos León de Mora
Profesor Titular de Universidad

Sevilla, noviembre de 2009

Índice general.

Método general de Extracción de información basado en el uso de Lógica Borrosa. Aplicación en portales web. *1*

Índice general. *3*

Índice de figuras. *7*

Índice de tablas. *11*

Agradecimientos. *13*

Capítulo 1. Introducción. *15*

1.1. Planteamiento y objetivos. *15*

1.2. Estructura de la tesis. *16*

Capítulo 2. Recuperación y Extracción del conocimiento; Procesado del Lenguaje Natural. *19*

2.1. Minería de datos. *20*

2.1.1. Definición **20**

2.1.2. Minería de datos y estadística. **20**

2.1.3. Modelos de minería de datos. **21**

2.1.4. Relación de la minería de datos con otras disciplinas. **21**

2.1.5. Aplicaciones de la minería de datos. **23**

2.1.6. Características de la Minería de datos. **23**

Atributos. 23

Aprendizaje supervisado / no supervisado. 24

Algoritmos de aprendizaje. 24

2.2. Minería Web. *27*

2.2.1. Introducción a la Minería Web. **27**

2.2.2. Datos en minería web. **28**

2.2.3. La problemática de la Web 2.0. ..... **29**

Inteligencia colectiva. 30

Tipos de Inteligencia Colectiva. 30

Extracción de inteligencia a partir de etiquetas. 31

Tipos de contenido Web. 33

Arañas web (web crawlers) 34

2.2.4. Categorías de la minería web.....36
WCM (*Web Content Mining*). 36
WSM (*Web Structure Mining*).....37
WUM (*Web Usage Mining*).....37

2.2.5. Componentes de la minería web y metodologías.....38
Recuperación de Información (*Information Retrieval*, *IR*). 38
Extracción de Información – *Information Extraction*, IE -.....39
Generalización. 39
Análisis.....40

2.2.6. Limitaciones de los métodos existentes de minería web. 40
Recuperación de Información, IR. 40
Extracción de Información, IE. 41
Generalización. 41
Análisis.....42

2.3. Recuperación de Información (Information Retrieval, IR). 42

2.3.1. Introducción. 42

2.3.2. Modelo de Espacio Vectorial (Vector Space Model, VSM).....43
Indexado de documentos 43
Introducción de pesos para los términos.....44
Coeficientes de semejanza. 46

2.4. Extracción de Información (Information Extraction, IE). 47

2.4.1. Introducción. 47

2.4.2. Modelos básicos en IE. 48
Modelos para texto. 48
Modelos para hipertexto. 49
Modelos para datos semiestructurados. 49

2.5. Procesado del Lenguaje Natural.....50

2.5.1. Aproximación al Procesado del Lenguaje Natural.....50

2.5.2. Técnicas NLP.....51

2.5.3. Sintáctica y semántica. Webs Semánticas.....53

Capítulo 3. Herramientas basadas en Inteligencia Computacional para la búsqueda automática de conocimiento. 63

3.1. Inteligencia Computacional (Computational Intelligence, CI).....64

3.2. Herramientas de Inteligencia Computacional. 67

3.2.1. Lógica Borrosa (Fuzzy Logic, FL). 68

3.2.2. Conjuntos borrosos. 68

3.2.3. Inferencia borrosa. 72

3.2.4. Borrosificación (Fuzzyfication).....73

3.2.5. Desborrosificación (Defuzzyfication). 75

3.3. Aplicaciones de la AI a la búsqueda de conocimiento. 76

3.3.1. Aplicaciones de las ANN a la búsqueda de conocimiento.....78

3.3.2. Aplicaciones de la FL a la búsqueda de conocimiento. 79
FL basada en el VSM 81
FL basada en ontologías 87

Capítulo 4. Método general para la búsqueda de conocimiento basado en FL. 93

4.1. Agentes inteligentes para la búsqueda de conocimiento. 94

4.1.1. Agentes Inteligentes en investigación 95

4.1.2. Agentes Inteligentes comerciales en entornos web. 107

4.2. Caracterización del Agente Inteligente propuesto. 112

4.2.1. Objetivos del Agente Inteligente. 112

4.2.2. Estructura jerárquica. 114

4.2.3. Construcción del Agente Inteligente. 117
Representación en Lenguaje Natural (*Natural Language*, NL). 117
Elección de términos índice. 117
Asignación de pesos (Term Weighting). 118

4.2.4. Modo de operación del Agente Inteligente. 120

4.3. Sistema de Lógica Borrosa. 123

Capítulo 5. Aplicación: diseño de un Agente Inteligente para el portal web de la Universidad de Sevilla. 129

5.1. El portal web de la Universidad de Sevilla. 130

5.2. Estructura del conjunto de conocimiento. 131

5.2.1. Estructura jerárquica. 131

5.2.2. Definición de preguntas tipo y extracción de palabras clave. 133

5.2.3. Respuesta ante una consulta de usuario. 137

5.3 Sistema de Lógica Borrosa. 142

5.3.1. Variables de E/S. 145

5.3.2. Definición de los conjuntos borrosos de E/S. 150

5.3.3. Definición de reglas. 156

Capítulo 6. Nuevo método para la asignación de pesos basado en FL. 159

6.1. Introducción 160

6.2. Descripción del método de asignación de pesos TF-IDF. 160

6.3. Descripción del método de asignación de pesos con FL. 162

6.4. Implementación de ambos métodos de asignación de pesos. 163

6.4.1. Implementación del método TF-IDF. 163

6.4.2. Implementación del método basado en FL. 165

Asignador de coeficientes manual 177
6.5. Comparación: método TF-IDF vs. método basado en FL. 178
6.5.1. Pruebas realizadas..... 178
6.5.2 Análisis de resultados. 183
Análisis según las clases de preguntas tipo. 183
Análisis según el número de preguntas tipo de cada Objeto. 190
Capítulo 7. Resumen, conclusiones y futuras líneas de trabajo..... 199
7.1. Resumen. 199
7.2. Conclusiones..... 200
7.3. Futuras líneas de trabajo. 201
Anexo A. El programa Un-fuzzy. 203
A.1. Ventana principal. 203
A.2. Universo de entrada. 205
A.3. Universo de salida. 206
A.4. Base de reglas. 207
A.5. Simulación. 209
A.6. Generación de código. 210
Anexo B. Resultados obtenidos con los dos métodos de TW. 213
B.1. Resultados obtenidos con el método basado en FL..... 213
B.2. Resultados obtenidos con el método TF-IDF. 218
Anexo C. Reglas borrosas..... 223
C.1. Reglas borrosas para la extracción de información 223
Reglas borrosas para el motor borroso de tres entradas. 223
Reglas borrosas para el motor de cinco entradas..... 224
C.2. Reglas borrosas para la definición de pesos. 229
Bibliografía. 233
Glosario de abreviaturas..... 243

Índice de figuras.

2.1. Disciplinas que contribuyen a la minería de datos..... 17
2.2. Ejemplo de árbol de decisión. 20
2.3. Arquitectura de una ANN unidireccional de tres capas: entrada, oculta y salida 21
2.4. Categorías en Minería de Datos..... 31
2.5. Subtareas de minería web. 33
2.6. Modelo de espacio vectorial..... 41
2.7. Ejemplo de un documento simple y su código HTML. 44
2.8. Términos índice..... 47
2.9. Mapa conceptual de la Web Semántica. 49
2.10. Interfaz del buscador de Webs Semánticas Swoogle. 50
2.11: IDE de desarrollo de ontologías, Protégé..... 52
2.12: Vista de clases de la ontología para SOSEXP. 53
2.13: Vista de un individuo (atributo) en la ontología. 54
2.14: Vista de un individuo (caso) de la ontología para SOSEXP..... 54
3.1. Modelo de arquitectura de computadores de Von Neumann..... 58
3.2. Ejemplos de conjuntos..... 62
3.3. Ejemplo. Grado de pertenencia de una persona al conjunto borroso gente joven..... 63
3.4. Distintos tipos de funciones de pertenencia (I)..... 64
3.5. Distintos tipos de funciones de pertenencia (II). 65
3.6. Ejemplo de funciones de pertenencia para la altura de un hombre. 67
3.7. Ejemplo de desborrosificación en MATLAB..... 69
3.8. Asociaciones entre nodos en AIR..... 71
3.9. Aplicaciones de FL a IR. 74
3.10. Algoritmo de clustering jerárquico aglomerativo. 76
3.11. Red de conceptos borrosa..... 78
3.12. Red de conceptos borrosa..... 79

3.13. Ejemplo. Jerarquía del concepto “Fallo” en la máquina AV_2011..... 83

4.1. Asistente virtual de NationalRail (UK), realizado con la tecnologia Lingubot. 90

4.2. Arquitectura del Agente Inteligente Ville 91

4.3: Arquitectura de una red bayesiana. 94

4.4. Avatar creado por el autor de la tesis con la demo del programa Q-avatar. 96

4.5. Interfaz gráfica de Bea, asistente virtual de Cajamadrid. 100

4.6. Interfaces gráficas de Beatriz y Clara (Agentes Inteligentes de Groupama y Clickseguros, respectivamente). 101

4.7. Interfaces gráficas de los Agentes Inteligentes de Ikea y Telefónica. 102

4.8. Interfaz gráfica de Bea, asistente virtual creada por Indisys. 103

4.9. Estructura jerárquica arborescente. 107

4.10. Base de datos con las respuestas a las consultas de usuario. 107

4.11. Esquema de la respuesta a una consulta de usuario 108

4.12. Base de datos que contiene la estructura jerárquica del conjunto de conocimiento. 108

4.13. Base de datos de términos índice. 110

4.14. Constitución del Agente Inteligente. 111

4.15. Modo de operación del Agente Inteligente con dos niveles jerárquicos. 114

4.16. Esquema del editor de sistemas de inferencia borrosos. 116

4.17. Editor FIS de MATLAB. 117

4.18. Editor de funciones de pertenencia. 118

4.19. Editor de reglas. 119

5.1. Página de bienvenida del portal web de la US. 122

5.2. Herramienta utilizada para las pruebas de los parámetros de lógica borrosa. 134

5.3. Selección de una pregunta tipo con la herramienta para pruebas. 135

5.4. Memoria y precisión en el sistema. 137

5.5. Resultados con tres entradas al motor de inferencia. 138

5.6. Resultados con cinco entradas al motor de inferencia. 139

5.7. Resultados con cinco entradas al motor de inferencia utilizando umbrales de salida variables. 140

5.8. Definición del número de entradas al motor borroso y de los umbrales de certeza en la herramienta de pruebas. 141

5.9. Resultados con un número variable de entradas al motor de inferencia utilizando umbrales de salida variables. 142

5.10. Definición de los conjuntos borrosos de entrada con el programa Un-fuzzy. 143

5.11. Definición de los conjuntos borrosos de salida con el programa Un-fuzzy. 144

5.12. Definición del número de parámetros de borrosificación y desborrosificación y de los conjuntos borrosos de entrada y salida..... 145

6.1. Esquema de la generación de coeficientes de peso para el método basado en FL. 156

6.2: Conjuntos borrosos de entrada. 159

6.3: Conjuntos borrosos de salida. 163

6.4. Asignador manual de coeficientes. 167

6.5: Base de datos de Excel con las preguntas tipo. 169

6.6: Base de datos con los coeficientes de peso para ambos métodos. 170

6.7: Resultados con el método TF-IDF. 172

6.8: Resultados con el método basado en FL. 173

6.9: Comparación de resultados para preguntas tipo principales para ambos métodos de TW. 175

6.10: Comparación de resultados para preguntas tipo similares para ambos métodos de TW. 177

6.11: Comparación de resultados para preguntas tipo imprecisas para ambos métodos de TW. 178

6.12: Comparación de resultados para preguntas tipo concretas para ambos métodos de TW. 179

6.13: Comparación de resultados para ambos métodos de TW para el caso de una pregunta tipo por Objeto. 182

6.14: Comparación de resultados para ambos métodos de TW para el caso de una pregunta tipo por Objeto. 183

6.15: Comparación de resultados para ambos métodos de TW para el caso de una pregunta tipo por Objeto. 185

6.16: Comparación de resultados para ambos métodos de TW para el caso de más de diez preguntas tipo por Objeto. 186

A.1. Ventana principal de Un-fuzzy. 192

A.2. Barra de botones de Un-Fuzzy. 193

A.3. Cuadro de diálogo Definir variable de entrada. 194

A.4. Cuadro de diálogo Variables del Universo de Salida. 195

A.5. Cuadro de diálogo Definición Rápida de la Base de Reglas 196

A.6 Cuadro de diálogo Definición de las Reglas de la Máquina de Inferencia. 196

A.7. Cuadro de diálogo Calcular Salidas. 197

A.8. Cuadro de diálogo Análisis Paso a Paso. 198

A.9. Cuadro de diálogo Generación de Código Fuente. 199

Índice de tablas.

2.1: Diferencias entre Web 1.0 y Web 2.0. 32

2.2: Tipos de contenido en la Web 2.0. 37

3.1. Operaciones básicas de la lógica borrosa. 74

3.2: Instancias de Relaciones Biológicas borrosas extraídas de la recopilación GENIA (en inglés). 90

4.1: Posibles estados emocionales de un Agente Inteligente. 106

4.2: Posibles tipos de personalidad de un Agente Inteligente. 106

5.1. Estructura jerárquica del portal web de la US. 131

5.2. Ejemplo de definición de preguntas tipo. 133

5.3. Resumen de la metodología empleada. 135

5.4. Coeficientes de peso para el Nivel de Tema para distintos términos índice. 136

5.5. Salidas del motor de inferencia borroso para el Nivel de Tema para los distintos subconjuntos (Temas). 137

5.6. Coeficientes de peso para el Nivel de Apartado para los distintos Apartados del Tema 10. 137

5.7. Coeficientes de peso para el Nivel de Apartado para los distintos Apartados del Tema 12. 138

5.8. Salidas del motor de inferencia borroso para el Nivel de Apartado para los distintos Apartados del Tema 10. 138

5.9. Salidas del motor de inferencia borroso para el Nivel de Apartado para los distintos Apartados del Tema 12. 138

5.10. Coeficientes de peso para el Nivel de Objeto para los distintos Objetos del Apartado 6 del Tema 12. 139

5.11. Salidas del motor de inferencia borroso para el Nivel de Objeto para los distintos Objetos del Apartado 6 del Tema 12. 139

5.12. Objetos del conjunto de conocimiento devueltos por el Agente Inteligente. 139

5.13. Preguntas tipo asociadas a los Objetos devueltos por el Agente Inteligente. 140

6.1. Lista de palabras más usadas en el conjunto de conocimiento acumulado. 165

6.2. Valor de la entrada al Asignador de Coeficientes asociada a P1 para el nivel jerárquico de Tema. 166

6.3. Lista de palabras más usadas en un solo Tema. 168

6.4. Valor de la entrada al Asignador de Coeficientes asociada a P1 para el nivel jerárquico de Apartado. 168

6.5. Valor de la entrada al Asignador de Coeficientes asociada a P1 para el nivel jerárquico de Objeto. 169

6.6. Valor de la entrada al Asignador de Coeficientes asociada a P2 para el nivel jerárquico de Tema. 169

6.7. Valor de la entrada al Asignador de Coeficientes asociada a P2 para el nivel jerárquico de Apartado. 169

6.8. Valor de la entrada al Asignador de Coeficientes asociada a P3. 170

6.9. Valor de la entrada al Asignador de Coeficientes asociada a P4 170

6.10. Definición de las reglas borrosas para la asignación de coeficientes en los niveles jerárquicos de Tema y Apartado. 172

6.11. Definición de las reglas borrosas para la asignación de coeficientes en el nivel jerárquico de Objeto. 172

6.12. Clases de preguntas tipo definidas para las pruebas. 176

6.13. Resultados obtenidos con los métodos TF-IDF y basado en FL. 180

6.14: Resumen de resultados para las distintas clases de preguntas tipo. 188

6.15: Agrupación de Objetos según el número de preguntas tipo por Objeto 189

6.16: Resumen de resultados para las distintas clases preguntas tipo. 195

B.1. Resultados detallados para el Tema 3 con el método basado en FL. 212

B.2. Resultados detallados para el Tema 3 con el método TF-IDF. 217

C.1. Reglas borrosas para el motor borroso de tres entradas para la Extracción de Información. 219

C.2. Reglas borrosas para el motor borroso de cinco entradas para la Extracción de Información. 224

C.3. Reglas borrosas para la asignación de coeficientes de peso (Niveles de Tema y Apartado). 226

C.4. Reglas borrosas para la asignación de coeficientes de peso (Nivel de Objeto). 227

Agradecimientos.

Realizar una tarea de tanta complejidad como una tesis doctoral es imposible sin la ayuda de las personas que nos rodean y a las que me gustaría agradecer su apoyo.

En primer lugar, me gustaría agradecer a mi director de tesis, Carlos León, la posibilidad que me brindó de entrar en el ámbito de la investigación hace ya algunos años, permitiéndome adentrarme en el apasionante mundo de la Inteligencia Artificial. Así mismo, también agradezco a mi compañero Ariel Gómez las largas horas compartidas investigando en este campo. También a Julio Barbancho, quien fue en gran medida responsable de que empezara a trabajar en esta gran familia que es el Departamento de Tecnología Electrónica, a cuyos miembros también doy las gracias por el gran compañerismo que han demostrado durante todos estos años.

Por último, me gustaría mencionar a mis padres, por el soporte que han constituido durante toda mi vida.

Capítulo 1. Introducción.

1.1. Planteamiento y objetivos.

La gran cantidad de información disponible en la actualidad provocada por el auge de las Tecnologías de la Información constituye una enorme ventaja para las necesidades de búsqueda de esta por parte de los usuarios de las nuevas tecnologías. Sin embargo, al mismo tiempo, surge también un gran problema derivado de la dificultad existente para distinguir la información necesaria de entre toda la enorme cantidad de datos innecesarios.

Por esta razón, los conceptos de Búsqueda de Información (*Information Retrieval*, IR) y Extracción de Información (*Information Extraction*, IE) han saltado a la palestra con fuerza en los últimos tiempos. En principio, ambas surgieron para la búsqueda y extracción de documentos, pero en los últimos años se ha generalizado su uso para la búsqueda de cualquier otro tipo de información, como puede ser la información contenida en una base de datos, una página web o, en general, cualquier conjunto de conocimiento. En particular, está muy extendido el uso del denominado Modelo de Espacio Vectorial (*Vector Space Model*, VSM), el cual está basado en la utilización de términos índice a los que se asocia un cierto peso, que representa la importancia de estos en el conjunto de conocimiento a considerar. Estas técnicas funcionan razonablemente bien a la hora de extraer información en muchos ámbitos, pero poseen el inconveniente de no ser tan eficientes cuando las búsquedas de un usuario no son demasiado concretas, o bien cuando existe una ingente cantidad heterogénea de información.

Por este motivo, se propone en esta tesis doctoral el desarrollo de un Agente Inteligente (también denominado en la bibliografía Asistente Virtual) que sea capaz de responder a las necesidades de los usuarios en su proceso de encontrar la información deseada en entornos en los que la información es ingente, heterogénea, vaga, imprecisa o desordenada. La aportación principal de esta tesis es la creación de un método general para la búsqueda y extracción de información basado en el uso de la Lógica Borrosa (*Fuzzy Logic*, FL), la cual es una herramienta ideal para la gestión de una información de las características antes mencionadas, y, en particular, la aplicación y validación de este método para la extracción de información de portales web, dado que los portales web son un claro exponente de información heterogénea y desordenada.

La otra aportación importante realizada por esta tesis está relacionada con la asignación automática de pesos en el Modelo de Espacio Vectorial (VSM). En esta tesis, se define un nuevo modelo de asignación de pesos basado también en Lógica Borrosa y que trata de sustituir al método clásico de asignación de pesos, denominado TF-IDF. Para demostrar la validez del

nuevo método, se han realizado pruebas sobre el portal web de la Universidad de Sevilla, demostrándose la mejora introducida con el nuevo método en cuanto a extracción de la información solicitada, además de conseguir extraer también información relacionada, que puede ser de interés para los usuarios del portal.

1.2. Estructura de la tesis.

La memoria de esta tesis doctoral se ha dividido en siete capítulos y tres anexos, cuya estructura se resume a continuación.

El capítulo 2 constituye una introducción a la Recuperación y Extracción del conocimiento y a las técnicas de procesado del lenguaje natural. Teniendo en cuenta la posterior aplicación realizada en la extracción de información de portales web, se presentan conceptos relacionados con la minería de datos y la minería web. Así mismo, se presenta el Modelo de Espacio Vectorial, en el cual está basado el diseño del Agente Inteligente o Asistente Virtual que realiza la Extracción de la Información. Como este Agente necesita interactuar con los usuarios en Lenguaje Natural, también se introducen las técnicas mediante las cuáles este es procesado, amén de la inclusión de una sección sobre las Webs Semánticas, con el fin de presentar una posible alternativa con un enfoque semántico al trabajo realizado en esta tesis, el cual tiene un enfoque vectorial.

El capítulo 3 explora la herramienta utilizada para dotar al sistema de Inteligencia Artificial, la Lógica Borrosa. Desde una pequeña introducción al mundo de la Lógica Borrosa se llega a la presentación de aplicaciones de la Lógica Borrosa a la búsqueda de conocimiento existentes en la bibliografía, tanto aquellas basadas en un modelo vectorial, como aquellas basadas en ontologías.

El capítulo 4 presenta el concepto de Agente Inteligente. En primer lugar, se expone el estado del arte en este campo referido a la investigación, para después pasar a analizar algunos de los Agentes Inteligentes (también denominados Asistentes Virtuales) en el ámbito comercial. Este análisis nos lleva a considerar cuáles son los mayores inconvenientes derivados del enfoque actual y por qué consideramos necesaria la utilización de la Lógica Borrosa para el diseño de este tipo de Agentes. El capítulo 4 finaliza con la presentación de un método general basado en Lógica Borrosa para la extracción de conocimiento en entornos en los que la información relevante es difícil de distinguir de la que no lo es.

En el capítulo 5 se valida el método de extracción de información mediante la aplicación de dicho método al portal web de la Universidad de Sevilla, comprobándose la idoneidad del uso de la Lógica Borrosa y utilizándose un sistema basado en una estructura jerárquica (con lo que se aprovecha la estructura de un portal web) y en el Modelo de Espacio Vectorial. Así mismo, se estudian los parámetros necesarios para un óptimo funcionamiento del sistema de Lógica Borrosa.

A continuación, y dado que era necesario definir automáticamente los pesos asignados a los términos índice generados para llevar a cabo el mencionado Modelo de Espacio Vectorial, se definió un nuevo método automático de asignación de pesos basado en la Lógica Borrosa en el

capítulo 6. En primer lugar, se presenta el método clásico de asignación de pesos, denominado TF-IDF, y se presenta así mismo el nuevo método basado en Lógica Borrosa. Posteriormente, se realiza un análisis comparativo de los resultados obtenidos con ambos métodos para el portal web de la Universidad de Sevilla.

En el capítulo 7 se exponen las principales conclusiones de los estudios realizados, así como las aportaciones realizadas a la comunidad científica, proporcionándose vías de continuación de las investigaciones realizadas por el autor en el campo de la Búsqueda y Extracción de Información.

Por último, en los anexos A, B, y C, se muestran, respectivamente, una descripción del programa Un-Fuzzy, utilizado para la creación de sistemas de Lógica Borrosa; una muestra de una pequeña parte de los resultados obtenidos para los dos métodos de asignación de pesos; y, finalmente, el desarrollo de las reglas borrosas utilizadas.

Un último apunte de esta introducción se refiere al uso del idioma inglés. Afortunada o desafortunadamente (afortunadamente por su uso como estándar *de facto* y desafortunadamente para los que no somos angloparlantes), el inglés se ha convertido en el idioma clave en la investigación científica. Se han intentado traducir muchos de los términos técnicos, a la vez que se han mantenido los términos en inglés y sus acrónimos, puesto que pueden resultar de un gran interés para quien esté atraído por este campo de investigación. Se incluye, así mismo, un glosario de términos al final de esta tesis para facilitar su comprensión.

Capítulo 2. Recuperación y Extracción del conocimiento; Procesado del Lenguaje Natural.

El acceso a los contenidos de un conjunto extenso de conocimiento acumulado (una base de datos, un conjunto de documentos, contenidos web, etc.) es un problema de creciente interés en los últimos años. Los usuarios de estas colecciones de datos pueden encontrarse con grandes dificultades para encontrar la información requerida. Estas dificultades se ven incrementadas cuando el usuario en cuestión no es un experto en la materia, cuando existen contenidos ambiguos o mal organizados, si el tema es complejo o cuando hay una gran cantidad de información difícil de gestionar.

Finalmente, las tentativas infructuosas de búsqueda de información pueden llegar a ser frustrantes para los usuarios por no usar el término o los términos apropiados para realizar las consultas (una máquina solo responderá adecuadamente si se le pregunta de manera exacta), pudiendo terminar todo en una gran paradoja: mientras menos sabe uno, más difícil es encontrar las respuestas. En muchos casos, la solución es la búsqueda de una persona experta en el asunto y, en realidad, la ayuda demandada consiste en un intérprete que tenga la capacidad de generar una búsqueda sintáctica y semánticamente correcta que conduzca a la obtención de las respuestas deseadas. Por lo tanto, existe la necesidad de un agente que interprete la información vaga de la que se dispone, proporcionando respuestas concretas que estén relacionadas de alguna manera con los contenidos del conjunto de conocimiento. Esto debe estar basado en la estimación de la certeza de la relación entre lo que hemos expresado en lenguaje natural y los contenidos almacenados en este conjunto de conocimiento.

En este capítulo de la tesis se aborda el estudio de cómo conseguir encontrar información útil en conjuntos de conocimiento extensos y de las diversas formas de afrontar este problema.

Dada la necesidad de extraer información a partir de la ingente cantidad de datos disponibles, en la sección 2.1 se realiza una introducción a la minería de datos, centrándonos a partir de la sección 2.2 en la minería web en particular. Las secciones 2.3 y 2.4 abordan las dos tareas cuya atención va a ocupar más en los capítulos posteriores: la Recuperación de Información y la Extracción de Información, respectivamente. Por último, se dedica un apartado

al Procesado del Lenguaje Natural, que tiene una importancia capital en las dos tareas anteriores.

2.1. Minería de datos.

2.1.1. Definición.

La industria y el mundo de los negocios están en la actualidad inundados de datos. De hecho, este es el signo más evidente de la revolución en las tecnologías de la información en la que nos encontramos inmersos. Es innegable que nos encontramos ante una sociedad muy rica en datos, aunque también lo es que dicha sociedad es pobre en conocimiento, requiriéndose grandes esfuerzos para descubrir conocimiento en tal cantidad de datos [HIROTA99].

La Minería de Datos (*Data Mining*, DM) o Descubrimiento del Conocimiento en Bases de Datos (*Knowledge Discovery in Databases, KDD*) se define habitualmente como la tarea de identificar patrones de interés y describirlos de una forma concisa y con significado [FRAWLEY91] o bien como la extracción automática de patrones de interés implícitos en grandes colecciones de datos [KLOGSEN02]. La minería de datos no es en realidad sino una parte de KDD, el cual consta de tres etapas: pre-procesado, minería de datos y post-procesado [ROMERO07].

2.1.2. Minería de datos y estadística.

La Minería de Datos es el descendiente y, según algunos, el sucesor de la estadística tal y como esta se utiliza actualmente. Estadística y Minería de Datos conducen al mismo objetivo, el de efectuar modelos compactos y comprensibles que rindan cuenta de las relaciones establecidas entre la descripción de una situación y un resultado (o un juicio) relacionado con dicha descripción. Las técnicas de Minería de Datos permiten ganar tanto en prestaciones como en manejabilidad e incluso en tiempo de trabajo. [HERNÁNDEZ04].

Podría definirse la estadística como la “ciencia y la práctica del desarrollo del conocimiento a través del empleo de datos empíricos expresados de forma cuantitativa” [HERNÁNDEZ04]. Está basada en la teoría estadística, la cual es una rama de la matemática aplicada. Dentro de la teoría estadística, la aleatoriedad y la incertidumbre se modelan según la teoría de probabilidad. La ciencia de la estadística está orientada principalmente hacia la extracción de características cuantitativas y estadísticas de los datos. Por ejemplo, cuando se determina la covarianza entre dos variables, lo que vemos en realidad es si esas variables varían de forma conjunta y medimos la fuerza de esta relación. Sin embargo, no se puede caracterizar esta dependencia a un nivel conceptual y producir una explicación causal y una descripción cualitativa de esta relación: no se puede intuir ninguna razón de esta dependencia, puesto que esta se refiere a factores que no se encuentran explícitamente en los datos analizados. En cambio, el proceso de minería de datos es interactivo, iterativo y exploratorio y, además, existe un pre-procesado de datos que es esencial para cualquier proyecto de minería de datos. La reducción y compresión de datos, la

limpieza de datos y la transformación de estos son procesos muy importantes en la minería de datos, pero que no tienen nada que ver con la estadística.

En definitiva, la minería de datos es el proceso automático de analizar datos con el fin de descubrir patrones y construir modelos predictivos. La minería de datos tiene unos fundamentos teóricos muy fuertes y bebe de fuentes tales como las matemáticas, la estadística y el aprendizaje automático.

2.1.3. Modelos de minería de datos.

La minería de datos tiene como objetivo analizar los datos para extraer conocimiento. Este conocimiento puede venir dado en forma de relaciones, patrones o reglas inferidos de los datos, constituyendo estos el modelo de los datos analizados. Existen muchas formas de representar los modelos, aunque básicamente pueden agruparse en dos tipos.

- Modelos predictivos: pretenden estimar valores futuros o desconocidos de variables (variables objetivo o dependientes) usando otras variables o campos de una base de datos (variables independientes o predictivas). Los modelos predictivos requieren de un conjunto de pruebas y de interacciones de entrenamiento. Algunas tareas que producen modelos predictivos son la clasificación y la regresión.

- Modelos descriptivos: identifican patrones que explican o resumen los datos, es decir, exploran las propiedades de los datos examinados. Algunas tareas que producen modelos descriptivos son las reglas de asociación o la agrupación (*Clustering*).

2.1.4. Relación de la minería de datos con otras disciplinas.

La minería de datos se ha desarrollado en paralelo con otras tecnologías, por lo que la investigación y los avances en la Minería de Datos se nutren de los que se producen en estas áreas relacionadas. Se pueden destacar como disciplinas más influyentes las siguientes:

- Bases de datos: los almacenes de datos y el procesamiento analítico en línea (*Online Analitic Processing*, OLAP) tienen una gran relación con la minería de datos, extrayéndose conocimiento novedoso y comprensible de grandes cantidades de datos. Cabe destacar las técnicas de indexado y acceso eficiente a los datos.

- Recuperación de información (*Information Retrieval,* IR): Consiste en obtener información desde datos textuales y en la búsqueda por Internet. Es la tarea básica en el desarrollo de esta tesis y es tratada en apartados posteriores.

- Estadística: ya se ha comentado en un apartado anterior su relación con la minería de datos y solo queda añadir que muchos de los conceptos utilizados en Minería de Datos provienen de la estadística: media, varianza, regresiones, etc.

- Aprendizaje automático: es el área de la Inteligencia Artificial que se ocupa de desarrollar algoritmos y programas capaces de aprender. La máquina aprende un modelo a partir de ejemplos y los usa para resolver el problema.

- Sistemas para toma de decisiones: el objetivo es proporcionar la información necesaria para la toma de decisiones efectivas en el ámbito empresarial o en tareas de diagnóstico.

- Visualización de datos: permiten al usuario descubrir, intuir o entender patrones difíciles de "ver" a partir de descripciones matemáticas o textuales de los resultados. Ejemplos de estas técnicas son las técnicas gráficas (diagramas de barras, histogramas, etc.), las icónicas (basadas en figuras o colores, p.e.), las basadas en píxeles, las jerárquicas y muchas otras.

- Computación paralela y distribuida: el coste computacional de las tareas más complejas de Minería de Datos se reparte entre distintos procesadores o computadores.

En la Figura 2.1, se resumen todas las disciplinas que tienen alguna relación con la minería de datos.

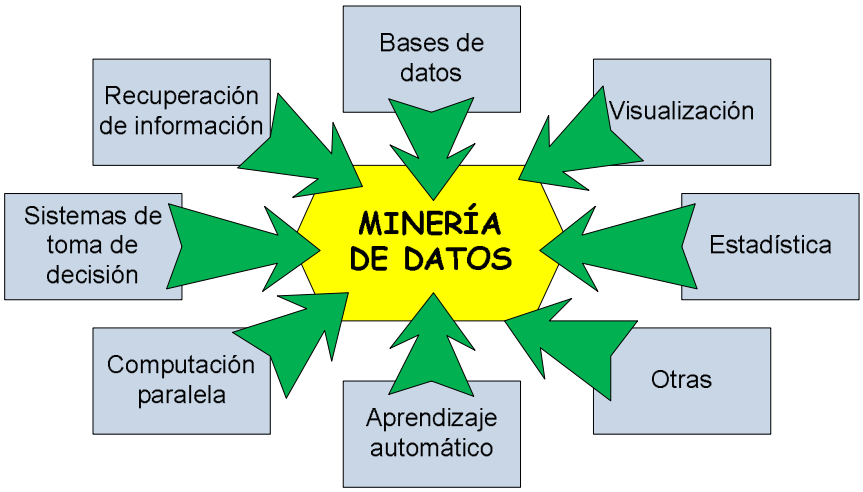


Figura 2.1. Disciplinas que contribuyen a la minería de datos

2.1.5. Aplicaciones de la minería de datos.

Teniendo en cuenta, pues, que la minería de datos se centra en las técnicas de extracción implícita de información potencialmente útil y previamente desconocida de grandes cantidades de datos, existen una serie de aplicaciones para la Minería de Datos, entre las que cabe destacar:

- Comercio electrónico (*e-commerce*), aprendizaje electrónico (*e-learning*) y sistemas educacionales. [ROMERO07]
- Aplicaciones financieras y de mercado [VERCELLIS09], [OLSON07].
- Solución de problemas (*problem solving*). [LIU07]
- Biología, medicina y bioingeniería [GREENES06].
- Telecomunicaciones [PIERRE02].
- Minería de textos (*Text Mining*). [CHAKRABATI00], [LOH03]
- Minería web (*Web Mining*). [PAL02], [KOSALA00], [TAO08]

Dado el carácter de esta investigación, en la cual se realiza una aplicación para la Recuperación de Información en la World Wide Web (WWW), el punto en el que se focalizará la atención es en la denominada Minería Web (*Web Mining*, WM), que constituye la sección 2.2 de esta tesis.

2.1.6. Características de la Minería de datos.

Una vez definida la minería de datos y sus aplicaciones, en este apartado se definen algunos de los conceptos claves para la minería de datos, como son la existencia de los denominados atributos, el tipo de aprendizaje utilizado y los algoritmos que existen para ello, y por último, los pasos que deben seguirse en proceso de minería de datos.

Atributos.

Un algoritmo de aprendizaje necesita datos para poder aprender y encontrar patrones. Estos datos pueden venir dados en forma de ejemplos, en los que cada ejemplo consta de los denominados atributos o dimensiones. Cada atributo debe ser independiente de otros atributos, no siendo posible computar un valor de un atributo basado en el valor del otro. Dependiendo de si los atributos pueden ser ordenados y el tipo de valores que puedan tomar, se pueden categorizar de esta forma:

- Valores numéricos: están asociados a un valor real. Pueden ser comparados y tomar valores continuos o discretos. Al poder ser comparados, los atributos

numéricos pueden ser también ordenados, existiendo además una medida de la diferencia de sus magnitudes.

- Valores ordinales: son también discretos, pero existe un orden asociado a ellos. Por ejemplo, el conjunto de atributos [pequeño, mediano, grande] puede caracterizarse por el tamaño del objeto, aunque en este caso no es posible establecer una medida de la diferencia entre atributos, solo sabemos que "pequeño" es menos que "mediano", pero no cuánto.

- Valores nominales: son valores discretos, a menudo denominados también valores categóricos. Por ejemplo, el color de los ojos de una persona. No existe ningún orden ni medida, los ojos azules no tienen por qué estar delante o detrás de los verdes, ni más cerca de estos que de los negros.

Existen algoritmos que trabajan tanto con valores continuos como nominales, siendo posible convertir valores continuos en discretos y viceversa. En el último ejemplo, los atributos pueden tomar el valor 1 (verdadero) cuando los ojos de una persona tengan el color del atributo correspondiente (por ejemplo, ojos verdes) y 0 (falso), cuando no lo tengan. La relatividad de este concepto dará origen a la lógica borrosa, de la cual se hablará en posteriores apartados de esta tesis.

Aprendizaje supervisado / no supervisado.

En el aprendizaje supervisado, se dispone de un conjunto de entrenamiento, con una serie de instancias o ejemplos para los que el valor predicho es conocido. Cada ejemplo consta de unos atributos de entrada y un atributo predicho. El objetivo del algoritmo es la construcción de un modelo matemático que pueda predecir el atributo de salida dados una serie de atributos de entrada. Los árboles de decisión, las redes neuronales, las curvas de regresión o las redes bayesianas son ejemplos de este tipo de modelos predictivos. La exactitud del modelo predictivo viene dada por el buen comportamiento de dicho modelo en los datos no utilizados para el entrenamiento.

En el caso del aprendizaje no supervisado, no hay ningún valor predicho del que aprender. El algoritmo analiza los datos de manera que se formen grupos o *clusters*, con similares características. El agrupamiento de medias k o agrupamiento *k-means* y el clustering jerárquico son ejemplos de este tipo de aprendizaje.

Algoritmos de aprendizaje.

En este apartado, se da una visión de más alto nivel de algunos de los algoritmos de aprendizaje más habituales: árboles de decisión, *clustering*, curvas de regresión, redes neuronales y algoritmos bayesianos. A continuación se describe cada uno de ellos:

- Árboles de decisión: son de los clasificadores más extendidos y tratan únicamente con atributos nominales. Existen una serie de nodos (*nodes*) y enlaces (*links*),

siendo fácil llegar al valor final del árbol desde el nodo raíz mediante una serie de reglas del tipo SI…ENTONCES. En la Figura 2.2 se muestra un ejemplo de cómo acceder al valor final en función del valor de los atributos y una serie de reglas (SI el tono de la piel es claro y el índice de radiación está entre 3 y 5, ENTONCES el factor de protección a usar es de 25.

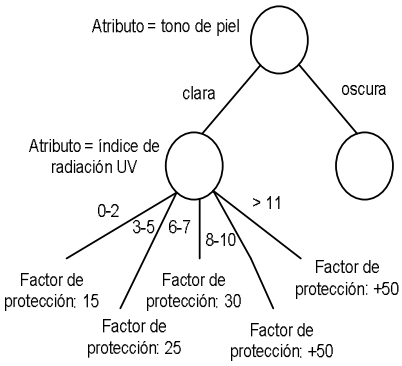


Figura 2.2. Ejemplo de árbol de decisión.

- Algoritmos de *clustering* (agrupamiento):

    - Algoritmo *k-means*: se suelen definir un número predefinido de k clusters aleatorios. Cada instancia se asocia entonces con el cluster cuyo centro esté más cercano a dicha instancia. Al final de cada iteración, las medias de los k clusters son recalculadas en función de los puntos asociados al cluster, continuando este proceso hasta que las instancias no muevan los clusters o bien se alcance un número máximo de iteraciones.

    - Clustering jerárquico: cada dato empieza perteneciendo a su propio cluster. A la siguiente iteración, los puntos más parecidos se combinan en un nodo, repitiéndose este proceso hasta que no quedan más puntos que combinar.

- Curvas de regresión: se basan en encontrar una función que una una serie de puntos en un espacio multi-dimensional. Los algoritmos basados en modelos de regresión representan los datos en forma de matriz, transformándola de manera que se computen los parámetros requeridos. Se requieren atributos numéricos para crear los modelos predictivos, de manera que se minimice el error cuadrático medio entre los valores predichos y el valor real para un cierto conjunto de entrenamiento.

- Redes Neuronales Artificiales (ANN, *Artificial Neural Networks*): son muy usadas, tanto para construir modelos predictivos como para realizar clasificadores. Para ello se usan básicamente el perceptrón multicapa (*Multi-Layer Perceptron*, MLP) y las funciones de base radial (*Radial Base Functions*, RBF). MLP consiste básicamente de un cierto número de capas, una de ellas de entrada, cuyo número es igual al número de atributos de entrada a considerar. Los valores de entrada

pueden ser escalados según la función de transformación de los nodos. Los enlaces de la red corresponden a un peso por el cual es multiplicada la salida de cada nodo para obtener, mediante una combinación lineal basada en los pesos de los enlaces que llegan a cada nodo de la siguiente capa, el valor de la entrada a cada nodo de dicha capa. De esta manera, existen una serie de capas, denominadas capas ocultas, hasta llegar a la capa de salida, la cual predice el atributo o los atributos de interés. Realizar un modelo predictivo MLP consiste en estimar los pesos asociados a cada uno de los enlaces, utilizándose un algoritmo de gradiente descendente para aprender los pesos asociados. El proceso general se conoce como retropropagación (BP, Back Propagation). En cuanto a las RBF, en primer lugar los datos se agrupan en k clusters usando el algoritmo k-means, mencionado anteriormente. Cada cluster corresponde a un nodo de la red, siendo la salida dependiente de la proximidad de la entrada al centro del nodo. La salida de esta capa es una combinación lineal de las entradas utilizando un aprendizaje por regresión lineal. En la Figura 2.3, se observa la estructura de una ANN con una capa oculta.

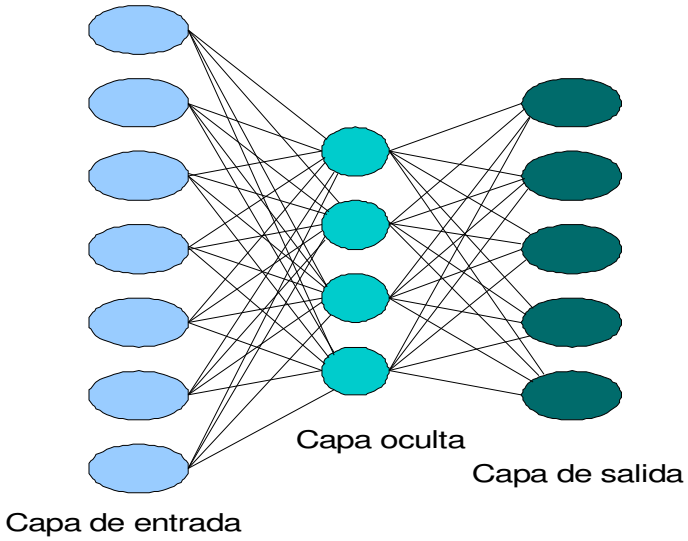


Figura 2.3. Arquitectura de una ANN unidireccional de tres capas: entrada, oculta y salida.

- Máquinas de vectores de soporte (SVM, *Support Vector Machine*): se trata de un algoritmo relativamente nuevo y que se está haciendo bastante popular para problemas de clasificación. Si se considera un espacio bidimensional con un gran número de puntos, existe un gran número de líneas que pueden usarse para dividir estos puntos en dos segmentos. Estas líneas se conocen como líneas de separación. Así mismo, se define el margen como la distancia entre una línea de separación y la línea paralela que pasa por el punto más cercano a esta. SVM selecciona la línea que tiene un mayor margen y los puntos por los que pasa esta línea se denominan puntos de vector de soporte.

- Algoritmos bayesianos: están basados en la teoría de la probabilidad. Ejemplos muy extendidos son el clasificador Naïve Bayes, en el que el algoritmo hace una

predicción estimando probabilidades a partir de los datos de entrenamiento, y las redes bayesianas, en la que se emplean una serie de grafos en los que un enlace representa una distribución condicional entre los nodos padre e hijo.

- Modelo de Espacio Vectorial (*Vector Space Model*, VSM): está basado en la representación vectorial de un documento o de cualquier conjunto de contenidos en general. Por su importancia para la extracción de información, se encuentra descrito más en profundidad en el punto 2.3.2 de esta tesis.

2.2. Minería Web.

2.2.1. Introducción a la Minería Web.

En la última década se ha producido un crecimiento explosivo de la información disponible en la World Wide Web (WWW). Hoy en día los buscadores de internet proporcionan ingentes cantidades de fuentes de datos de texto y multimedia. Esta profusión de recursos ha creado la necesidad de desarrollar técnicas automáticas de minería de datos (*data mining*) en la WWW, a las cuales se les ha dado el nombre genérico de minería web (*web mining*). [PAL02]. Ya en el año 2000 existían más de 800 millones páginas que cubrían la mayoría de los temas del conocimiento humano en la WWW [CHAKRABATI00] y el ritmo de crecimiento ha sido desde entonces cada vez mayor. Para saber aproximadamente el número de páginas web existentes, las mejores candidatas son Google y Yahoo!, ya que estas trabajan con los mejores motores de búsqueda [BOUTELL08]. Lamentablemente, ninguna empresa actualmente hace público el tamaño exacto de su índice y ni Google ni Yahoo! lo han hecho desde agosto de 2005 [YAHOO05]. En ese momento, el número de páginas web indexadas era de 19200 millones. Con estas perspectivas, se puede decir que Internet es un campo fértil para la investigación en minería de datos, de manera que se pueden establecer diferencias en la efectividad de la Recuperación de Información.

La necesidad de crear sistemas inteligentes tanto por parte del servidor como del cliente para extraer eficientemente el conocimiento tanto de internet como de sitios web particulares ha atraído la atención de investigadores procedentes de los dominios de Recuperación de Información (*Information Retrieval*, IR), Descubrimiento de Conocimiento (*Knowledge Discovery*, KD), Aprendizaje Automático (*Machine Learning*) e Inteligencia Artificial (*Artificial Intelligence*, AI) entre otros. Sin embargo, el problema de desarrollar herramientas automáticas para encontrar, extraer, filtrar y evaluar la información deseada por los usuarios a partir de datos web heterogéneos, distribuidos y no etiquetados está bien lejos de ser resuelto.

Para intentar sobreponernos a las limitaciones de las metodologías existentes, es una buena candidata la denominada Inteligencia Computacional (*Computational Intelligence*). La Inteligencia Computacional es un consorcio de metodologías que proporcionan capacidad de manejar situaciones ambiguas existentes en la vida real. Su objetivo es explorar la tolerancia, la imprecisión, la incertidumbre, el razonamiento aproximado y la verdad parcial para poder

conseguir tratabilidad, robustez, soluciones de bajo coste y un parecido razonable con la forma humana de razonar.

En la actualidad, las principales herramientas de Inteligencia Computacional incluyen la Lógica Borrosa (*Fuzzy Logic*, FL), las Redes Neuronales Artificiales (*Artificial Neural Networks*, ANN), los Algoritmos Genéticos (*Genetic Algorithms*, GA) y la teoría de Conjuntos Aproximados (*Rough Sets*, RS). FL proporciona un marco natural para el tratamiento de la incertidumbre; las redes neuronales (NNs) se usan para modelar funciones complejas y proporcionan aprendizaje y capacidad de generalización; los GA son una herramienta eficiente de búsqueda y optimización; y por último, los RS son de gran ayuda en el descubrimiento de conocimiento. [PAL02]

2.2.2. Datos en minería web.

La web no es más que una colección enorme de datos diversos y dinámicos, lo que provoca problemas de escalabilidad, heterogeneidad y dinamismo. La minería de datos intenta dar respuesta a la nada trivial tarea de identificar los datos válidos, novedosos o potencialmente útiles que encierra la web.

La minería web se puede definir de forma amplia como el descubrimiento y análisis de la información útil procedente de la WWW. En minería web, los datos pueden ser recogidos por parte del servidor, el cliente u obtenidos de bases de datos de una organización. Dependiendo de la localización de la fuente, el tipo de datos puede diferir, existiendo una gran variación en los contenidos (p.e, textos, imágenes, audio, símbolos). Esto hace que las técnicas utilizadas en minería web cambien según la tarea particular que haya que llevar a cabo. Sin embargo, algunas características comunes de los datos web son las siguientes:

- No etiquetados.
- Distribuidos.
- Heterogéneos
- Semiestructurados
- Variables en el tiempo.

Por lo tanto, la minería web trata básicamente con información de gran tamaño, con hiperenlaces y con las características antes mencionadas. Además, al ser un medio interactivo, la interfaz humana es un componente clave en la mayoría de los usos de la web. Algunas cuestiones que han salido a la luz, por consiguiente, se centran en:

- La necesidad de manejar preguntas sensibles al contexto e imprecisas.
- La necesidad de resumir y deducir.
- La necesidad de que exista una personalización y un aprendizaje.

Así, la minería web, en vez de ser considerada un caso particular de minería de datos, se garantiza un campo de investigación independiente, principalmente debido a las características anteriores de los datos y a las cuestiones relacionadas con el razonamiento humano. [PAL02]

2.2.3. La problemática de la Web 2.0.

En 2004, O'Reilly & Associates acuñaron el término Web 2.0, el cual consiguió finalmente una aceptación mayoritaria como término para esta nueva era (así como *puntocom o* Web 1.0 describió la época anterior). A pesar de la actual popularidad de la Web 2.0, aún no es fácil implementar muchos de sus principios [ALAG08]. Web 2.0 no es un concepto uniforme, sino un término genérico o un concepto para las nuevas tecnologías de Internet y sus aplicaciones.

La Web 2.0 puede ser vista como un renacimiento, una intensificación, una renovación o aún como una segunda generación de Internet, en la cual el usuario que genera los contenidos ocupa un lugar central. No es fácil definir el Web 2.0 debido a la longitud de este concepto. Por eso, se suelen hacer descripciones generales de la Web 2.0 en lugar de definiciones específicas. En [OSIMO07], se afirma que la Web 2.0 es un concepto relacionado tanto con la tecnología como con la actitud. Miller describe la Web 2.0 como la red usada como una plataforma, incluyendo a todos los dispositivos conectados; las aplicaciones Web 2.0 son aquellas que aprovechan al máximo las ventajas intrínsecas de esta plataforma, desarrollando software como un servicio continuamente puesto al día y que mejora a medida que es usado por más personas; consumiendo y remezclando datos de múltiples fuentes; incluyendo usuarios individuales; proporcionando sus propios datos y servicios de forma que permite la participación de otros, creando efectos de red mediante una arquitectura de participación; y yendo más allá del concepto de página que existía en la Web 1.0 para proporcionar experiencias más ricas al usuario. [MILLER05]. En la Tabla 2.1 se exponen algunas de las principales diferencias entre Web 1.0 y Web 2.0.

Web 1.0	**Web 2.0**
Enciclopedia Británica	Wikipedia
Páginas web personales	Blogs (Web logs)
Publicación	Participación
Directorios	Etiquetado
Netscape	Google
Nombres de dominio	Optimización de motores de búsqueda
Sistemas de gestión de contenido	Wikis
Consumo	Co-producción

Tabla 2.1: Diferencias entre Web 1.0 y Web 2.0.

La Web 2.0 se presenta a menudo como un modo revolucionario de compilación, organización y compartición de información. Los ejemplos más conocidos de uso de la Web 2.0

usos son Google, Weblogs, Wikipedia, YouTube, MySpace, Tuenti, Facebook y Second Life, entre otros [DE_KOOL08]. A pesar de que muchos usuarios han abrazado la Web 2.0, existen también voces críticas, según las cuales la Web 2.0 ha tenido una promoción exagerada y existen dudas acerca de si el potencial de la Web 2.0 se pondrá realmente en práctica.

A continuación, se describen algunos conceptos clave para la Web 2.0.

Inteligencia colectiva.

En la era post-puntocom, la web no deja de transformarse hacia la nueva era Web 2.0. Existen nuevas aplicaciones web en las que confían los usuarios, que los invitan a interactuar, a conectarse con otros usuarios, obteniendo información constante y permitiendo que la aplicación mejore gracias a esta interacción.

Los usuarios se expresan, tanto compartiendo sus opiniones sobre un producto o un servicio, como etiquetando contenidos, mediante su participación en comunidades online o bien distribuyendo nuevos contenidos entre los demás usuarios.

Este incremento de la interacción y participación de los usuarios provoca lógicamente un incremento de los datos que pueden ser convertidos en inteligencia por la aplicación, por lo que es necesario el uso de la llamada Inteligencia Colectiva para personalizar un sitio web para un usuario concreto, ayudándole a buscar soluciones y tomar decisiones. La Inteligencia Colectiva (*Collective Intelligence*, CI) se define como el uso efectivo de la información que proveen otros usuarios con el fin de mejorar la aplicación de uno [ALAG08].

Además de extraer inteligencia de un conjunto de interacciones y contribuciones de los usuarios, CI se encarga de actuar como filtro para saber que lo que puede resultar importante de una aplicación para cada usuario. Este filtro puede ser desde una simple influencia (puntuaciones, revisiones,…) a un modelo personalizado de recomendaciones de contenidos por parte de un usuario.

Tipos de Inteligencia Colectiva.

En una aplicación, existen distintos tipos de CI, a saber:

- Explícita: Es proporcionada por el usuario a la aplicación. Se trata de revisiones (*reviews*), etiquetas (*tags*), marcadores (*bookmarks*) y recomendaciones, entre otros.

- Implícita: se trata de información que los usuarios proporcionan tanto dentro como fuera de la aplicación y que está en formato no estructurado. Blogs, wikis, comunidades o redes sociales aportan este tipo de inteligencia.

- Inteligencia derivada: está basada en la información recogida de las dos anteriores, mediante técnicas como el agrupamiento (*clustering*), las búsquedas, la minería web y la minería de textos.

Mediante su interacción con una aplicación web, los usuarios proporcionan un rico conjunto de informaciones que pueden ser convertidas en inteligencia. Existen dos fuentes principales de alimentación para la inteligencia de una aplicación:

- Basada en contenidos (*content-based*): los contenidos son, generalmente, palabras o frases clave.

- Basada en la colaboración (*collaborative-based*): basada en la interacción entre usuarios.

Extracción de inteligencia a partir de etiquetas.

En aplicaciones centradas en el contenido, sobre las cuales volveremos en el punto 2.2.4 de esta tesis, los usuarios suelen navegar por el contenido mediante categorías o menús creados por los editores del sitio web. Estas categorías pueden estar estructuradas de manera jerárquica, ayudando a que se pueda encontrar con mayor facilidad el contenido buscado. Desde el punto de vista de la experiencia de los usuarios, esta navegación puede ser tediosa, dado que un usuario podría tener que navegar a través de múltiples subcategorías antes de encontrar el contenido deseado.

Una alternativa a la categorización manual es realizar sistemas que sean capaces de aprender de cada usuario, y del contenido en el que este esté interesado, y construir dinámicamente vínculos de navegación e hipervínculos a otros temas relevantes, cuyas palabras y frases clave resulten también familiares para el usuario.

En la actualidad, el uso de etiquetas (palabras y frases clave añadidas) está en todas partes en la web. Este simple proceso en el que el usuario agrega etiquetas o marcadores a los objetos produce una gran cantidad de inteligencia. Esta inteligencia puede manifestarse de las siguientes formas:

- Encuentro de objetos relacionados con otros mediante etiquetas comunes.

- Conexión de los usuarios con otros usuarios que busquen objetos con etiquetas similares.

- Direccionamiento al usuario a etiquetas alternativas que conduzcan al mismo objeto buscado.

- Encuentro de otros objetos relacionados.

Por otra parte, las etiquetas pueden ser creadas de diversas maneras:

- Etiquetas generadas profesionalmente: para aplicaciones que proporcionan diferentes tipos de contenidos – por ejemplo, artículos, videos, fotos, blogs – a sus usuarios., pueden usar sinónimos o palabras similares, usar términos compuestos y utilizar *stop*

words, es decir, suprimir las palabras que no aporten ninguna información. Las etiquetas generadas por profesionales tienen las siguientes características:

- Extraen conceptos relacionados con el texto.
- Capturan el valor semántico asociado mediante el uso de palabras clave que pueden no encontrarse en el texto.
- Pueden proporcionar una visión que no esté únicamente centrada en el contenido de interés, siendo posible dar una visión más general del tema.
- Se pueden usar sinónimos o palabras similares.
- Se pueden usar frases multi-término.
- Se puede usar un conjunto de palabras controlado, con un vocabulario también controlado.

Este tipo de etiquetas requieren un alto nivel de conocimiento por parte del profesional y pueden ser muy caras, especialmente si se está generando mucho contenido nuevo.

- Etiquetas generadas por usuarios: se suelen conocer por la palabra inglesa *tagging (etiquetado)*. Permiten a los usuarios asociar texto libre a un objeto, de forma que este le pueda resultar familiar, en lugar de usar una terminología fija determinada por el administrador del contenido o que hubiera sido creada profesionalmente. Tiene la ventaja de que los objetos más populares son etiquetados frecuentemente por los usuarios, lo que beneficia la utilización de la CI. Las etiquetas generadas por los usuarios tienen las siguientes características:
  - Los términos utilizados son familiares para el usuario.
  - Permiten la extracción de conceptos relacionados con el objeto.
  - Capturan el valor semántico asociado, utilizando palabras que pueden no estar en el texto.
  - Pueden usar frases multi-término.
  - Pueden incluir una gran variedad de términos que están cercanos en significado.

  Puede ser que sea necesario incluir aspectos de enraizado, los cuales se exponen más en profundidad en el apartado 2.3.2 de esta tesis, para plurales y palabras derivadas y que sea necesario un filtrado para distintos aspectos, tales como las palabras obscenas.

- Etiquetas generadas automáticamente: son generadas mediante un algoritmo automático. Las etiquetas generadas de esta forma tienen las siguientes características:

- Usan términos contenidos en el texto de la información, a excepción de sinónimos que hayan sido introducidos artificialmente.
- Habitualmente se trata de términos simples, dado que es difícil extraer automáticamente términos compuestos.
- Se pueden generar muchas etiquetas ruidosas (múltiples significados dependiendo del contexto, incluidos la polisemia y la homonimia).

Tipos de contenido Web.

El contenido no es más cualquier objeto que tenga un cierto texto asociado. Este texto puede estar en forma de titulo y cuerpo, palabras asociadas, preguntas y respuestas, o simplemente un título asociado a una foto o vídeo. El contenido puede haber sido desarrollado por un profesional o bien mediante los usuarios de un sitio web.

Se pueden distinguir distintos tipos de contenido web, como pueden ser artículos, productos, blogs, foros, y un largo etcétera. Los tipos de contenido, junto a la forma en la suelen ser generados, se resumen en la Tabla 2.1

Tipo de contenido	Descripción	Fuente
Artículos	Texto sobre un tema en particular. Contiene un título, un cuerpo y, a veces, subtítulos.	Creados profesionalmente o por medio de usuarios; noticias; agregados de otros sitios
Productos	Un objeto vendido en un sitio web. Generalmente se compone de título, descripción, palabras clave, reseñas, valoraciones y otros atributos como el precio, el fabricante y su disponibilidad en ciertas zonas geográficas.	Creado por el sitio web o por usuarios (p. e., eBay).
Términos de clasificación	Términos ad hoc con palabras clave o etiquetas asociados. Son creados para facilitar la navegación	Creados profesionalmente o automáticamente. Con menor frecuencia, creados por los usuarios.
Blogs	Diarios personales online en los que se escribe para compartir con otros usuarios: los usuarios pueden comentar las entradas.	Administradores del sitio o generados por los usuarios.
Wikis	Herramientas de colaboración online en las que los usuarios pueden editar, añadir o borrar páginas web de forma	Normalmente generadas por el usuario.

	cómoda.	
Grupos y foros de mensajes	Sitios en los que se pueden colocar preguntas y otros pueden responderlas, así como calificarlas por su utilidad.	Habitualmente generadas por los usuarios aunque las respuestas más complejas pueden requerir de la ayuda de un experto que trabaje para el sitio web.
Contenido multimedia	Vídeos, fotos, música…	Creado profesionalmente o por medio de usuarios.
Encuestas	Preguntas realizadas por un usuario, siendo la respuesta elegida entre un puñado de opciones.	Creadas profesionalmente o por medio de usuarios
Términos de búsqueda	Consultas realizadas por los usuarios.	Generadas por los usuarios.
Páginas de perfiles	Página del perfil de un usuario. Habitualmente, creada a partir de un listado de preferencias o informaciones sobre un usuario	Generadas por los usuarios.
Herramientas de trabajo.	Disponibles en el sitio web.	Creadas profesionalmente.
Chat logs	Transcripciones de chats online.	Expertos hacia los usuarios y viceversa.
Bancos de preguntas-respuestas (FAQ, Frequently Asked Questions)	Respuestas a preguntas planteadas por los usuarios a un administrador de un sitio web.	Preguntas generadas por los usuarios y respuestas generadas por expertos.
Reseñas	Reseñas acerca de un objeto, el cual puede pertenecer a cualquiera de los contenidos anteriores.	Creadas profesionalmente o por medio de usuarios.
Clasificados	Anuncios con un título y un cuerpo. Opcionalmente, pueden tener asociadas palabras clave.	Creadas profesionalmente o por medio de usuarios.

Tabla 2.2: Tipos de contenido en la Web 2.0.

Arañas web (web crawlers)

Un último concepto asociado a la Web 2.0 es el de araña web o *web crawler*. Este concepto surge por las propias características de la WWW, dado que esta es:

- Enorme, con billones de páginas web.

- Dinámica, con páginas que son añadidas, borradas o actualizadas constantemente.

- Un ente en constante expansión.

Con el fin de encontrar la información realmente útil entre los miles de millones de páginas web, surgen las arañas web (*web crawlers*). Una araña Web es un pequeño software que recorre el entramado de páginas Web de Internet de forma automática y sistemática.

En definitiva, una araña Web es un tipo especializado de lo que se denomina *webbot* - robot de la Web - que se encarga de llevar a cabo un tipo concreto de tareas. En particular, se encarga de recorrer las páginas Web de Internet, descargarlas al ordenador local, analizarlas sintácticamente y procesarlas. Las arañas, como cualquier otro tipo de software, pueden ser utilizadas con fines diversos, aunque su uso más conocido es el de agente software en los motores de búsqueda, donde su función básica es proporcionar al software encargado del indexado el contenido apropiado. Algunas arañas Web de este tipo son Googlebot o Yahoo slurp, las arañas web de Google y Yahoo, respectivamente. También existen arañas web con fines ilegales, los denominado *spambots*. Estos programas tienen un propósito malicioso y suelen recurrir a técnicas maliciosas como la suplantación de identidad (*phishing*) para lograr sus objetivos.

Por lo general, una araña Web dispone de un conjunto inicial de URLs (*Uniform Resource Locutor, Localizador Uniforme de Recursos)*, que no son más que una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos [RFC1738]. Este conjunto de inicial de URLs es conocido como semillas (*seeds*). La araña web va descargando las páginas web asociadas a las semillas y buscando dentro de éstas otras URLs. Cada nueva URL encontrada se añade a la lista de URLs que la araña Web debe visitar. A este proceso se le denomina recolección de URLs. Existen distintas políticas para escoger la siguiente URL que la araña Web visitará. En general, estas políticas se basan en las respuestas a preguntas tipo, como por ejemplo: ¿cómo de importante es la página en la que estoy?, ¿cómo de importante es el sitio en el que se encuentra la página web actual?, ¿he visitado ya alguna página web del dominio de la página a la que tengo intención de dirigirme?

A medida que la araña web accede a una nueva URL, la página web asociada se descarga al ordenador local. Una vez ahí, éstas son analizadas sintácticamente y procesadas. Es importante mencionar que ninguna araña Web puede acceder a todas las URLs que hay en Internet, pues el número de páginas existentes es gigantesco. Cuando la araña Web analiza sintácticamente una página Web, lo que hace es decidir qué partes de ésta son de utilidad. Por ejemplo, puede quedarse sólo con los enlaces, sólo con imágenes, sólo con texto, etc.

Por último, la araña Web procesa la información disponible, es decir, aplica algún tipo de algoritmo para conseguir el objetivo establecido. Por ejemplo, comprobar la disponibilidad de un enlace o verificar el tamaño de una imagen.

2.2.4. Categorías de la minería web.

La minería web puede ser de tres tipos:

- Minería Web de Contenidos (WCM, *Web Content Mining*).
- Minería Web de Estructuras (WSM, *Web Structure Mining*).
- Minería Web de Uso (WUM, *Web Usage Mining*).

WCM clasifica los documentos automáticamente o construye una base de información web multicapa. WSM extrae la estructura de una página web; WUM descubre patrones de acceso a las páginas en los usuarios [TAO08]. En la Figura 2.4, se ilustran las categorías de la Minería Web.

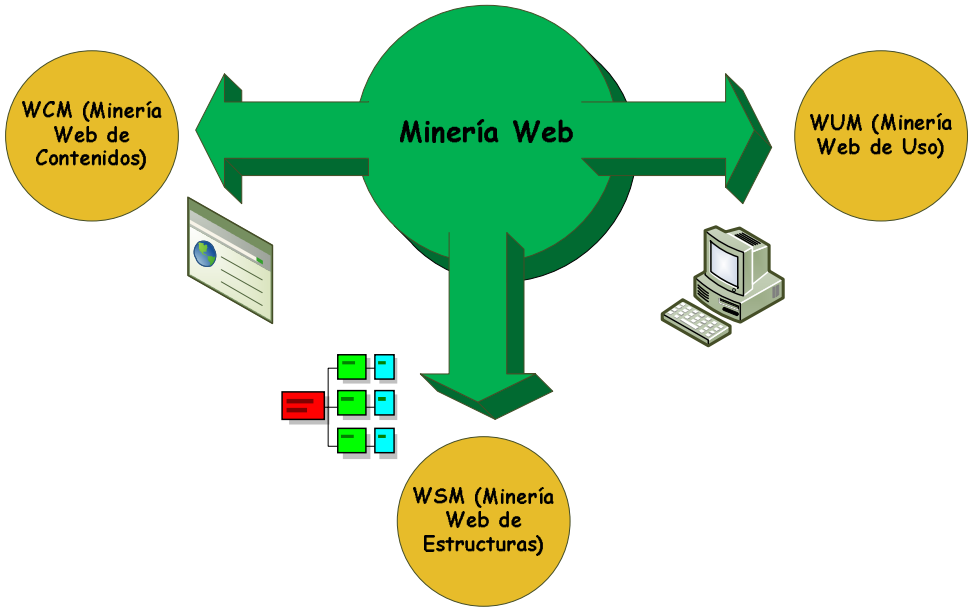


Figura 2.4. Categorías en Minería de Datos.

A continuación, se detallan las características de estas tres categorías [PAL02].

WCM (*Web Content Mining*).

WCM trata con el descubrimiento de información útil de los contenidos/datos/documentos/servicios de la web. Sin embargo, los contenidos web no se

componen únicamente de texto, sino también de audio, vídeo, datos simbólicos e hiperenlazados y metadatos.

WCM e IR están separados por una línea delgada. Algunos afirman que WCM en la web es un ejemplo de IR, mientras que otros asocian WCM con una IR inteligente. Según el contenido, existen dos estrategias para la WCM: aquellas que realizan la minería directamente con los contenidos de los documentos (*web page content mining*) y las que mejoran la búsqueda de contenidos de otras herramientas como los motores de búsqueda (*search result mining*).

Por otra parte, según la forma de afrontar el problema, podemos contemplar dos aproximaciones para WCM: la aproximación basada en agentes y la basada en bases de datos.

- La aproximación basada en agentes implica el desarrollo de sofisticados sistemas de AI que pueden actuar de forma autónoma o semi-autónoma en nombre de un usuario particular para descubrir y organizar información contenida en la web. Generalmente, la aproximación basada en agentes se puede a su vez subdividir en tres categorías: agentes de búsqueda inteligente, categorización/filtrado de información y agentes web personalizados [COOLEY97].

- La aproximación centrada en bases de datos se centra en las técnicas para organizar datos semi-estructurados de la web en conjuntos de recursos más estructurados, utilizando mecanismos de búsqueda y técnicas de minería de datos para analizarlos.

WSM (*Web Structure Mining*).

WSM extrae la estructura de los hiperenlaces, es decir, como están los documentos estructurados respecto a los otros (estructura interdocumental, a diferencia de la estructura intradocumental de WCM). La estructura se representa como un grafo de los enlaces en un sitio web o entre sitios web. WSM revela más información que la información contenida en los documentos: por ejemplo, los enlaces que apuntan a un documento pueden indicar la popularidad o importancia de un documento, mientras que los enlaces salientes indican la riqueza o variedad de los temas que contiene. Esto nos lleva a una organización jerárquica por temas que puede ser inferida directamente de los patrones de enlazado. Es posible incluso no especificar los documentos mediante palabras clave, sino mediante documentos ejemplares.

Un concepto íntimamente relacionado con la WSM por la importancia de la estructura, y que se trata en el apartado 2.5 de esta tesis, es el de web semántica.

WUM (*Web Usage Mining*).

Mientras que la minería de contenidos (WCM) y la minería de estructura (WSM) usan los datos reales o primarios de la web, la minería de uso (WUM), utiliza datos secundarios generados por la interacción de los usuarios con la web. WUM incluye datos de las conexiones a los servidores web, servidores Proxy o buscadores, perfiles de usuarios, archivos de registro, sesiones de usuario, búsquedas, clicks de ratón o scrolls, carpetas de favoritos, etc.

2.2.5. Componentes de la minería web y metodologías.

La minería web puede dividirse en cuatro tareas [ETZIONI96] como puede verse en la Figura 2.5.

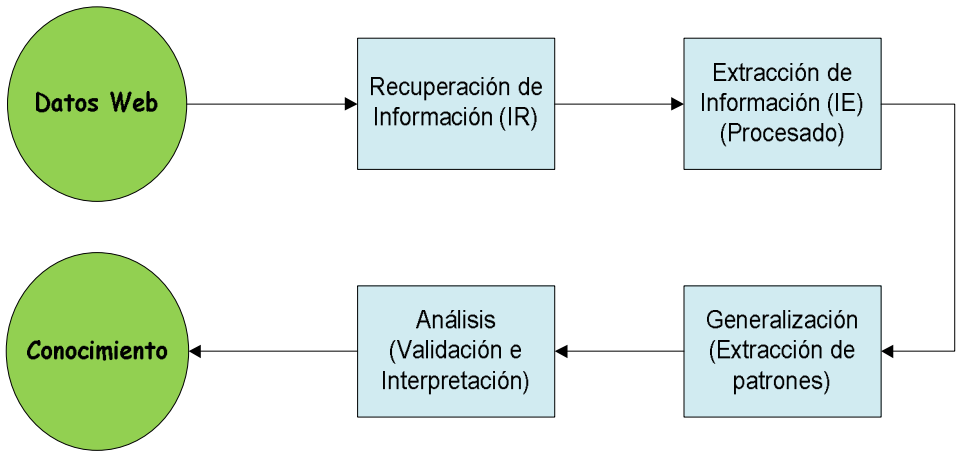


Figura 2.5. Subtareas de minería web.

A continuación, se describe cada tarea con una revisión de las metodologías/instrumentos existentes.

Recuperación de Información (*Information Retrieval*, *IR*).

La Recuperación de Información (*Information Retrieval*, IR) trata acerca de la recuperación automática de todos los documentos relevantes de un conjunto de conocimiento, asegurando al mismo tiempo que los documentos no relevantes recuperados sean los menos posibles. El proceso de IR incluye principalmente la representación de documentos, el indexado, y la búsqueda de documentos.

Existen diferentes técnicas de minería de contenidos web utilizadas por distintos autores para la Recuperación de Información en documentos semiestructurados. Estas técnicas se basan generalmente en la utilización de índices. Un índice es, básicamente, una colección de términos con indicadores a los lugares en los que puede encontrarse la información sobre los documentos. Sin embargo, indexar páginas web para facilitar la recuperación es un proceso bastante complejo, y un reto si se compara con el problema correspondiente asociado a bases de datos clásicas, donde las técnicas directas son suficientes. El enorme número de páginas web, su dinamismo, y su frecuente puesta al día hace que las técnicas de indexado parezcan aparentemente imposibles de aplicar. Actualmente, existen cuatro aproximaciones para el indexado de documentos en la web: indexado humano o manual; indexado automático; indexado inteligente o basado en agentes; e indexado basado en metadatos.

En la sección 2.3, se trata más en profundidad IR, dado que nuestras investigaciones se centran en técnicas IR aplicadas a conjuntos de conocimiento extensos.

Extracción de Información – *Information Extraction*, IE -

La Extracción de Información (*Information Extraction*, IE) consiste en la transformación de una colección de documentos, habitualmente con la ayuda de sistemas de IR, en información más fácil de asimilar y analizar. IE intenta extraer hechos relevantes de los documentos, mientras que IR selecciona los documentos relevantes. Por tanto, podríamos decir que IE trabaja con una granularidad más fina que IR. En todo caso, los conceptos IE e IR pueden llegar a confundirse en la práctica. [KOSALA00]

Aunque en nuestras investigaciones no nos hemos centrado en la estructura de los conjuntos de conocimiento, se trata de otro enfoque distinto y utilizado con frecuencia en nuestro campo de investigación, por lo que en la sección 2.4 se describen más en profundidad las técnicas de IE así como los modelos más comúnmente utilizados.

Generalización.

En esta fase, se usan el reconocimiento de patrones (*pattern recognition*) y las técnicas de aprendizaje automático (*machine learning*) sobre la información extraída. La mayoría de los sistemas de aprendizaje en máquinas utilizados en la web aprenden más sobre los intereses de los usuarios que sobre la propia web. Un gran obstáculo en el aprendizaje web es el problema del etiquetado: los datos son abundantes en la web, pero no están etiquetados. Muchas de las técnicas de minería de datos necesitan entradas etiquetadas como positivas o negativas respecto a algún concepto. Por ejemplo, si se proporciona un amplio conjunto de páginas web etiquetadas como positivas y negativas del concepto *homepage*, entonces es fácil diseñar un clasificador que prediga si una página web desconocida es una *homepage* o no. Desafortunadamente, las páginas web no están etiquetadas. Técnicas como el muestreo de incertidumbre (*uncertainty sampling*) reducen la cantidad necesaria de datos etiquetados, pero no eliminan completamente el problema. Una aproximación a la solución está basada en el hecho de que la web es mucho más que una colección enlazada de documentos, es un medio interactivo.

Los documentos procesables han llevado al desarrollo del concepto web semántica, que está inspirado en el hecho de que la mayoría de la información en la web se diseña para el consumo humano e incluso si deriva de una base de datos, la estructura de los datos no es evidente para búsquedas automáticas. A diferencia de la AI, en la que se procura emular el comportamiento del cerebro humano, la web semántica, en su lugar, desarrolla lenguajes para expresar información de forma procesable para las máquinas. Se trata este concepto en la sección 2.5.

Análisis.

El análisis de los datos es un aspecto que asume que existen suficientes datos para que la información potencialmente útil pueda ser extraída y analizada. Los seres humanos juegan un papel importante en el proceso de descubrimiento del conocimiento y la información en la web, dado que la propia web es un medio interactivo. Esto es especialmente importante para la validación y la interpretación de los patrones de minería. Una vez que los patrones han sido descubiertos, los analistas necesitan herramientas apropiadas para comprender, visualizar e interpretar estos patrones.

En definitiva, la minería web puede verse como el uso de las técnicas de minería de datos aplicadas a la búsqueda, extracción y evaluación automática de información para el descubrimiento del conocimiento de los documentos y servicios web.

2.2.6. Limitaciones de los métodos existentes de minería web.

Como resultado del rápido incremento y cambio de la información de la web, los sistemas pueden encontrar problemas en las diferentes etapas de la minería web. Algunos de estos problemas se describen a continuación.

Recuperación de Información, IR.

Se pueden encontrar las siguientes dificultades durante el proceso de IR:

- **Subjetividad, imprecisión e incertidumbre:** el objetivo de un sistema IR es estimar la adecuación de cada documento a las necesidades de los usuarios expresada en forma de búsqueda de usuario. Esta es una tarea dura y compleja que la mayoría de los sistemas IR encuentran difícil de manejar debido a la subjetividad, imprecisión e incertidumbre inherentes a dichas búsquedas. La mayoría de los sistemas IR existentes ofrecen un modelado muy simple de búsqueda, dando prioridad a la eficiencia en perjuicio de la exactitud. El procesado de búsqueda en los motores buscadores, que son parte importante en los sistemas IR, se basa ciegamente en encontrar palabras clave, sin tener en cuenta el contexto y la relevancia de las búsquedas respecto a los documentos.

- **Deducción:** Los motores de búsqueda no tienen capacidad de deducción.

- **Decisión suave (*Soft decisión*):** Las técnicas de procesado de búsqueda siguen el principio del rechazo de plano (*hard rejection*) cuando determinan la relevancia de un documento devuelto respecto a una búsqueda. Esto no es del todo correcto, dado que debería de ser una propiedad *gradual*.

- **Ranking de páginas:** Los rankings de páginas son importantes, ya que es difícil para los seres humanos revisar una lista entera de documentos devueltos por un motor de búsqueda en respuesta a una consulta del usuario. No obstante, no hay una fórmula

definitiva que refleje realmente la relevancia de los documentos. El esquema de determinación de rankings de páginas debe incorporar: pesos para los diversos parámetros que influyen en el documento devuelto, como la localización, la proximidad y la frecuencia; pesos que incluyan la "reputación" de la fuente; un ranking relativo al usuario, etc.

- **Dinamismo, escalado y heterogeneidad:** Los sistemas IR encuentran dificultades en el tratamiento del dinamismo, escalado y heterogeneidad de los documentos web por la variabilidad en el tiempo de los datos web. Muchos de los documentos devueltos por los motores de búsqueda pueden estar obsoletos, ser irrelevantes o no estar disponibles en el futuro.

Extracción de Información, IE.

La mayoría de los algoritmos IE utilizados por distintas herramientas se basan en la técnica del envoltorio (*wrapper*). Los envoltorios son procedimientos para extraer información particular de los recursos web. Su mayor limitación viene impuesta por el hecho de que cada envoltorio es un sistema IE personalizado para un sitio web en particular, no siendo aplicable universalmente.

Generalización.

Los problemas que se pueden encontrar en esta fase de la Minería Web se deben fundamentalmente a los siguientes aspectos:

- **Clasificación**: La comunidad IR ha explorado la clasificación de documentos como una alternativa a la organización de los resultados devueltos, pero aún existen problemas debidos a la propia naturaleza de los datos web: no solo son distribuidos, heterogéneos e imprecisos, sino que también tienen una gran dimensión y se pisan entre ellos (*overlap*).

- **Exclusiones:** Los servidores web realizan muchas exclusiones (malas observaciones), incluyendo datos incompletos, ruidosos y vagos debido a varias razones inherentes a la navegación web.

- **Minería de reglas de asociación**: Las técnicas actuales de minería en asociación de reglas no son apropiadas para hacer asociaciones lingüísticas que sean más comprensibles por los seres humanos. Algunos algoritmos que convierten reglas lingüísticas en numéricas sufren el problema del rechazo de plano. Ocurre lo mismo con los límites tajantes en los intervalos, los cuales no son intuitivos para los seres humanos. Por ejemplo, se podría separar joven de viejo en mayor/menor de 50 años, lo cual no se corresponde con la percepción humana.

Análisis.

El mayor problema que se presenta en esta fase es desde el punto de vista del descubrimiento y modelado del conocimiento. La salida de los algoritmos de minería del conocimiento no está habitualmente disponible para la interpretación directa por parte de los seres humanos. Esto es debido a que la mayoría de los patrones descubiertos están, principalmente, en forma matemática.

2.3. Recuperación de Información (*Information Retrieval*, IR).

2.3.1. Introducción.

El espectacular avance de las tecnologías de la información y, sobre todo, de internet ha causado un enorme aumento de la información disponible para los usuarios. Hoy en día, no es solo cuestión de encontrar la información, sino de seleccionar aquella que es esencial. La Recuperación de Información (*Information Retrieval*, IR) comenzó tratando con grandes colecciones de material textual, siendo su objetivo satisfacer preguntas y necesidades de los usuarios, habiéndose ampliado la definición en la actualidad a otros tipos de información [KWOK89].

IR consistía en principio en la búsqueda automática de todos los documentos relevantes en una colección de documentos, intentando además que el número de documentos no relevantes devueltos fuera el mínimo posible. Los objetivos principales de IR eran el indexado de textos y la búsqueda de los documentos útiles de una colección. Además, en la actualidad, IR incluye el modelado, la clasificación y la categorización de documentos, interfaces de usuario, visualización de datos, filtrado, etc. [BAEZA-YATES99]. El fin debe ser alcanzar una mejora en los resultados de recuperación en relación con mejores valores obtenidos para la memoria y la precisión, donde la memoria (*recall)* es el número de documentos relevantes recuperados dividido por el número de todos los documentos relevantes (Ecuación 2.1) mientras que la precisión (*precision*) es el número de documentos relevantes recuperados dividido por el número de todos los documentos recuperados (Ecuación 2.2). [RUIZ98]

$$\text{Memoria} = \frac{\textit{número de documentos relevantes recuperados}}{\textit{número total de documentos relevantes}}$$

Ecuación 2.1. Memoria.

$$\text{Precisión} = \frac{\textit{número de documentos relevantes recuperados}}{\textit{número total de documentos recuperados}}$$

Ecuación 2.2. Precisión.

Por ejemplo, en una colección de 100 documentos, en los que sólo 20 son relevantes para el usuario, si la búsqueda devuelve 18 documentos relevantes y 7 no relevantes, la Memoria tendrá un valor de 18/20, es decir, del 90 %, mientras que la precisión tendrá un valor de 18/25 y, por tanto, del 72 %.

En definitiva, durante mucho tiempo los investigadores del campo de la IR han tenido que afrontar el problema de cómo gestionar esta gran cantidad de información. IR ha sido usada principalmente para la clasificación de textos [ARONSON94], [LIU01] habiéndose introducido propuestas como el Modelo de Espacio Vectorial (*Vector Space Model*, VSM), el método del vecino K más próximo (*K nearest neighbour method*, KNN), también denominado *k-means*, el modelo de clasificación Bayesiano, Redes Neuronales y las Máquinas de Soporte Vectorial (*Support Vector Machine*, SVM) [LU02]. Estas técnicas se describieron brevemente en el apartado anterior de la tesis. VSM es el modelo usado con más frecuencia, debido a las ventajas que proporcionan su simplicidad y su alta velocidad de procesado, [LIU01], [LU02], [ZHAO02]. Debido a estas ventajas, este modelo se describe en profundidad en el siguiente apartado.

2.3.2. Modelo de Espacio Vectorial (Vector Space Model, VSM).

En el Modelo de Espacio Vectorial (*Vector Space Model*, VSM), el contenido de un documento está representado por un espacio multidimensional representado por un vector. Posteriormente, se pueden decidir las clases correspondientes del vector dado comparando las distancias entre vectores. El procedimiento del modelo de espacio vectorial puede ser dividido en tres etapas [RAGHAVAN86].

- La primera etapa consiste en el indexado del documento, donde los términos más relevantes son extraídos del texto de éste.

- La segunda etapa se basa en la introducción de pesos para los términos indexados, con el fin de mejorar la búsqueda de lo que puede ser relevante para el usuario.

- La última etapa clasifica el documento, en lo que concierne a la pregunta o consulta, según una cierta medida de semejanza.

Indexado de documentos

Es obvio que muchas de las palabras en un documento no describen el contenido, palabras como *el, ser* o *de*. Utilizando el indexado automático de documentos, estas palabras no significativas son eliminadas del vector correspondiente al documento, por lo que este sólo será representado por el contenido de las palabras relevantes. Este indexado puede estar basado en la frecuencia del término, de la que se hablará en profundidad en capítulos posteriores. En la práctica, el empleo de esta técnica es difícil de poner en práctica en un indexado automático. En

cambio, se utiliza una lista que contenga palabras comunes de alta frecuencia, lo que hace el indexado dependiente del lenguaje. En general, el 40-50 % del número total de palabras de un documento es suprimido con la ayuda de una de estas listas, denominadas *stop lists*, siendo las palabras que pertenecen a estas listas denominadas *stop words*. [SALTON83], [VANRIJSBERGEN79].

En cuanto al análisis morfológico, que es el análisis de la estructura interna de las palabras, puede ser aplicado a los términos en las consultas y en los documentos. La idea es que los resultados de recuperación puedan ser mejorados si se reducen las variantes morfológicas de un término a una sola forma.

Una técnica simple es desprender el sufijo de la palabra basándose en una lista de finales de palabra frecuentes en la lengua considerada. Existen algoritmos eficientes para este proceso, denominado de enraízamiento (*stemming*), dado que no se necesita consultar un diccionario. Sin embargo se producen multitud de errores debido a que esta técnica provoca la disminución de la información lingüística. Además, puede que las raíces producidas no correspondan a palabras existentes [PAIJMANS99]. Un enraízador bien conocido es el enraízador Porter [PORTER80]; otras herramientas utilizadas para el *stemming* son el enraízador-S y el enraízador Lovins, habiendo sido todos ellos desarrollados para el inglés. Un estudio hecho por Harman en 1991 mostró que ninguno de estos enraízadores mejora los resultados de recuperación de manera consistente [HARMAN91]. Otro estudio posterior para enraízadores similares al Porter, pero para el esloveno, han mostrado que la precisión podría ser mejorada [POPOVIC92], concluyendo que este tipo de enraízadores puede mejorar los resultados de recuperación para lenguas morfológicamente más complejas.

Por otra parte, existen algoritmos que, desde el punto de vista lingüístico para el análisis morfológico, comprueban las formas resultantes comparándolas con las entradas de un diccionario. A pesar de la multitud de problemas como la inconsistencia o la posible incompletitud del diccionario, errores de escritura en las pruebas, nombres propios, variaciones en acentos o guiones, etc., esta técnica muestra mejoras comparadas con el algoritmo de Porter. [KROVETZ93]. Los métodos no lingüísticos de indexado (NLI) también han sido puestos en práctica. El indexado probabilístico está basado en la suposición de que hay alguna diferencia estadística en la distribución del contenido de las palabras relevantes [VANRIJSBERGEN79]. Excede de la finalidad de la investigación una mayor profundidad en este tema, aunque se puede encontrar más información en artículos de Chakrabarti et al. y de Bookstein et al. [CHAKRABARTI98], [BOOKSTEIN95].

En conclusión, para poder aplicar VSM es imprescindible realizar un pre-procesado de la información. Para ello se utilizan las denominadas técnicas de Procesado del Lenguaje Natural (*Natural Language Processing*, NLP), que se describen en la sección 2.5 de este capítulo.

Introducción de pesos para los términos

La introducción de pesos para los términos ha sido explicada mediante el control de la exhaustividad y de la especificidad de la búsqueda, donde la exhaustividad está relacionada con la memoria y la especificidad con la precisión. Los pesos en el modelo de espacio vectorial se basan completamente en la estadística para términos simples, influyendo tres factores principalmente, el factor de frecuencia de término; el factor de frecuencia del conjunto y el

factor de normalización de longitud. Estos tres factores se multiplican para conseguir el peso del término.

La representación más usada es la llamada TF-IDF, en la que el peso de los términos está basado principalmente en la frecuencia de término (TF, *Term Frequency*) y la frecuencia de documento inversa (IDF, *Inverse Document Frequency*).

Para calcular el peso del término, la aproximación TF-IDF considera dos factores:

- TF: frecuencia de ocurrencia del término en el documento. Así, tf_{ik} es la frecuencia de ocurrencia del término Tk en el documento i.

- IDF: varía de forma inversamente proporcional con el número de documentos nk a los que el término Tk es asignado en una colección de N documentos. El típico factor IDF viene representado por la expresión log(N/nk + 0.01).

Introduciendo la normalización para simplificar los cálculos, la fórmula obtenida finalmente para el cálculo de los pesos es la definida en la Ecuación 2.3 [LIU01].

$$W_{ik} = \frac{tf_{ik} \times \log(N / n_k + 0.01)}{\sqrt{\sum_{k=1}^{m} tf_{ik} \times \log(N / n_k + 0.01))^2}}$$

Ecuación 2.3. Fórmula para el cálculo de pesos.

El tercer factor utilizado habitualmente es el factor de normalización de longitud del documento. Los documentos largos tienen por lo general un conjunto de términos mucho más grande que los documentos cortos, lo que hace que los primeros sean más probablemente recuperados que los segundos [SALTON83], [VANRIJSBERGEN79], [SALTON96]. El peso de un término que usa un factor de normalización de longitud viene dado por la ecuación 2.4.

$$W_{ik} = \frac{w_{ik}}{\sqrt{\sum_{i=1}^{m} (w_i)^2}},$$

Ecuación 2.4. Pesos aplicando un factor de normalización.

En la Ecuación 2.4, los w_i corresponden a los pesos de los otros componentes del vector [SALTON88].

Coeficientes de semejanza.

La semejanza en los modelos de espacio vectorial se determina por medio de coeficientes asociativos basados en el producto escalar del vector del documento y el vector de la consulta, donde la superposición de palabras indica la semejanza. El producto escalar, por lo general, se normaliza. La medida de semejanza más popular es el denominado coeficiente del coseno, que mide el ángulo entre el vector del documento y el vector de la consulta. La similitud entre los documentos i y j, tomada como dicho coeficiente del coseno, se puede expresar de la forma expresada en la Ecuación 2.5. [SALTON88]

$$Similitud(i,j) = \frac{\sum_{k=1}^{t} w_{ik} \times w_{jk}}{\sqrt{\sum_{k=1}^{t} (w_{ik})^2 \times \sum_{k=1}^{t} (w_{jk})^2}}$$

Ecuación 2.5. Similitud entre documentos.

En definitiva, y tras el análisis de estos tres últimos puntos, podemos representar el modelo de espacio vectorial de la manera que se muestra en la Figura 2.6.

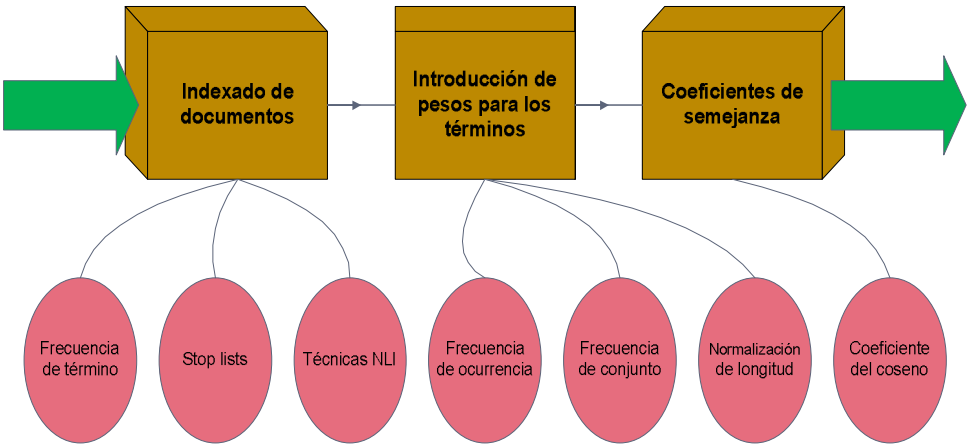


Figura 2.6. Modelo de espacio vectorial.

2.4. Extracción de Información (*Information Extraction*, IE).

2.4.1. Introducción.

Una vez que los documentos han sido recuperados, el desafío debe ser extraer el conocimiento y otras informaciones requeridas automáticamente, sin la interacción humana. La Extracción de Información (*Information Extraction, IE*) es la tarea de identificar los fragmentos específicos de un documento o, en general, de cualquier conjunto de conocimiento, que constituyen su principal contenido semántico.

Hasta ahora, los métodos principales de IE incluyen los denominados envoltorios (*wrappers*) de escritura que trazan un mapa de los documentos hacia algún modelo de datos. Un *wrapper* no es, pues, más que un patrón de diseño.

Los sistemas de extracción de información funcionan interpretando los distintos conjuntos de conocimiento y extrayendo la información de ellos. Por ejemplo, se pueden considerar los diversos sitios web como fuentes de conocimiento. Para hacer eso, el sistema procesa los documentos del sitio web para extraer fragmentos de texto relevantes y se usa una librería de envoltorios en la que cada envoltorio es un sistema personalizado de IE para cada sitio particular de Internet [KUSHMERICK02]. Otro método de IE para hipertexto viene dado en [FREITAG98], donde cada página se reduce a un conjunto de preguntas estándar o preguntas tipo. En todo caso, la mayoría de los sistemas de IE implican el uso de *wrappers*. Algunos ejemplos del empleo de *wrappers* para IE pueden encontrarse en el proyecto STAVIES [PAPADAKIS05] o en OMINI [LIU05]. El primero de ellos presenta un envoltorio automático que permite la extracción de tokens estructurales de un documento web. Esto se realiza usando etiquetas HTML, una estructura jerárquica y una agrupación (*clustering*) usando coeficientes de similitud S [i,j] entre los distintos nodos de la estructura. De manera similar funciona el proyecto OMINI, pero en este caso la agrupación se basa en el estudio de las propiedades de las etiquetas.

El problema, por lo tanto, se reduce a la identificación de los fragmentos de texto que contestan a estas preguntas específicas, a las que se han denominado preguntas estándar. Por consiguiente, IE intenta extraer conocimiento nuevo de los documentos recuperados aprovechando la estructura y la representación del documento, mientras que los expertos IR ven el texto de documento tal como una bolsa de palabras (*bag-of-words*) y no prestan la atención a la estructura del documento. La escalabilidad es el desafío más grande para la IE; no es factible construir sistemas IE que sean escalables teniendo en cuenta el tamaño y el dinamismo de la web. Por lo tanto, la mayoría de los sistemas IE extraen información de sitios específicos, centrándose en áreas más definidas.

Por otra parte, se requiere un sistema de preprocesado robusto para extraer cualquier clase de conocimiento de las bases de datos, incluso aunque estas tengan un tamaño mediano. Cuando un usuario entra en una página web, se accede a una gran variedad de archivos como imágenes, sonido, vídeo, y código html ejecutable. Por consiguiente, el servidor contiene muchas entradas

redundantes o irrelevantes para las tareas de minería de datos que, por lo tanto, deben ser eliminadas a través del preprocesado. Una de las técnicas de preprocesado usadas en IE es la inclusión del LSI (*Latent Semantic Indexing*, Indexado Semántico Latente), cuya función es transformar los vectores del documento original a un espacio de inferior dimensión, analizando la estructura de los términos en la totalidad de los documentos, de manera que los documentos similares que no comparten los mismos términos se colocan en la misma categoría. El enraizado (*stemming*) es otra técnica de preprocesado (como se ha indicado en la sección anterior), que reduce el tamaño las entradas extrayendo la raíz de las palabras. Por ejemplo, *informado*, *información* e *informando* se reducirían a su raíz *inform-*.

Debido a la naturaleza de la Web, muchos de los sistemas de IE se centran en sitios Web específicos para extraer información. Otros usan el aprendizaje automático o técnicas de minería de datos para realizar el aprendizaje de patrones y reglas de los documentos de forma automática o semi-automática [KUSHMERICK02]. Desde este punto de vista, la minería Web sería una parte de la extracción de información de una Web. Los resultados de un proceso de IE podrían estar en la forma de una base de datos estructurada o un resumen de los textos o documentos originales, por poner un ejemplo. Podríamos considerar incluso que IE consta de un proceso de Minería Web y un cierto pre-procesado, o que IE puede usarse para mejorar el proceso de IR, por lo que las fronteras entre todos estos campos no están excesivamente claras.

2.4.2. Modelos básicos en IE.

Básicamente, se pueden distinguir tres tipos de modelos en IE [CHAKRABARTI00]:

- Modelos para texto.
- Modelos para hipertexto.
- Modelos para datos semiestructurados.

A continuación, se describen con más detalle estos tres modelos.

Modelos para texto.

En el dominio de IE, los documentos se han representado tradicionalmente mediante el Modelo de Espacio Vectorial, ya descrito anteriormente en el apartado 2.3.2. Así mismo, se utilizan técnicas de procesado del lenguaje natural, las cuales se describen en el punto 2.5 de esta tesis.

Alternativamente, uno puede construir modelos probabilísticos para la generación de documentos [MARTHI03]. Ninguno de estos modelos tiene en cuenta la coherencia semántica o gramática. El modelo estadístico más simple es el modelo binario. En este modelo, cada documento es un conjunto de términos, el cual a su vez es un subconjunto del léxico (universo de términos posibles), y en el cual, el número de veces que aparece una palabra no cuenta. En el modelo multinómico, se puede imaginar un dado con tantas caras como palabras existan en el

léxico, teniendo cada cara una probabilidad *p* asociada, relacionada con el número de veces que aparece una palabra en el documento

Modelos para hipertexto.

El hipertexto consta de hiperenlaces añadidos al texto. En su modelo más simple, el hipertexto no es más que un grafo directo (D, L) donde D es el conjunto de nodos, documentos o páginas y L es el conjunto de enlaces. Modelos más refinados incluyen distribuciones de probabilidad para todos estos componentes.

Modelos para datos semiestructurados.

Además de los hiperenlaces, existen otras estructuras en la web. Una de las formas más utilizadas de estructuras interdocumentales son los directorios temáticos, como los que usa Yahoo!. Estos servicios han construido, con gran esfuerzo humano, una taxonomía gigante de directorios temáticos.

Los datos semiestructurados son un punto de convergencia importante entre la web y las bases de datos: la primera trata con documentos y las segundas con datos. Representaciones emergentes de datos semiestructurados (como XML) son variaciones del Modelo de Intercambio de Datos (*Object Exchange Model*, OEM). En OEM, los datos están en forma de objetos atómicos o compuestos; los objetos atómicos pueden ser enteros o cadenas de caracteres, mientras que los objetos compuestos se refieren a otros objetos etiquetados. HTML es un caso especial de esta estructura interdocumental.

La WWW consiste principalmente en documentos escritos en lenguaje HTML (*Hypertext Markup Language*). HTML es un lenguaje muy adecuado para la definición de los aspectos relacionado con la presentación de los documentos pero carece de cualquier medio para la definición de contenido semántico. Por lo tanto, la información contenida en los documentos apenas puede ser interpretada y procesada por un ordenador.

HTML no proporciona los medios suficientes para realizar una descripción semántica automática del contenido de documento. Las técnicas NLP usadas para la extracción de información de un texto simple, y que se comentan en el apartado siguiente, no son aplicables, dado que los documentos de HTML, por lo general, no contienen muchas oraciones completas o bloques de texto continuo. Por otra parte, las etiquetas HTML insertadas en el texto del documento proporcionan una información adicional que puede ser usada para identificar los datos [BURGET04]. Por ejemplo, la Figura 2.7 representa un documento simple y la parte relevante de su código de HTML. Si imaginamos que queremos extraer los nombres de países y sus capitales de este documento, mirando el código HTML, se puede observar que cada nombre de país está flanqueado por las etiquetas y (negrita) y que cada nombre de una capital está flanqueado por las etiquetas y (cursiva). Así, para extraer la información deseada de este documento, se puede crear un procedimiento lea el código HTML del documento, descubra estas etiquetas y almacene el texto existente entre cada par de estas. Este procedimiento es el que se denominó en la sección 2.3 de esta tesis *wrapper* o envoltorio.

Capitales **Francia** - *París* **China**- *Pekín*	Capital Cities France - Paris Japan - Tokyo

Figura 2.7. Ejemplo de un documento simple y su código HTML.

2.5. Procesado del Lenguaje Natural.

2.5.1. Aproximación al Procesado del Lenguaje Natural.

Las técnicas de Tratamiento del Lenguaje Natural (*Natural Language Processing*, NLP) se pueden usar para la Recuperación de Información (IR) de varias formas. Como se apuntó en secciones anteriores, el objetivo principal de aplicar el tratamiento del Lenguaje Natural a IR es alcanzar una mejora en los resultados de recuperación en relación con mejores valores obtenidos para la memoria (*recall*) y la precisión (*precision*).

Desde el punto de vista histórico, las técnicas NLP centradas en el Indexado Motivado Lingüísticamente (*Linguistically Motivated Indexing*, LMI) han sido el foco principal de investigación; LMI ha sido diseñado y evaluado en contraste con el Indexado No Lingüístico (*Non-Linguistic Indexing*, NLI) [SPARCK-JONES99]. Los sistemas LMI utilizan técnicas lingüísticas, usando la semántica y la sintaxis para identificar términos, encontrar unidades compuestas de varias palabras o caracterizar la estructura interna de una frase o documento. Por otra parte, los sistemas NLI no utilizan ninguna de estas técnicas y se limitan a aplicar criterios estadísticos y utilizar las denominadas *stop words*.

Hay dos aproximaciones diferentes para integrar técnicas NLP y recursos en la IR:

- El Indexado Motivado Lingüísticamente (LMI) se usa para crear términos índice para un modelo de espacio vectorial, al que nos hemos referido en la sección 2.3.2, o para sistemas de búsqueda booleana. En el primer caso, los documentos o búsquedas son convertidas en vectores con un cierto peso, devolviéndose aquellos vectores similares a los de la consulta; el segundo caso se basa en el álgebra de Boole.

- Sistemas basados en Inteligencia Artificial (*Artificial Intelligence*, AI), que tratan de emparejar una consulta con las representaciones semánticas de los textos de entrada.

Además de para NLP, otros usos de LMI se encuentran, por ejemplo, en la traducción automática (*Machine Translation*, MT), complementando a la traducción basada en reglas, a la traducción basada en estadística y a la traducción basada en ejemplos.

2.5.2. Técnicas NLP.

Las tentativas de aplicar módulos NLP para crear índices son casi tan antiguas como la recuperación automática en general. Los primeros desarrollos trataron de imitar la clasificación humana, y bastantes de estos sistemas aplicaron la indexación asignada, basada en el razonamiento humano. Un ejemplo utilizado actualmente basado en estos sistemas es el MeSH (*Medical Subject Headings*), creado con el propósito de indexar los artículos y libros sobre las ciencias biológicas [ARONSON97].

Las técnicas de minería de datos utilizan análisis estadísticos para descubrir reglas de asociación y patrones de interés entre distribuciones de los denominados términos índice o palabras clave (*keywords*). El problema estriba en que la asignación de estas palabras clave a cada conjunto de conocimiento (p.e., un texto o una página web) debe ser realizada previamente a la aplicación de dichas técnicas. [LOH03].

Existen diversas formas de realizar esta asignación, entre las que cabe destacar las siguientes:

- Extracción automática de los términos más comunes de un texto: los términos más frecuentes se asignan como palabras clave. Sin embargo, los textos no son preprocesados por lo que no se analiza el contexto en el que se encuentran dichos términos y se genera un modelo difícil de entender sin una lectura plena del texto.

- Extracción automática de términos usando información sintáctica: la idea es no usar todos los términos sino sólo los realmente significativos. El problema viene dado por los errores semánticos que pueden causar la sinonimia (palabras diferentes para el mismo significado), la polisemia (una palabra con diferentes significados) y las palabras con la misma raíz.

Además, como las experiencias con motores de búsqueda en la WWW muestran, buscar en grandes bases de datos mediante un solo término, a menudo devuelve muchos (demasiados) aciertos. Es por tanto importante restringir estas búsquedas mediante, por ejemplo, consultas multi-término. Esto requiere la posibilidad de comparar dicha consulta con los términos índice de los documentos, que puede estar basada en dos métodos de indexado [PAIJMANS99]:

- Precoordinación de términos índice: es el proceso de usar términos compuestos para describir un documento. Por ejemplo, esta sección puede ser incluida en un índice con el término compuesto "procesado del lenguaje natural".

- Postcoordinación de términos índice: es el proceso de usar términos simples para describir un documento pudiendo ser combinados en base a una consulta dada. Por ejemplo, esta sección puede ser incluida en un índice mediante las palabras "natural", "lenguaje", "procesado". Habría que combinar estos términos basados en una consulta como por ejemplo "natural AND lenguaje AND procesado".

Las técnicas LMI permiten el tratamiento de términos índice multipalabra de una forma diferente y quizás más elaborada, aunque en ambos casos, precoordinación y postcoordinación son posibles.

Un término índice puede ser un término solo (una palabra sola o una palabra raíz) o uno compuesto: este último puede ser un término complejo (encontrado por cualquier técnica LMI) o un término relacionado o similar. En la Figura 2.8 se clasifican los distintos términos índice. Estos pueden ser términos simples (palabras sencillas o raíces) o bien términos compuestos (complejos o relacionados).

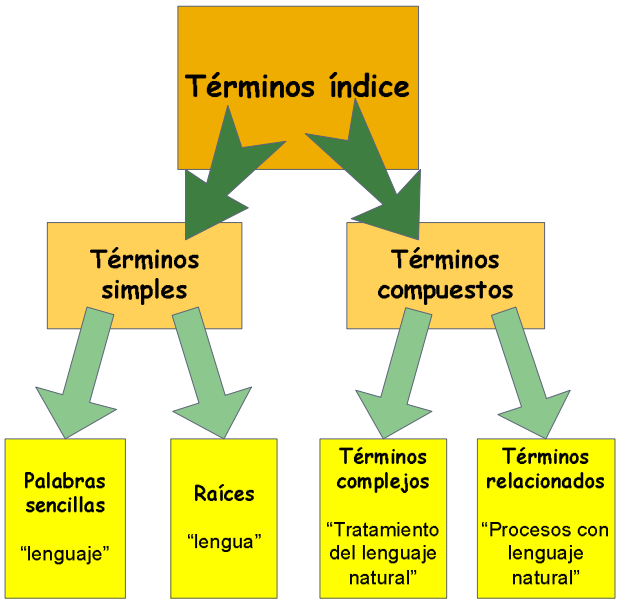


Figura 2.8. Términos índice.

Otro concepto importante a tener en cuenta son los denominados tesauros (*thesaurus*). Estos tesauros (del griego *thesauros*, almacén) suelen denominar un listado de palabras con significados similares, relacionados u opuestos. Por ejemplo, un libro de jerga para un campo especializado o, más técnicamente una lista de temas relacionados entre sí jerárquicamente utilizado para la indexación y recuperación de documentos.

En definitiva, un tesauro en un sistema de IR puede verse como una Base de Conocimiento que representa un modelo conceptual del dominio de algún tema. [LARSEN93]

2.5.3. Sintáctica y semántica. Webs Semánticas.

Aunque el tratamiento del lenguaje natural sea difícil, sus ventajas potenciales para la recuperación de documentos han hecho que una gran cantidad de investigadores se centren en el empleo de tanto de un tratamiento sintáctico como de un tratamiento semántico [ARONSON94].

La sintaxis es la parte de la gramática que se encarga de estudiar las reglas que gobiernan la forma en que las palabras se organizan en sintagmas y, a su vez, estos sintagmas en oración, incluyendo el modo en que las oraciones se organizan en estructuras de texto. En cuanto a la semántica, esta se dedica al estudio del significado de los signos lingüísticos y de sus combinaciones. Por tanto, la web semántica vendría a ser una extensión de la Web actual dotada de significado, es decir, un espacio donde la información tendría un significado bien definido, de manera que pudiera ser interpretada tanto por agentes humanos como por agentes automáticos [BERNERS-LEE02].

De hecho, los orígenes de la Web se basaron en el carácter abierto y universal de la base de la Web: el lenguaje HTML, y el empleo de archivos ASCII y gráficos GIF y/o JPG, lo que permite a los buscadores clasificar los documentos HTML de la red y ponerlos en una página web a modo de índice o catálogo, que se puede mostrar por medio de un navegador. Gracias a que el lenguaje HTML se ajusta a unas normas estandarizadas, todos los ordenadores pueden reproducir correctamente estos documentos. Sin embargo, el lenguaje HTML se quedaba corto como lenguaje orientado a la presentación de datos, dado que la información que ofrece es muy limitada, no permitiendo describir datos y no siendo extensible, es decir, únicamente ofrece un pequeño número de etiquetas. El sistema evolucionó y se realizaron algunas mejoras para hacer este lenguaje algo más dinámico con la introducción de otros elementos como DHTML (HTML dinámico), Javascript, hojas de estilo e, incluso, se añadieron a la Web otros lenguajes que permitieran ofrecer una información más estructurada.

Así pues, el desarrollo de la Web semántica requiere la utilización de otros lenguajes como el lenguaje estructurado XML (*Extensible Markup Language*) y el lenguaje RDF (*Resource Description Framework*) que puedan dotar a cada página, a cada archivo y a cada recurso o contenido de la red, de una lógica y un significado, y que permiten a los ordenadores conocer el significado de la información que manejan con el fin de que esta información pueda no sólo ser presentada en pantalla, sino también que pueda ser integrada y reutilizada. De hecho, XML ha logrado convertirse hoy en un lenguaje estándar. Para poder explotar la Web semántica, existen también lenguajes semánticos más potentes, es decir, lenguajes capaces de representar el conocimiento basándose en el uso de metadatos y ontologías. Utilizando lenguajes como RDF y RDF Schema se puede, mediante relaciones taxonómicas, crear una jerarquía de conceptos. Además, estos lenguajes deben ser estandarizados y formalizados para que su uso sea universal, reutilizable y compartido a lo largo y ancho de la Web. Se necesita un lenguaje común basado en la web y con suficiente capacidad expresiva y de razonamiento para representar la semántica de las ontologías. De esta forma, la utilización de lenguajes tales como OWL (*Web Ontology Language*, Lenguaje de Ontologías Web) es un paso más en la consecución de la Web Semántica. Si los metadatos sirven para la estructuración del contenido, tanto los tesauros, mencionados en el punto anterior de esta tesis, como las ontologías, hacen posible una semántica para construirlos. Una ontología es una especificación de una conceptualización, es

decir, un marco común o una estructura conceptual sistematizada y de consenso, no sólo para almacenar la información, sino también para poder buscarla y recuperarla [ARANO03]. Una ontología define los términos y las relaciones básicas para la compresión de un área del conocimiento, así como las reglas para poder combinar los términos para definir las extensiones de este tipo de vocabulario controlado.

Para una Web Semántica es necesario, pues, crear una ontología o biblioteca de vocabularios descriptivos/semánticos, definidos en formato RDF y ubicados en la Web para determinar el significado contextual de una palabra por medio de la consulta a la ontología apropiada. De esta forma, agentes inteligentes y programas autónomos podrían rastrear la Web de forma automática y localizar, exclusivamente, las páginas que se refieran a la palabra buscada con el significado y concepto precisos con el que interpretemos ese término. Por lo tanto, para potenciar el uso de ontologías en la Web, se necesitan aplicaciones específicas de búsqueda de ontologías, que indiquen a los usuarios las ontologías existentes y sus características para utilizarlas en su sistema. La Web Semántica debe ser capaz de procesar contenido, razonarlo y hacer deducciones lógicas a partir de éste, y realizar, cuando un usuario quiera delegar ciertas tareas en el software, todas estas acciones de forma automática. En suma, el objetivo de la Web Semántica es que la Web pase de ser una colección de documentos a convertirse en una base de conocimiento. En la Figura 2.9, se observa un mapa conceptual de la Web Semántica.

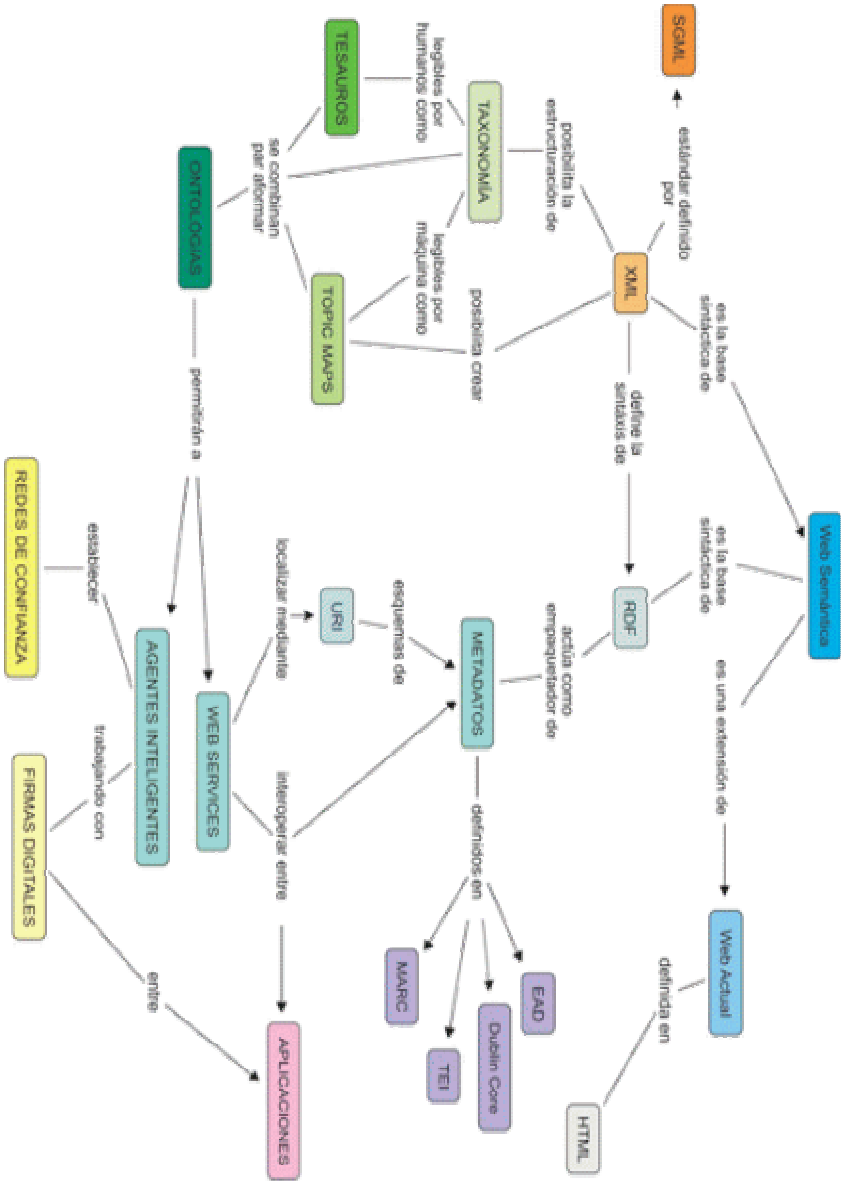


Figura 2.9. Mapa conceptual de la Web Semántica. [RODRÍGUEZ05]

Además, existen algunos buscadores para la Web Semántica, los cuales realizan búsquedas sobre ontologías y lenguajes semánticos como RDF y OWL. El más conocido es SWoogle: (Semantic Web Search, http://swoogle.umbc.edu/) desarrollado por la Universidad de Maryland (Baltimore, USA). Se trata de un buscador que rastrea ontologías (tiene indizadas más de 10.000), documentos y términos escritos tanto en RDF como OWL, es decir, busca documentos de la Web Semántica o *Semantic Web Documents* (SWDs). Curiosamente, tanto su propio

nombre como su interfaz de presentación son muy similares a los de Google, como se puede ver en la Figura 2.10.

Figura 2.10. Interfaz del buscador de Webs Semánticas Swoogle.

Dentro del amplio dominio NLP, la léxico-semántica es un problema clave, ya que representa el punto de convergencia entre la representación conceptual del conocimiento y ontologías extraídas de la web semántica [PRINCE06], entendiendo por ontología un esquema conceptual dentro de un dominio, con la finalidad de facilitar el conocimiento y la compartición de información entre diferentes sistemas. La idea de una ontología es, pues, la armonización del mundo real con la visión que puede ofrecer un modelo gramático. [GEHLERT07]. Las ontologías se componen de [GRUBER93]:

- Conceptos: son las ideas básicas que se intentan formalizar. Los conceptos pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento, etc.

- Relaciones: representan la interacción y enlace entre los conceptos de un dominio. Suelen formar la taxonomía del dominio. Por ejemplo: subclase-de, parte-de, parte-exhaustiva-de, conectado-a, etc.

- Funciones: son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología. Por ejemplo, pueden aparecer funciones como: asignar-fecha, categorizar-clase, etc.

- Instancias: se utilizan para representar objetos determinados de un concepto.

- Reglas de restricción o axiomas: son teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología. Por ejemplo: "Si A y B son de la clase C, entonces A no es subclase de B", "Para todo A que cumpla la condición B1, A es C", etc. Los axiomas, junto con la herencia de conceptos, permiten inferir conocimiento que no esté indicado explícitamente en la taxonomía de conceptos.

Como ejemplo de ontología se puede considerar el trabajo de J. Barbancho [BARBANCHO09], para el diseño y construcción de un sistema experto de apoyo a la gestión de usuarios del SOS, unidad de Soporte de Operaciones y Sistemas, constituida por el Centro de Atención de Llamadas y los Equipos de Intervención en los puestos de trabajo de los usuarios, y perteneciente al SIC, Servicio de Informática y Comunicaciones de la Universidad de Sevilla. El SOS se encarga de las incidencias producidas en el uso de ordenadores y redes. Se realiza una ontología, la cual constituye una taxonomía de los distintos conceptos que emplean los expertos (técnicos del SOS). La estructura del caso definido por el experto debe ser consistente con esta ontología.

La estructura del caso del sistema experto contemplado en SOSEXP conlleva una serie de atributos con diferentes campos:

- Nombre.

- Tipo: La propiedad puede ser entendida como: entero, *double*, booleano, cadena de caracteres, archivo, concepto, enumeración predefinida.

- Peso: Porcentaje de 0 a 1 que indica la importancia de la propiedad en el caso.

- Similitud local: Criterio de evaluación sobre el cual debe ser entendida la similitud sobre esta propiedad → Igualdad, similitud del tipo hermandad, similitud del tipo primos, similitud dentro de una distancia frontera, similitud del tipo coseno, similitud detallada, similitud profunda.

- Parámetros de similitud.

La generación de la ontología para SOSEXP permite plasmar la diversidad de incidencias posibles. Con ello se pretenden diseñar métodos de correspondencia entre casos más eficientes.

El entorno de desarrollo de ontologías elegido es Protégé (Figura 2.11), disponible bajo licencias GPL (*General Public License*). La ontología inicial establecida por los técnicos del SOS se muestra en la Figura 2.12. Las Figuras 2.13 y 2.14 reflejan la estructura jerárquica de la las distintas clases e individuos.

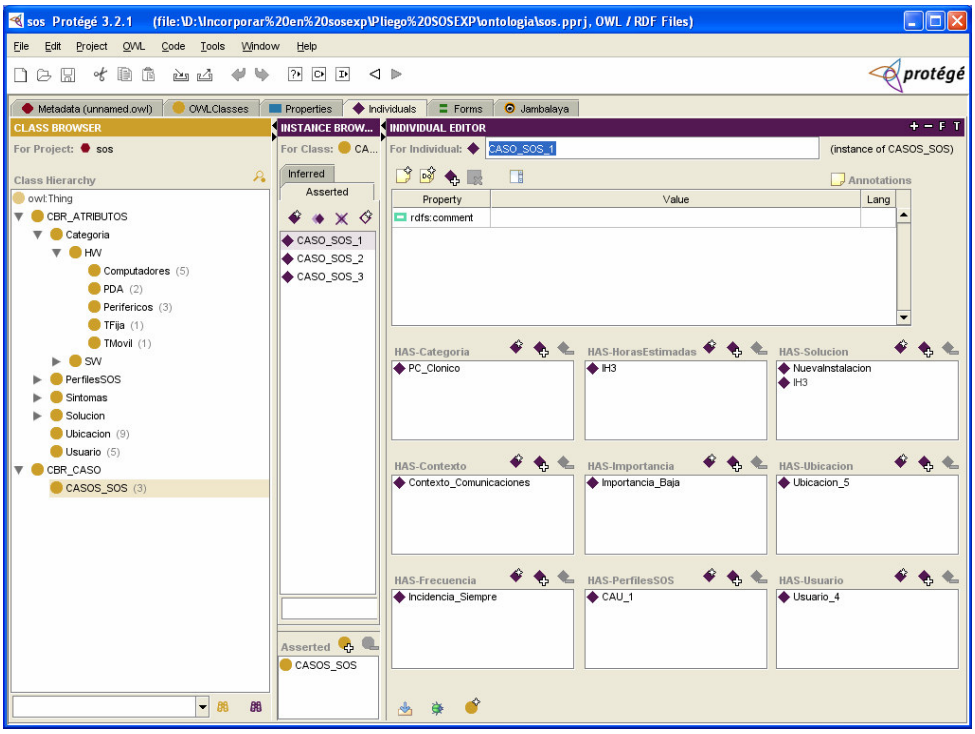


Figura 2.11: IDE de desarrollo de ontologías, Protégé.

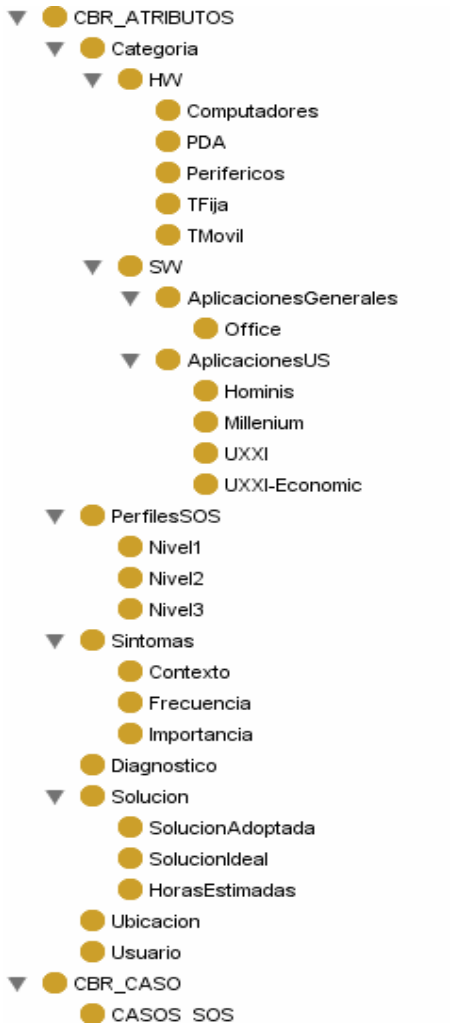


Figura 2.12: Vista de clases de la ontología para SOSEXP.

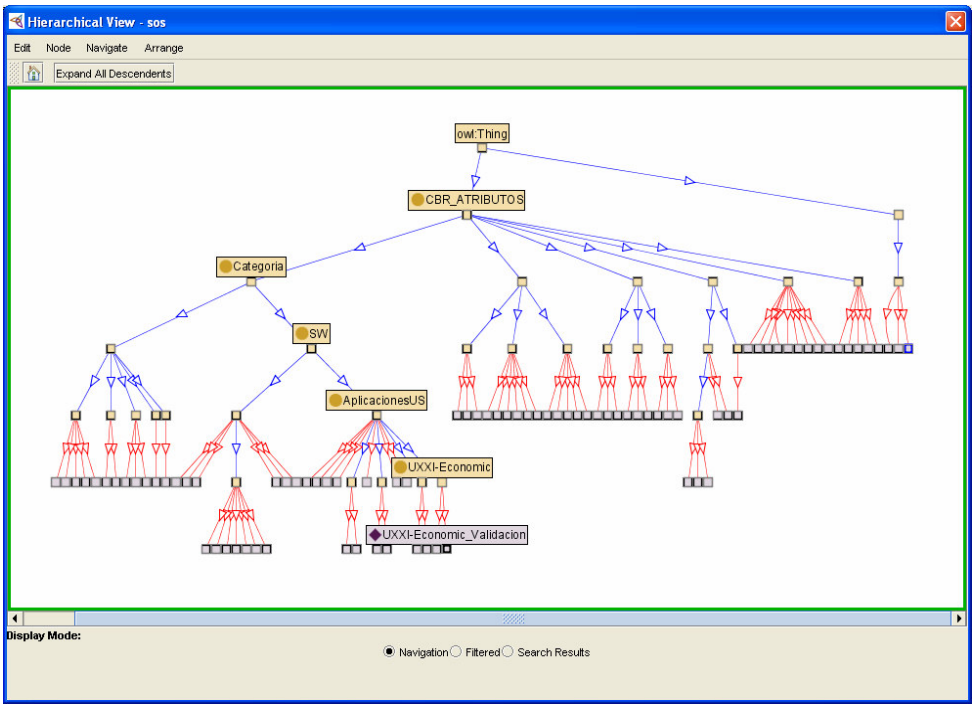


Figura 2.13: Vista de un individuo (atributo) en la ontología.

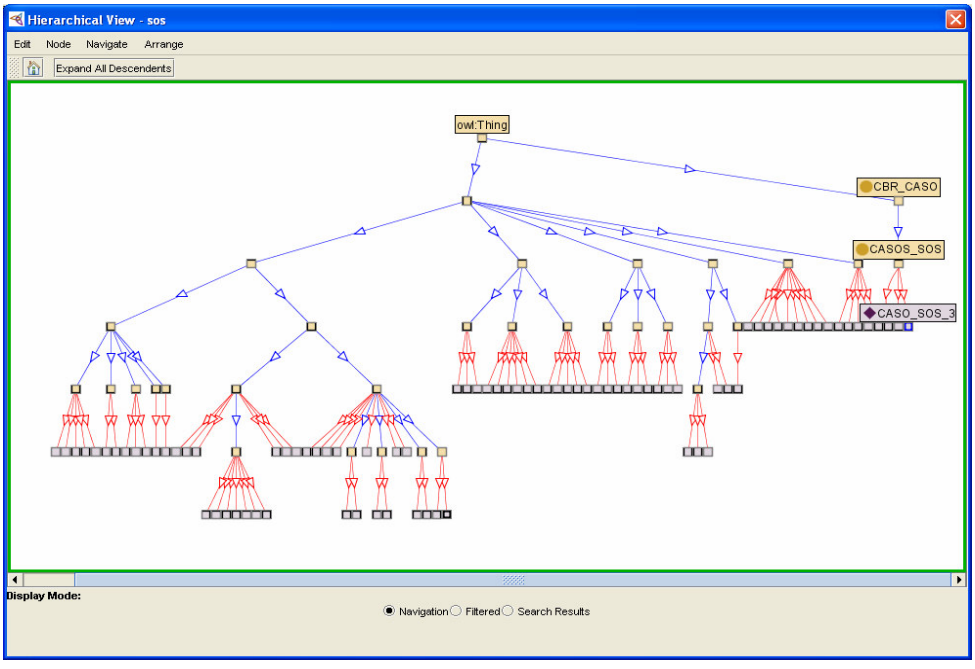


Figura 2.14: Vista de un individuo (caso) de la ontología para SOSEXP.

Por tanto, y en definitiva, existe la posibilidad de trabajar con dos enfoques distintos: el modelo vectorial, analizado anteriormente, y el modelo basado en webs semánticas. Si en el modelo vectorial, la recuperación y extracción de información se basaba en el *qué* de la información, en el caso de las webs semánticas se basa en el *cómo* está estructurada dicha información para recuperarla o extraerla. El problema que surge es que, en la actualidad, la web no provee aún de un gran número de ontologías o esquemas: hay pocas disponibles y en muy pocas materias. Es más, construir una ontología desde el principio puede resultar una tarea costosa y muy dependiente del ingeniero de conocimiento que la desarrolle [IANNONE07].

Por estas razones, en esta tesis nos inclinamos por un enfoque vectorial, aunque consideramos el estudio del modelo basado en webs semánticas un interesante campo de investigación.

Capítulo 3. Herramientas basadas en Inteligencia Computacional para la búsqueda automática de conocimiento.

El objetivo de cualquier sistema de acceso al conocimiento debe ser el de satisfacer las necesidades del usuario al acceder a los recursos que contienen dicho conocimiento. Sin embargo, nos encontramos con algunos problemas que deben afrontar los diseñadores de estos sistemas [LARSEN99]:

- Las necesidades de información de los usuarios son vagas o difusas.

- Las necesidades de información pueden cambiar a medida que el usuario recibe esta información durante la sesión de búsqueda (*query*).

- Los usuarios no son conscientes de sus necesidades exactas de información.

- Las necesidades de información no son fáciles de expresar en una pregunta al sistema de acceso al conocimiento.

Si en el capítulo anterior de la tesis se abordaba el estudio de las diversas formas de acceder a la información, en este capítulo se describen las herramientas que posibilitan este acceso. Habida cuenta de los problemas descritos más arriba, puede ser deseable el uso de la Inteligencia Artificial (AI), por su capacidad para lidiar con informaciones imprecisas y su flexibilidad. Concretamente, nos ceñiremos a una de sus ramas, la Inteligencia Computacional (*Computational Intelligence*, CI).

En la sección 3.1 se realiza una introducción a la CI, con un interés particular en dos de sus ramas principales, las redes neuronales y la lógica borrosa. En la sección 3.2, se hace especial hincapié en esta última por su mayor aplicación a los sistemas de búsqueda de información (IR). Por otra parte, la sección 3.3 muestra distintas aplicaciones de la CI, mientras que la sección 3.4 se centra en distintas aplicaciones basadas en la CI existentes en la bibliografía para la búsqueda de conocimiento.

3.1. Inteligencia Computacional (*Computational Intelligence*, CI).

El avance de la electrónica y de la informática en los últimos tiempos ha sido inmenso, pese a lo cual aún existen diversas tareas que no son resueltas con eficacia. Estas tareas tienen relación habitualmente con el reconocimiento de patrones o el funcionamiento de sistemas en entornos ruidosos y/o imprecisos. Todo esto se podría resumir en lo que denominamos "mundo real" [MARTÍN_DEL_BRÍO01]. Las máquinas construidas por el hombre, sin embargo, obtienen mejores resultados en tareas relacionadas con el cálculo y el razonamiento lógico. Una máquina puede jugar magníficas partidas de póker, pero le será casi imposible golpear un balón de fútbol en condiciones.

Esto es debido a lo que denominamos arquitectura de los computadores. La mayoría de las máquinas utilizadas en la actualidad están basadas en el llamado modelo de Von Neumann. Según Von Neumann, la clave para construir una máquina de propósito general es poder almacenar no sólo los datos y los resultados intermedios de una computación, sino también las instrucciones que definen dicho procedimiento de computación. En una máquina de propósito específico, el procedimiento puede ser parte de la máquina. Sin embargo, en una máquina de propósito general, cambiar las instrucciones tiene que ser tan fácil como cambiar los datos sobre los que actúan, codificar las instrucciones de forma numérica y guardarlas junto con los datos en la misma memoria, es decir, teóricamente se puede llevar a cabo cualquier tarea siempre y cuando se programe de la manera adecuada.

De esta forma, el computador está formado por cuatro subsistemas principales, unidos por un bus que permite la comunicación entre ellos (Figura 3.1):

- Memoria.
- Unidad Aritmético-Lógica (*Arithmetic Logic Unit*, ALU).
- Unidad de Control.
- Entrada y Salida, o simplemente E/S.

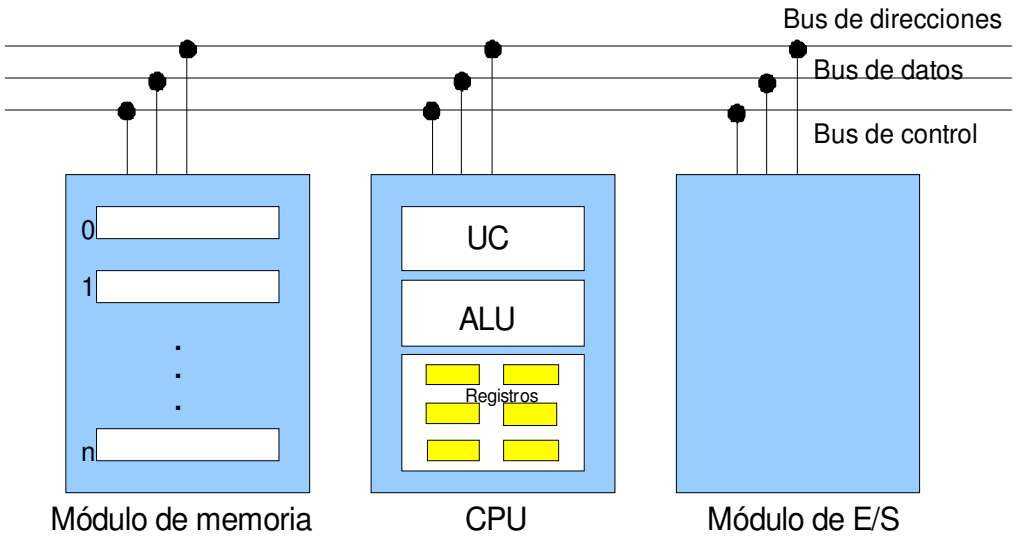


Figura 3.1. Modelo de arquitectura de computadores de Von Neumann.

Los programas se almacenan en la memoria durante su ejecución, llevándose a cabo las instrucciones de dicho programa de forma secuencial. La ALU y la Unidad de Control forman la Unidad de Procesamiento Central (*Central Processing Unit*, CPU) o procesador.

En la actualidad, la potencia de los llamados microprocesadores es ingente, por lo que se pueden llevar a cabo tareas de una gran complejidad de cálculo. Con un programa lo suficientemente preciso y una memoria suficiente bastaría para realizar cualquier tarea. Por ejemplo, si deseamos realizar una máquina que analice el mercado económico, es necesario reunir todos los conocimientos de los mejores economistas y volcarlos en un programa al que denominamos un sistema experto. Si, por el contrario, queremos realizar una predicción de fallos en el sistema eléctrico, hay que caracterizar estos sistemas y encontrar las condiciones en las que estos fallos se producen para realizar un programa que procese todos estos datos y conocimientos utilizando la potencia de cálculo y el razonamiento lógico, con el fin de llevarnos a la solución.

El problema llega cuando se han de resolver tareas en las que se debe tratar con enormes masas de datos y/o se precisan respuestas en tiempo real. En este caso, incluso grandes computadoras que operen en serie pueden tener serios problemas para alcanzar soluciones, o incluso no alcanzarlas. La información que proviene del mundo real es masiva, imprecisa y redundante, por lo que es evidente que es necesario el uso de computadores en paralelo. De esta forma, llegamos al elemento al que se quería emular cuando se crearon los ordenadores: el cerebro.

Es posible modelar el cerebro como un conjunto de pequeños microprocesadores elementales (lentos, simples y poco fiables), pero el número de estos procesadores es tan grande

(del orden de cien mil millones) y el número de interconexiones tan ingente, que su capacidad de procesamiento es enorme. Además, otra diferencia principal con los ordenadores es que las neuronas no son programadas, sino que aprenden del entorno, conectándose o inhibiendo las conexiones entre ellas de manera muy compleja. Aunque los procesadores actuales son cada vez más veloces y con mayor capacidad, su creciente utilización se debe fundamentalmente a los programas que se han desarrollado para ellos. En la actualidad, los ordenadores son un medio excelente para procesar símbolos, por lo cual, y a través de algoritmos, son capaces de simular determinados tipos de razonamiento y muchos de los procesos lógicos humanos. Sin embargo, y, aunque lo que hoy parece imposible de codificar podría no ser tan difícil en el futuro, hay un gran abismo entre el mundo real cotidiano y el frío y rígidamente acotado mundo de la computación.

Es probable que el algoritmo inicial contenga generalidades, ambigüedades y errores. Así, la tarea del programador es convertir el algoritmo en un programa (agregando detalles, superando los puntos conflictivos y corrigiendo los errores), de forma tal que el razonamiento humano cobre vida en la máquina. La ambigüedad, aunque tolerable en las conversaciones humanas, es una de las principales fuentes de errores en las computadoras. Por eso, los programadores deben tratar de anticiparse y responder a todas las combinaciones de órdenes y acciones llevadas a cabo por los usuarios, en cualquier condición.

Sin embargo, no hay atisbo de inteligencia: todo se reduce a una serie de caminos lógicos que la máquina sigue en función de lo que haga el usuario. En muchos casos, los métodos utilizados en la resolución de problemas mediante ordenadores se han basado tradicionalmente en sistemas expertos. En ellos, se codifica el comportamiento de los sistemas en formas de reglas de decisión. Esto tiene su mayor aplicación en problemas fácilmente modelables mediante reglas matemáticas, que son más cómodamente resolubles con la potencia de cálculo de los ordenadores. El mayor problema de los sistemas expertos viene dado por su incapacidad para abordar las "tareas del mundo real", en los que la información es masiva, imprecisa y distorsionada, con la consiguiente dificultad para obtener un buen modelo de la situación.

Para dar respuesta a estos inconvenientes surgen los sistemas basados en la denominada Inteligencia Computacional (*Computational Intelligence*, CI), cuyos principales exponentes son las Redes Neuronales Artificiales (*Artificial Neural Networks*, ANN) y la Lógica Borrosa (*Fuzzy Logic*, FL). Así mismo, otras técnicas como los Algoritmos Genéticos (*Genetic Algorithms*, GA) y la teoría de Conjuntos Aproximados (*Rough Sets*, RS) están también extendidas en menor medida.

Las redes neuronales están basadas en la introducción de sistemas de cálculo paralelo a imagen de lo que ocurre en el cerebro humano y en el procesamiento realizado por esos pequeños microcomputadores llamados neuronas. Como se ha dicho anteriormente, las redes de ordenadores y las redes neuronales tienen un gran paralelismo. Ambas se componen de unidades (ordenadores o neuronas), de capas y, por supuesto, lo que hace de ellas redes, las conexiones. Sin embargo, existen también profundas diferencias entre ambos tipos de redes. Lógicamente, el poder de computación de un ordenador es muchos órdenes mayor que el de una neurona pero las neuronas tienen la ventaja de su ingente número con lo cual su capacidad para trabajar en paralelo y obtener un gran rendimiento es mayor.

En cuanto a la lógica borrosa, surge como respuesta a la rigidez de la lógica binaria clásica. En ésta, no existe el término medio: o hace calor o hace frío. La lógica borrosa nos dará la

posibilidad de tener estados intermedios: fresco, templado, cálido… además de, mediante una serie de funciones, dar un grado de flexibilidad a estos calificativos: lo que para un sevillano puede ser fresco, puede ser templado para un berlinés.

Así mismo, la intención de la lógica borrosa es acercarse a la realidad. Si un bloque de cemento ha caído de una obra a nadie se le ocurre decir: "un objeto cúbico de 45 kg de peso está cayendo a una velocidad de 30 m/s" sino "¡cuidado!".

La lógica borrosa resulta pues útil para:

- Tratar la incertidumbre.

- Tratar la información precisa de que se dispone junto con la incertidumbre.

En este enfoque se dispone simultáneamente, por tanto, de información precisa y de incertidumbre. Utilizando lógica borrosa, se sacrifica cierta cantidad de precisión en función de la incertidumbre con la esperanza de obtener conclusiones que, aunque sean más vagas, sean más robustas [MATHWORKS02]. Lo normal es que los conceptos imprecisos se apoyen o basen en alguna medida precisa. Una posible medida precisa para, por ejemplo, el concepto difuso "soleado" podría ser "% de cobertura de nubes". El "% de cobertura de nubes" es absolutamente preciso.

Los algoritmos genéticos (GA) se llaman así porque se inspiran en la evolución biológica y su base genética. Estos algoritmos hacen evolucionar una población de individuos sometiéndola a acciones aleatorias semejantes a las que actúan en la evolución biológica (mutaciones y recombinaciones genéticas), así como también a una selección de acuerdo con algún criterio, en función del cual se decide cuáles son los individuos más adaptados, que sobreviven, y cuáles los menos aptos, que son descartados. Los GA son utilizados sobre todo en optimización y pueden ser combinados con ANN y FL con el fin de obtener los parámetros óptimos de ambos tipos de sistemas [LEU07].

Por último, los conjuntos aproximados (RS) pueden utilizarse como base teórica para algunos problemas de aprendizaje automático. Son particularmente útiles para la inducción de reglas y la selección de características [AN07].

3.2. Herramientas de Inteligencia Computacional.

Como se explica en el punto anterior, resulta evidente la necesidad de buscar un conjunto de metodologías que reflejen la notable capacidad del ser humano de tomar decisiones sensatas en un entorno de imprecisión e incertidumbre cuyo último fin es diseñar y construir máquinas con una inteligencia elevada. Este conjunto de metodologías, tal y como se explicaba en el apartado anterior, recibe el nombre de Inteligencia Computacional (*Soft Computing* o *Computacional Intelligence, CI*), destacando entre las herramientas principales de CI las ANN y la FL [ZADEH94].

La idea básica de estas técnicas es la de complementar los sistemas inteligentes tradicionales basados en la arquitectura de Von Neumann. En general, las ANN son capaces de descubrir automáticamente relaciones entrada-salida o rasgos característicos en función de datos empíricos, debido a su capacidad de aprendizaje a través de ejemplos. Los sistemas borrosos, por otro lado, permiten el empleo del conocimiento disponible para el desarrollo de sistemas inteligentes [ZADEH01]. Además, desde hace algún tiempo existe la tendencia de fusionar ambas técnicas con el fin de aprovechar las ventajas de ambas, lo que ha dado pie a los sistemas neuro-borrosos (*neuro-fuzzy systems*) [JANG92].

En el siguiente punto se describen las técnicas de CI centradas en la lógica borrosa, que es la que mejor se adecua a las condiciones del problema planteado, es decir, la necesidad de un agente inteligente que interprete las necesidades de un usuario que busca un conocimiento determinado dentro de entornos imprecisos y difusos, como, por ejemplo, es el entorno web.

3.2.1. Lógica Borrosa (Fuzzy Logic, FL).

La lógica borrosa nace en 1965 de la mano del profesor Lofti A. Zadeh [MARTÍN-DEL-BRÍO01; ZADEH65], básicamente con el fin de enfrentarse a algunos de los problemas que presenta la lógica binaria clásica. La lógica binaria es a veces inadecuada para la descripción del razonamiento humano, puesto que todo se define como 0 (Falso) o 1 (Verdadero). La lógica borrosa usa el intervalo completo comprendido entre 0 y 1 para caracterizar dicho razonamiento, por lo que se permiten los estados intermedios [JANTZEN98].

Además, el uso de la matemática borrosa permite razonar en términos lingüísticos como pequeño, medio o rápido en vez de en términos numéricos, de manera que las ambigüedades y contradicciones pueden ser manejadas cómodamente. Por ejemplo, en la lógica convencional, la afirmación (A y B) es únicamente verdad si A y B son verdad de manera individual (operación lógica AND). En la lógica borrosa, la verdad de (A y B) es la mínima verdad de ambas. Análogamente, la operación lógica A ó B (OR), consiste en el máximo de ambos valores [ZADEH65].

Dado que, como se ha comentado anteriormente, la lógica borrosa se adapta mejor a las necesidades del Agente Inteligente propuesto en esta tesis, nos extenderemos más en este punto que en el correspondiente a las ANN.

3.2.2. Conjuntos borrosos.

Uno de los conceptos lógicos más importantes de la lógica borrosa es el de conjunto borroso (*fuzzy set*). Los conjuntos lógicos no son más que un desarrollo posterior del concepto matemático de conjunto. El pionero en el estudio de los conjuntos fue el matemático alemán Georg Cantor, a finales del siglo XIX. Según su teoría, un conjunto es una colección de objetos que puede ser tratado como un todo. Un conjunto se especifica por sus miembros, caracterizando estos por completo a dicho conjunto [CANTOR06]. Un ejemplo de conjuntos puede ser observado en la Figura 3.2:

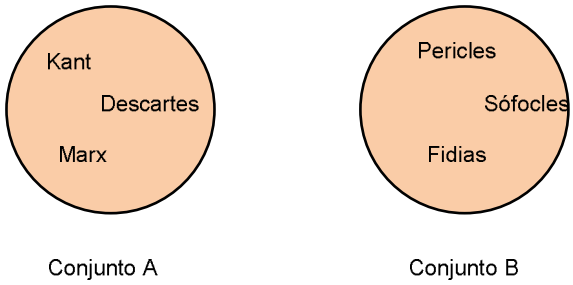


Figura 3.2. Ejemplos de conjuntos.

Es evidente que el conjunto A de la Figura 3.2 representa un conjunto de filósofos, mientras que el conjunto B representa un conjunto de personajes importantes de la Grecia clásica. Si escogiéramos a Aristóteles, este estaría presente en ambos conjuntos.

La lógica borrosa va, sin embargo, más allá de esta definición: los objetos pueden pertenecer a un conjunto borroso en diferentes grados, llamados grados de pertenencia (*membership degree*). La función que asigna estos valores entre 0 (no pertenencia a un conjunto) y 1 (pertenencia total a dicho conjunto) se denomina función de pertenencia (*membership function*). Si llamamos *U* a un conjunto de objetos, por ejemplo $U=\Re^n$, que se denomina universo de discurso, un conjunto borroso *F* en *U* queda caracterizado por una función de pertenencia μ_F que toma valores en el rango [0,1], es decir, $\mu_F : U \rightarrow [0,1]$, donde μ_F (u) representa el grado en el que $u \in U$ pertenece al conjunto borroso. Por tanto, se generaliza el concepto clásico de conjunto abrupto (*crisp set*), en el que este valor puede ser únicamente 0 (no pertenencia al conjunto) o 1 (pertenencia al conjunto), de manera que la función puede tomar también valores intermedios.

Tomemos, por ejemplo, un conjunto denominado *gente joven*. Es evidente que un bebé de un año pertenece a él con grado 1 (completamente), mientras que una persona de 100 años pertenece al conjunto con grado 0. Pero, ¿qué ocurre con una persona de 20, 30, 40 o 50 años? El grado de pertenencia describe un conjunto borroso: dicho grado oscila entre 0 y 1, no existiendo una base formal para determinarlo. La pertenencia de un hombre de 50 años al conjunto borroso *gente joven* dependerá del punto de vista de cada uno, es algo preciso pero subjetivo, y está en función del contexto [JANTZEN98].

Un ejemplo de esto puede verse en la Figura 3.3. Un observador fija el grado de pertenencia de una persona al conjunto borroso *gente joven* según su edad.

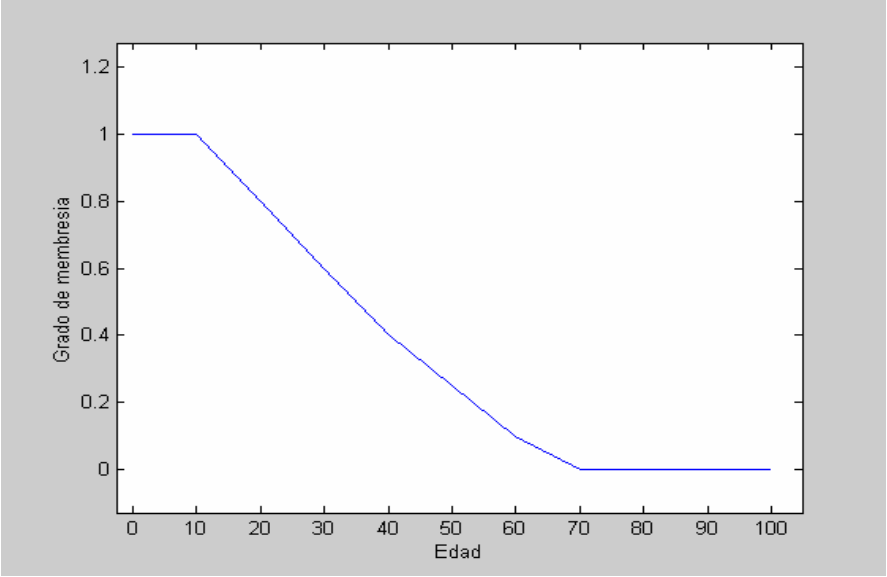


Figura 3.3. Ejemplo. Grado de pertenencia de una persona al conjunto borroso gente joven.

Es evidente que esta representación del conjunto borroso no es única. En este caso, está hecha desde la visión subjetiva de una persona de 33 años (el autor). Con toda probabilidad, esta visión sería muy diferente para una persona de 10 años u otra de 80. La gráfica dada en la Figura 3.3 representa una función de pertenencia.

La función de pertenencia o pertenencia de un conjunto borroso consiste en un conjunto de pares ordenados $F = \{(u, \mu_F (u)) / u \in U\}$ si la variable es discreta o una función continua si no lo es. Las funciones de pertenencia más utilizadas son:

- Función trapezoidal.
- Función singleton.
- Función triangular
- Función S.
- Función exponencial.
- Función tipo π.

A continuación se describen estas funciones de pertenencia.

La función de tipo trapezoidal se define por la expresión dada en la Ecuación 3.1:

$$S(u; a, b, c, d) = \begin{cases} 0 & u < a \\ \left(\frac{u-a}{b-a}\right) & a \le u \le b \\ 1 & b \le u \le c \\ \left(\frac{d-u}{d-c}\right) & c \le u \le d \\ 0 & u > d \end{cases} \quad (3.1)$$

Ecuación 3.1. Función trapezoidal.

Esta función resulta adecuada para modelar propiedades que comprenden un rango de valores. La función triangular es un caso particular en el que b = c. Para una función S abrupta, c = d = max (U), mientras que para que para una función singleton, a = b = c = d. La función singleton se utiliza habitualmente en sistemas borrosos para definir los conjuntos borrosos de las particiones de las variables de salida, pues permite simplificar los cálculos y requiere menos memoria para almacenar la base de reglas. En la Figura 3.4, se observan ejemplos de representación de todas estas funciones de pertenencia.

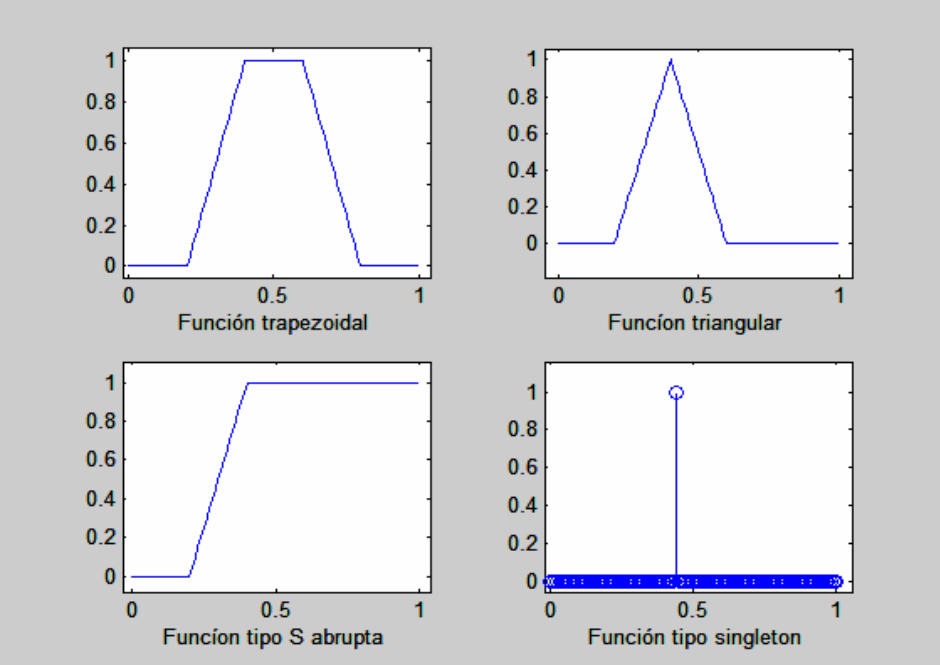


Figura 3.4. Distintos tipos de funciones de pertenencia (I).

Así mismo, también es habitual utilizar funciones S suaves (cuadráticas), muy común cuando hay que modelar propiedades como grande, mucho, positivo… y funciones tipo π,

adecuadas para conjuntos definidos en torno a un valor central con forma de campana. En la Figura 3.5 vemos un ejemplo de ambas.

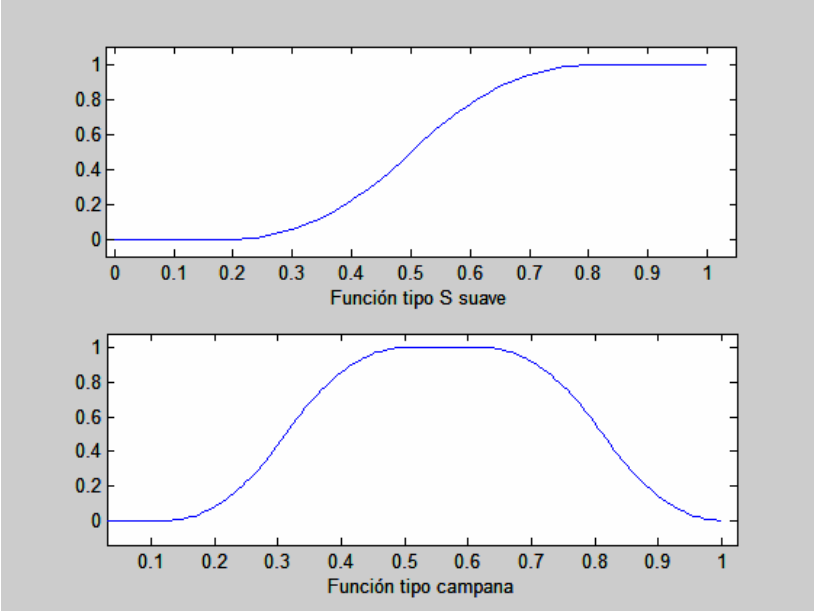


Figura 3.5. Distintos tipos de funciones de pertenencia (II).

3.2.3. Inferencia borrosa.

La lógica borrosa, al igual que la lógica booleana clásica, se basa en el concepto de inferencia. Antes de definir este concepto, sin embargo, debemos aclarar otros conceptos importantes en un sistema de lógica borrosa:

- Variable lingüística: es aquella que puede tomar un valor correspondiente a términos del lenguaje natural, como mucho, poco, algo, bastante, positivo, negativo, etc. Estas palabras desempeñan el papel de etiquetas en los conjuntos borrosos. De esta forma, en el ejemplo del punto anterior, la frase "es un chico bastante joven" es una variable lingüística. Dicha variable, además, puede tomar valores numéricos (por ejemplo, "Edad = 15 años"). En este mismo ejemplo, podríamos tomar todo el universo de discurso U (por ejemplo, entre 0 y 100 años) y asignar variables lingüísticas a todo el universo (por ejemplo, "muy joven", "bastante joven", "algo joven", "poco joven" y "nada joven", aunque también "niño", "joven", "adulto", "maduro" y "anciano"). En este caso, diremos que se han realizado subconjuntos borrosos.

- Operaciones borrosas: se pueden aplicar diversos operadores a los conjuntos borrosos. Se pueden definir tres operaciones básicas: complemento, unión e intersección. Estas operaciones se resumen en la Tabla 3.1.

Operación	Expresión
Complemento	$\mu_{A'}(x) = 1 - \mu_A(x)$
Unión	$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)]$
Intersección	$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]$

Tabla 3.1: Operaciones básicas de la lógica borrosa.

Por tanto, la lógica borrosa coincide con la lógica booleana en que ambas se ocupan del razonamiento con proposiciones pero, a diferencia de la lógica clásica, los valores de las proposiciones pueden tomar valores entre verdadero y falso.

Los conjuntos borrosos y los operadores borrosos son los sujetos y verbos de la lógica borrosa. Las reglas SI-ENTONCES (IF-THEN) son las afirmaciones condicionales que dan sentido a esta. Una regla SI-ENTONCES borrosa es del tipo "SI x es A, ENTONCES y es B, donde A y B son valores lingüísticos en los rangos (universos de discurso) X e Y, respectivamente. La parte "x es A" se denomina antecedente o premisa, mientras que la parte "y es B" se denomina consecuente o conclusión.

Un ejemplo de regla borrosa sería:

SI "el tamaño del coche" es "grande", ENTONCES "aparcar" es "difícil"

donde "grande" estaría representado por un número entre 0 y 1 (0 es diminuto y 1 enorme, por ejemplo) y "difícil" estaría representado por un conjunto borroso.

Estas reglas SI-ENTONCES pueden ser compuestas (si "x es A" y "z es B", entonces "y es C" y, además, admiten modificadores tales como bastante, casi, muy, etc. (si "x es bastante A", entonces "y es B").

3.2.4. Borrosificación (Fuzzyfication).

Para expresar un número en palabras, necesitamos una manera de traducir los valores numéricos de entrada en un conjunto borroso de descriptores lingüísticos: este es el llamado proceso de borrosificación [PANT04]. En matemática borrosa, esto es realizado por las funciones de pertenencia, de las que ya se habló en anteriores epígrafes. Estas funciones de pertenencia nos convierten un valor numérico en un valor lingüístico, como podemos ver en la Figura 3.6.

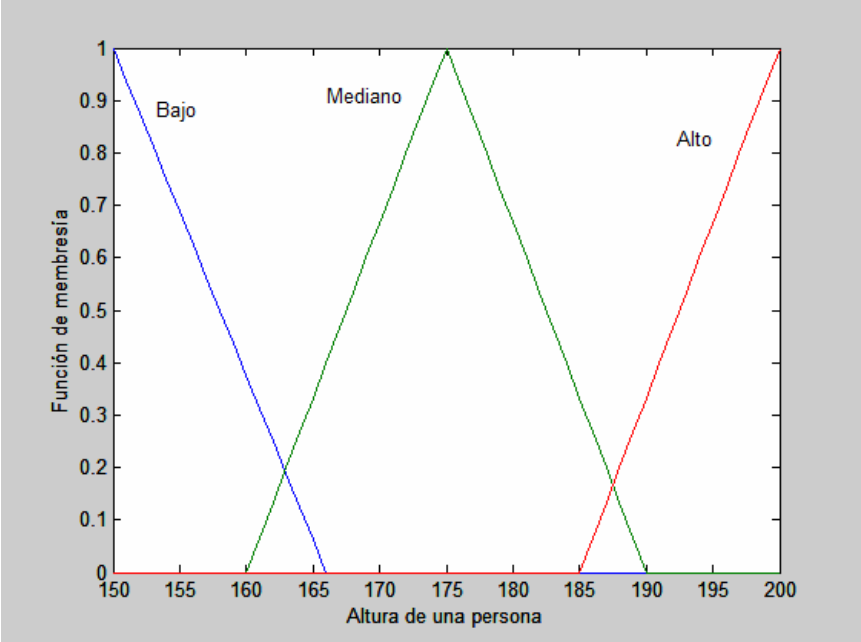


Figura 3.6. Ejemplo de funciones de pertenencia para la altura de un hombre.

El uso de funciones de pertenencia que se solapan, como en el ejemplo de la Figura 3.7, es extremadamente importante en los problemas de razonamiento borroso, ya que esto es lo que hace las fronteras entre conjuntos borrosos no estén delimitadas, pudiendo aprovechar así las ventajas de la lógica borrosa.

Por otra parte, la definición de estas funciones de pertenencia puede hacerse de distintas formas. Para ilustrarlas, baste el siguiente ejemplo.

Supongamos que estamos evaluando la altura de las personas y tenemos cinco conjuntos borrosos: MUY BAJO, BAJO, MEDIO, ALTO y MUY ALTO. ¿Qué peso tendría en cada conjunto, por ejemplo, la altura de Michael Jordan? Hay dos maneras principales de evaluarlo [JOHN95]:

- Por el método de frecuencia. Este método tiene en cuenta la respuesta afirmativa de un número de personas a una pregunta. Por ejemplo, ante la pregunta "¿Es Michael Jordan MUY ALTO?" la mayoría contestaríamos que si, pero posiblemente Pau Gasol contestaría que no. Si preguntamos a 100 personas y 99 nos dicen que sí, el índice de pertenencia de Michael Jordan (y el de todos los que midan lo mismo que él) al conjunto borroso MUY ALTO es de 0.99.

- Por estimación directa. Este método tiene la ventaja de ser mucho más sencillo de aplicar que el anterior, dado que depende de la percepción del autor del sistema. Por ejemplo, el autor de este trabajo podría asignarle a Michael Jordan un coeficiente de 0.98 en el conjunto borroso MUY ALTO. Podríamos pensar que este método tiene la desventaja de su

subjetividad, pero es precisamente esta la que dota a los sistemas de lógica borrosa de una gran flexibilidad, propia del pensamiento humano.

Aplicando uno u otro método se llega a la definición de las funciones de pertenencia que, a fin de cuentas, no son más que el resultado de aplicar estas técnicas a un rango continuo de valores.

Así mismo, en cuanto a la relación que se establece entre los puntos de entrada al sistema no borrosos y sus correspondientes conjuntos borrosos, se pueden aplicar varias estrategias de borrosificación:

- Borrosificación singleton: es el método más utilizado. Consiste en considerar los propios valores discretos como conjuntos borrosos.

- Borrosificación no singleton: se utilizan funciones exponenciales con forma de campana.

3.2.5. Desborrosificación (Defuzzyfication).

La desborrosificación es, como su nombre indica, el proceso opuesto a la borrosificación. En el método de implicación (reglas SI… ENTONCES), cada regla es aplicada al número dado por el antecedente, construyéndose un conjunto borroso V para el consecuente. Este conjunto V es la salida de un dispositivo de inferencia borroso y, generalmente, suele ser convertido en un valor no borroso $y \in V$. Para esta tarea se utilizan diversos métodos, entre los que se pueden destacar:

- Desborrosificador por máximo: y es el punto en el que el conjunto V alcanza su valor máximo.

- Desborrosificador por media de centros.

- Desborrosificador por centro de área o por el método del centroide. Es el más común de los métodos de desborrosificación y consiste en hallar el centro de gravedad de la superficie que se encuentra bajo la curva de la función correspondiente al conjunto borroso de salida.

En la Figura 3.7, se muestra un ejemplo de cómo se realiza la desborrosificación. El ejemplo está realizado con el toolbox de lógica borrosa de MATLAB. En la figura se puede apreciar que se han definido once reglas para tres entradas y la influencia que tiene cada una de las reglas en la salida final que, en este caso, ha sido calculada por el método del centroide.

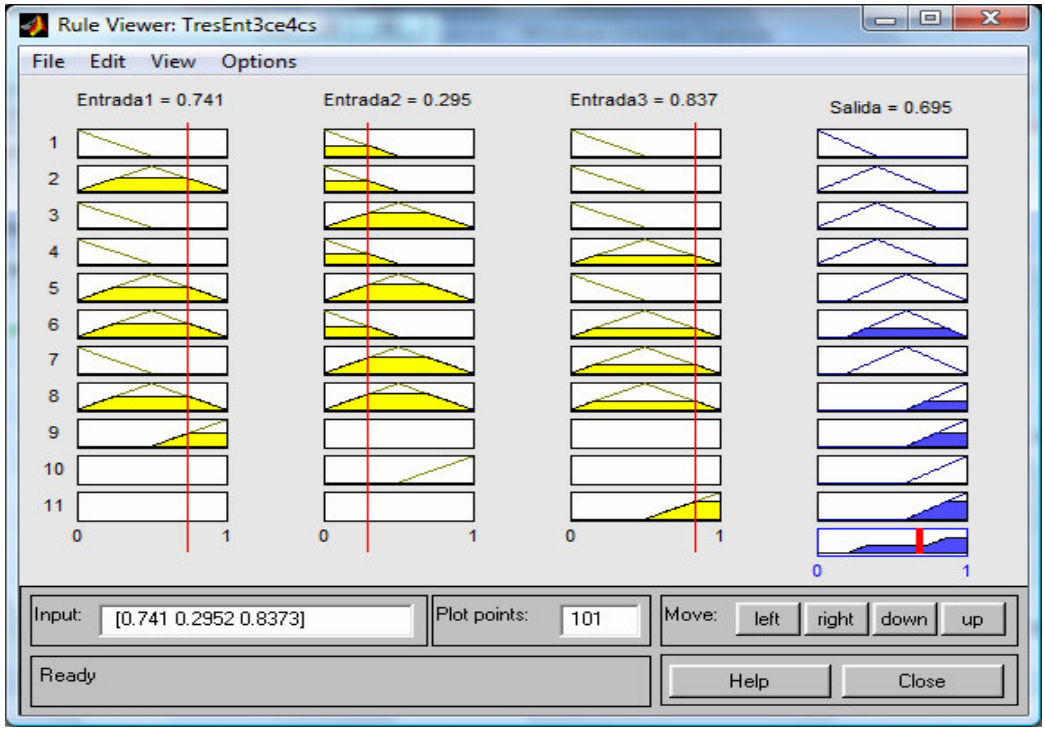


Figura 3.7. Ejemplo de desborrosificación en MATLAB.

3.3. Aplicaciones de la AI a la búsqueda de conocimiento.

Este apartado versa acerca de la aplicación de la AI a la búsqueda de conocimiento. Dado que esta tesis se centra principalmente en la búsqueda y extracción de información, se ha preferido proponer una sección aparte para las diversas aplicaciones a la búsqueda de conocimiento existentes en la bibliografía. Aunque hay algunas aplicaciones basadas en ANN, se hará especial hincapié en las aplicaciones de FL, tanto por el hecho de que utilizamos la FL para solucionar el problema planteado como por la gran cantidad de bibliografía existente al respecto.

Como se explicaba en la introducción a este capítulo, hay que tener en cuenta que el objetivo de un sistema de acceso al conocimiento es satisfacer las necesidades del usuario al acceder a los recursos de la información [LARSEN99]. Se comentaba así mismo que existen una serie de problemas en estos sistemas de acceso al conocimiento, a saber:

- La imprecisión en las necesidades de información de los usuarios.

- Los cambios en las necesidades de información a medida que el usuario recibe esta información durante la sesión de búsqueda.

- El posible desconocimiento de los usuarios acerca de sus necesidades exactas de información.

- La dificultad en expresar las necesidades de información al realizar una pregunta al sistema.

Los lenguajes de búsqueda habituales son demasiado limitados para expresar las necesidades del usuario, estando restringidos a criterios tajantes: conceptos y lógica (AND, OR, NOT). Un ejemplo bastante conocido de estos lenguajes es SQL (*Structured Query Language*, Lenguaje de Búsqueda Estructurado), soportado por RDBMS (*Relational Database Management Systems*, Sistemas de Gestión de Bases de Datos Relacionales) [JOHNSON08]. El problema es que los usuarios no expertos en la utilización de estos lenguajes pueden encontrarse con las siguientes trabas:

- El usuario debe conocer los términos (nombres de atributos, nombres de los temas, etc.) usados por el sistema de acceso al conocimiento.

- El usuario debe ser capaz de identificar los datos que contienen la información relevante de las que no la contienen.

- Los usuarios a menudo cometen errores en expresiones lógicas. Un ejemplo sería traducir "Quiero información sobre discos pop y rock" por "discos Y pop Y rock" cuando lo correcto sería "discos Y (pop O rock)".

- La graduación de la importancia al satisfacer criterios diferentes de búsqueda no es posible.

Además, hay que considerar la flexibilidad. Por ejemplo, ante la consulta "Quiero información sobre coches de segunda mano que satisfagan unos criterios C, y que el precio esté en la gama 10000 - 12000 euros", el criterio de precios puede no tener que significar que los coches con precios fuera de esta gama sean completamente descartables, pero sí que pueden ser menos interesantes. Por ejemplo, un coche que costara 8000 o 12500 euros todavía podría resultar interesante en mayor o menor medida, dependiendo del grado en el que se satisfagan los criterios C.

Otro aspecto a considerar en los sistemas de acceso al conocimiento es la forma que tiene el usuario de acceder a dicho conocimiento [LARSEN99]. Se pueden distinguir:

- Aproximación por formulación: El usuario formula una pregunta/consulta (*query*), que es entonces interpretada por el sistema.

- Aproximación inductiva: basada en la información acerca de los intereses del usuario con respecto a objetos en la base de información, el sistema deduce las necesidades de información que hay detrás de las preferencias del usuario. Está relacionada con la creación de perfiles de usuario.

Todos estos aspectos son tenidos en cuenta por los distintos sistemas de acceso a la información con ANN y FL que se presentan a continuación.

3.3.1. Aplicaciones de las ANN a la búsqueda de conocimiento.

Los primeros trabajos destacables que combinan redes neuronales con la recuperación de información son los realizados por R.K Belew y K.L. Kwok en 1989.

En el primero de ellos, Belew construye un sistema de recuperación adaptativo (*Adaptative Information Retrieval*, AIR). El objetivo del sistema es la recuperación de los documentos que más se parecen a aquellos que busca el usuario. Belew usa para ello una red neuronal de tres capas, correspondiendo cada capa a distintas "características": palabras clave, documentos y autores. Cada uno de estos representa un nodo de la correspondiente capa, como se puede ver en la Figura 3.8. [BELEW89].

En una línea parecida está Kwok, que también construye una red neuronal de tres capas, correspondiendo cada capa a las búsquedas, los términos índice y los documentos, respectivamente. Cada documento es evaluado en relación con los la activación de pesos por encima de un determinado umbral [KWOK89].

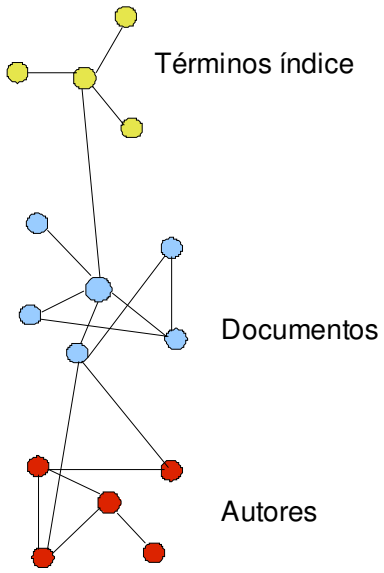


Figura 3.8. Asociaciones entre nodos en AIR.

Posteriores avances en la materia se han centrado en la posible reducción de nodos por asociación de palabras, la utilización de redes neuro-semánticas, el uso de tesauros o la modificación de características para obtener mejores resultados [SALTON88, HARMAN91].

Cabe también mencionar el uso de mapas de Kohonen para la recuperación de información mediante mapas semánticos autoorganizados [POPOVIC92].

Un ejemplo interesante es la utilización de redes neuronales por parte de Ruiz y Srinivasan para la asignación de frases MeSH (ver apartado 2.5.2 de esta tesis) basada en la frecuencia de término de palabras simples del título y el *abstract* [RUIZ02]. Para ello se basaron en un corpus de la colección de documentos Medline. Medline es una colección de documentos ampliamente utilizada en IR, en primer lugar por su ingente tamaño (unos 18 millones de documentos registrados) y, sobre todo, por la posibilidad de utilizar el sistema MeSH para recuperación de información ya existente [GÁLVEZ08]. Cada documento de esta colección incluye título, autores, información de citaciones, *abstract* y un conjunto de términos MeSH asignado manualmente. El proceso, en el caso de este trabajo, consistió en señalar las palabras de los títulos y *abstracts*, eliminar las palabras más comunes usando una *stop list*, realizar un enraizado de palabras y, por último, computar la frecuencia de cada raíz en cada documento.

Más de 12000 palabras-raíz y 4000 frases MeSH fueron encontradas. Dada la ingente cantidad de neuronas que serían necesarias, se utilizó un umbral mínimo de frecuencia para poder pertenecer a la colección, con lo que se redujeron mucho estos términos. Aún así, la red neuronal construida tenía 1016 nodos de entrada (palabras-raíz) y 180 nodos de salida (términos MeSH). A esto hay que sumar 540 nodos de una capa intermedia .El problema principal está en el tiempo de entrenamiento (más de un día), con el consiguiente problema cada vez que haya que actualizar la red. A pesar de todo, los resultados obtenidos fueron buenos.

Otra alternativa que utiliza las ANN, pero combinada con la lógica borrosa, la constituyen los sistemas neuro-borrosos (ANFIS, Artificial Neural Fuzzy Inference Systems). Un ejemplo de la utilización de estos sistemas para la búsqueda de conocimiento es el trabajo de Mateo et al., centrado en la localización de servicios para usuarios de móviles basados en la posición [MATEO06]. En el modelo de conjuntos borrosos, la asignación de clases atribuye a cada elemento un grado de pertenencia en el intervalo [0,1] para cada conjunto definido. Este grado de pertenencia se corresponde con el grado en el cual un elemento es similar al concepto o al prototipo representado por ese conjunto. El clasificador fuzzy k-means (FCM) usa un procedimiento iterativo que comienza con una asignación inicial arbitraria de los objetos clasificando estos en grupos o *clusters k*. La utilización de una técnica como FCM para *clustering* hace posible recuperar la información relevante de un entorno distribuido. Además, la lógica borrosa se combina con redes neuronales con la finalidad de encontrar la información exacta. El funcionamiento del sistema neuro-borroso se basa en el uso de un sistema borroso para representar el conocimiento en una manera interpretable y en la utilización de la capacidad de aprendizaje de las redes neuronales para determinar los valores de pertenencia de los conjuntos lógicos.

3.3.2. Aplicaciones de la FL a la búsqueda de conocimiento.

Como se ha apuntado en ocasiones anteriores, los motores de búsqueda, portales y técnicas de recuperación de documentos clásicas ofrecen una especie de búsqueda por palabras claves en la web. El resultado puede ser el hallazgo de miles de aciertos, muchos de ellos sin valor o, lo

que es peor, que no sepamos si son correctos o aplicables. Hay varios motivos para esta situación [HAASE02]:

- El usuario proporciona alguna sintaxis de una pregunta, no un significado (que el sistema tiene que adivinar).

- El usuario es anónimo (el sistema no conoce su estructura mental, es incapaz de deducir requisitos previos desconocidos).

- Las páginas de inicio (*homepages*) están muchas veces más llenas de anuncios publicitarios que de resúmenes.

- Existe mucha información secundaria.

El objetivo de un sistema de recuperación de datos es el de evaluar el grado de importancia de los documentos disponibles en lo que concierne a las preguntas de un usuario y recuperar los documentos con un alto grado de satisfacción para este. Para conseguir un buen funcionamiento, la respuesta al usuario debe ser capaz de responder de manera apropiada a lo que el usuario solicita. Muchos de los sistemas de recuperación de datos estaban basados tradicionalmente en el modelo lógico booleano. Este modelo asume que las preguntas de un usuario pueden ser caracterizadas con precisión por los términos índice. Sin embargo, esta suposición es inadecuada debido al hecho que las preguntas del usuario pueden adolecer de falta de claridad. La razón de esta falta de claridad contenida en las preguntas del usuario es que el usuario puede no conocer mucho sobre el objeto que busca o puede no estar familiarizado con el sistema de recuperación de datos. Para poder manejar la imprecisión y la incertidumbre en la representación de conceptos y palabras en el mundo real, la mayor parte de los modelos de aprendizaje automático han sido relacionados con la lógica borrosa. Estos modelos borrosos superan los problemas que pueden crear las separaciones abruptas entre los valores de los atributos [XIE05], proporcionando una transición suave y una buena precisión en relación con atributos continuos.

Como se comentó en el capítulo 2 de esta tesis, existen varios enfoques a la hora de manejar la información en un sistema de IR. Uno de ellos está basado en el Modelo de Espacio Vectorial (VSM). El otro está relacionado con los conceptos de ontología y web semántica. En la Figura 3.9, se puede ver un esquema conceptual sobre las aplicaciones de FL a IR.

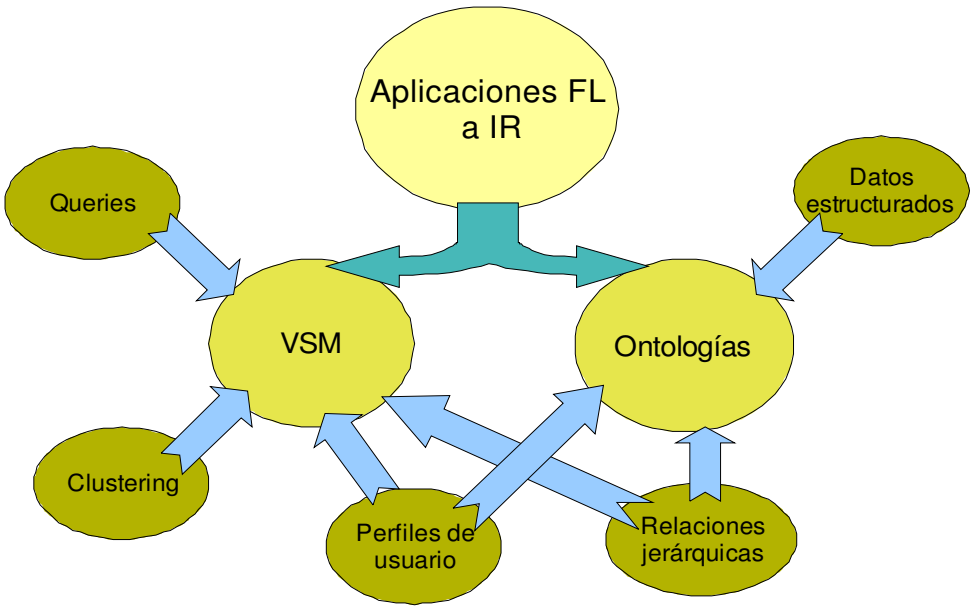


Figura 3.9. Aplicaciones de FL a IR.

A continuación, se analizan ambos enfoques para la gestión de la información en sistemas de IR y se exponen algunos ejemplos de aplicación.

FL basada en el VSM

Dentro del primer grupo, basado en el VSM, cobran importancia conceptos como el de *query* (búsqueda o consulta) y el de *clustering* (agrupamiento).

Un ejemplo del uso de consultas viene dado en [HAASE02]. WIP-AUSTRIA es un portal de Internet desarrollado por centros de investigación austriacos. Da acceso a bases de datos, documentos electrónicos e impresos, así como a consultas y servicios. La base de datos WIP está basada en objetos de conocimiento: el contenido de un documento o el servicio está definido por una matriz bidimensional (mapa de conocimiento) basado en un esquema de clasificación decimal y un juego de atributos. Se accede a WIP mediante consultas (*queries*), siendo estas interpretadas como variables lingüísticas que se pueden usar para construir un "mapa de preguntas". El emparejamiento de patrones junto a un proceso de razonamiento de lógica borrosa conduce a buenos resultados, encontrándose respuestas entre los candidatos más probables. Así mismo, en [CORDÓN04] se utilizan las denominadas búsquedas persistentes (*persistent queries*), que son una clase específica de preguntas utilizadas en sistemas IR para representar la necesidad de información a largo plazo de un usuario. Estas búsquedas pueden presentar muchas estructuras diferentes, siendo la llamada "bolsa de palabras" (bag-of-words), ya comentada en la sección 2.4.2 de esta tesis, la más comúnmente usada. El modelo bag-of-words es una suposición simplificada usada en el tratamiento del lenguaje natural y la recuperación de documentos. En este modelo, un texto es representado como una colección desordenada de palabras, desatendiendo la gramática e incluso el orden de las palabras. En el

trabajo de Cordón et al. se intenta llegar a las búsquedas persistentes con una estructura más representativa para la búsqueda de textos. Para ellos se utilizan técnicas de *soft computing*: la lógica borrosa se utiliza para inferencia y representación mejorando las técnicas booleanas existentes.

En una línea parecida se encuentran Mercier et al. [MERCIER05], basándose en la idea de que mientras más cercanos están en un documento los términos buscados, más relevantes son los términos en dicho documento. Con esta idea, se experimenta un método IR basado en el grado de proximidad borroso de las ocurrencias del término buscado en un documento para calcular su importancia en dicha búsqueda. La proximidad de término borrosa está controlada por una "función de influencia". Dados un término de la búsqueda y un documento, la función de influencia asocia a cada posición en el texto un valor dependiente de la distancia a la ocurrencia más cercana del mismo término.

Por otra parte, el *clustering* es empleado en otros trabajos como el de Friedman et al. [FRIEDMAN04]. Muchos de los métodos existentes para el *clustering* de documentos están basados en el modelo de espacio vectorial clásico, que representa cada documento por un vector de términos clave o frases claves de tamaño fijo. En colecciones de documentos grandes y diversas como la World Wide Web, esta aproximación sufre de una enorme sobrecarga computacional, ya que el tamaño constante de los vectores de término es igual al número total de términos índice en todos los documentos. Se propone una aproximación borrosa al *clustering* de documentos en el que estos son representados por vectores de tamaño variable. Dado que un documento generalmente contiene sólo un pequeño subconjunto de los términos del sistema, la matriz de términos asociada al documento es habitualmente muy poco densa con casi el 99 % de entradas iguales a cero.

Se considera un conjunto S de n vectores con longitudes variables y una longitud máxima m. Cada vector representa un todo o una parte de un documento web con k componentes denominados "frases clave" (*key phrases*), donde $1 \leq k \leq m$. El número total de las distintas "frases clave" en todos los vectores es n_o y para un vector arbitrario $x = (t_1, t_2, \ldots, t_k)^T$ se asigna a cada t_i un valor w_i (peso), que puede ser calculado por un modelo de indexado basado en la frecuencia o por un algoritmo de extracción de frases clave. El objetivo es dividir S en varios *clusters*, tantos como sea necesario. El número final de *clusters* está determinado por el algoritmo y la naturaleza de S. La única exigencia es que cada *cluster* incluirá vectores similares entre sí, mientras que los que pertenecen a *clusters* distintos, serán radicalmente diferentes. Se han desarrollado algoritmos de *clustering*, basados en los pesos locales y la medida de semejanza del coseno, comentados en el capítulo 2 de la tesis. Al utilizar el método del coseno para *clustering*, es necesario definir un cluster central c, que es la suma normalizada de todos los vectores del cluster C. Entonces se normaliza el vector de entrada y se calcula su producto interno con el cluster central, que considera únicamente los términos índice que aparecen en ambos vectores. Este producto interno mide el coseno del ángulo entre dos vectores y tiene un valor entre 0 y 1. La semejanza máxima ocurre cuando los vectores son idénticos, produciendo un producto interno igual a 1. El otro caso extremo ocurre cuando los vectores son ortogonales y el producto interno es cero.

Otra aplicación interesante desde el punto de vista de la investigación en este campo es la propuesta de Subasic y Huettner, que fusiona el procesado del lenguaje natural y las técnicas de lógica borrosa para analizar el contenido afectivo de un texto [SUBASIC01]. Un aspecto principal en este sistema es el de "conjunto de afecto", un conjunto de categorías borrosas

semánticas basadas en características positivas y negativas y que servirá para agrupar los documentos o textos en diversos *clusters* basados en las características afectivas de dichos textos.

En [HORNG05] se presenta un método jerárquico aglomerativo de clustering, que salva la desventaja del método tradicional, en el que un documento no puede pertenecer a múltiples clusters al mismo tiempo. Existen algoritmos jerárquicos de clustering: El clustering jerárquico construye (*clustering* aglomerativo, *agglomerative*), o rompe (*clustering* divisivo, *divisive*) una jerarquía de clusters. La representación tradicional de esta jerarquía es un árbol (llamado dendograma) con elementos individuales en un extremo y un único cluster que contiene cada elemento en el otro. Los algoritmos aglomerativos comienzan en las hojas del árbol, mientras que algoritmos divisivos comienzan en la raíz (Figura 3.10).

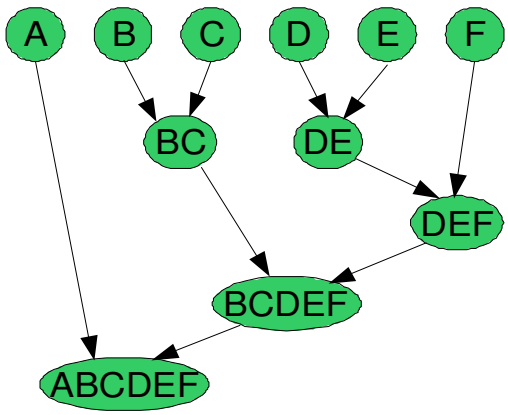


Figura 3.10. Algoritmo de clustering jerárquico aglomerativo.

Tanto el método de clustering borroso jerárquico aglomerativo propuesto como el método k-means tienen la ventaja de su flexibilidad en tanto que permiten que un documento pertenezca a múltiples clusters al mismo tiempo. La diferencia entre ambos métodos es que el método k-means tiene que predefinir el número de clusters mientras que el método propuesto usa un "valor de umbral de semejanza" α y un "valor de umbral de diferencia" λ para realizar un tratamiento automático del proceso de clustering. Como el método jerárquico aglomerativo tradicional de clustering es un método tajante (no borroso) de clustering, donde cada documento sólo puede pertenecer a exactamente un grupo, no existe flexibilidad.

Una pregunta de usuario puede ser representada por un vector de búsqueda *q* de la siguiente manera:

$$q = [w_{q1}, w_{q2}, \ldots, w_{qs}];$$

donde cada w_{qi} denota el peso del término t_i en el documento a recuperar. Aquí, $w_{qi} \in [0,1]$ (0 no contiene el término, 1 tiene una importancia máxima) o vale "-" (no se considera en el proceso – *stop word* -). El artículo presenta un modificador borroso de pesos. Por ejemplo, si

asumimos que el usuario quiere recuperar documentos sobre el tema "tratamiento del lenguaje natural" y tenemos en cuenta que 13 documentos entre los 247 informes de investigación del corpus utilizado son relevantes. Asumimos que los pesos de los términos "natural", "lenguaje", y "procesado" en la pregunta del usuario son 0.9, 0.9, y 0.8, respectivamente. El método de modificación de búsquedas emplea las reglas lógicas borrosas generadas basadas en la "Categoría de Asociación Positiva" para modificar la búsqueda original. Aplicando el método de modificación de pregunta, la pregunta q1 del usuario original puede ser transformada en la pregunta q1+ que, por un lado, tiene los mismos pesos de los términos "natural", "lenguaje", y "procesado" y, por otro lado, tiene un término adicional "palabra" con un peso 0.46. Aplicando el método de modificación de pregunta propuesto, la pregunta del usuario original puede ser transformada en la pregunta q1* que también tiene los mismos pesos de términos "natural", "lenguaje", y "procesado" y, además, cuatro términos adicionales: "palabra", "dictionari" (raíz de "diccionario"), "corpu" (la raíz de "corpus") y "discurso" con los pesos 0.46, 0.44, 0.39, y 0.37, respectivamente. En el experimento, se comparan las tasas de precisión y de memoria para los primeros *p* documentos recuperados con la pregunta del usuario original q_i, y las preguntas del usuario modificadas (q_i+ y q_i*), aunque las definiciones de precisión y de memoria son más abiertas, ya que se considera que devuelve *p* documentos, de los que *x* están relacionados.

Ríos et al. presentan un acercamiento conceptual para mejorar el contenido de un sitio Web [RÍOS06]. Como se explicó en la sección 2.2.3 de esta tesis, las técnicas WUM estudian los intereses de los usuarios a la hora de navegar por el sitio web con el fin de refinar la búsqueda de la información con arreglo a sus características. Muchas investigaciones en el área de la IR olvidan la información semántica que también poseen las páginas web. Por tanto, se propone combinar el Descubrimiento de Conocimiento en Textos basado en el Concepto (*Concept-Based Knowledge Discovery in Text)* con las sesiones de los visitantes del sitio para realizar una tarea de personalización. De esta forma, es posible obtener la información objetivo de los usuarios que navegan por un sitio web. Es más, es posible dar recomendaciones para una mejor navegación y ayudar a los gestores de los sitios web a mejorar el contenido. Además, se prueba esta idea sobre un sitio Web real para demostrar su eficacia. La solución está basada en la idea de unir la relación entre los conceptos y los documentos que puede ser representada como una composición borrosa, mostrada en la Ecuación 3.2. El operador ◦ representa el operador composicional de inferencia [ZADEH65].

$$[Conceptos \times Palabras] = [Conceptos \times Términos] \circ [Términos \times Palabras]$$

Ecuación 3.2. Relación entre conceptos y documentos.

De esta forma, se llaman "términos" a las palabras especiales que representan un concepto y "palabras" a cualquier palabra en un documento como una página web. Para aplicar la expresión de la Ecuación 3.2 se define una lista de conceptos y de términos que representan estos conceptos. Sin embargo, todavía habría que establecer los valores de pertenencia (pesos) para esta relación. Se usan conocimientos expertos para definirlos (método directo con un experto) y, además, un modelo simple que utiliza la frecuencia de palabras relativa sobre cada documento que define la segunda relación borrosa [Términos × Palabras]. Los documentos fueron preprocesados para eliminar los códigos HTML y JavaScript y se utilizó una lista de *stop words* para eliminar las palabras que no son importantes, intentando mantener sustantivos, adjetivos y verbos. Así mismo, no se utilizó el enraizado (*stemming*). Al final, se obtiene la matriz [Conceptos × Palabras], en la que cada fila es un concepto y cada columna es una página

web. Cada valor de la matriz representa la posibilidad de que un concepto sea representado en una página web.

Por otro lado, hay que transformar las páginas web a un modo más útil, para lo que se usa el VSM para representar cada documento como un vector de palabras usando el método TF-IDF para establecer el peso de cada palabra en cada documento. Por otra parte, es necesario que preprocesar las entradas (*logs*) a los servidores de web para entender las sesiones de los visitantes. Después de este paso, se dispone de las páginas y el tiempo pasado por los visitantes en cada documento. Después de la etapa de procesado previo hay que aplicar un proceso de generalización. Se utiliza un Mapa Autoorganizado (SOFM) puesto que se trata de un algoritmo no supervisado, lo que quiere decir que no se necesita conocer de antemano cuántos grupos (*clusters*) hay. Finalmente, se obtienen los grupos que están relacionados por su significado conceptual. Por lo tanto, se consigue una clasificación conceptual de los documentos sobre la sesión de cada visitante, lo que, en última instancia, ayuda a los *webmasters* a mejorar la utilidad de sus sitios web.

El uso de perfiles de usuario es pues un paradigma de uso común en investigaciones recientes. Moradi et al. proponen la personalización de los resultados de sistemas de recuperación de datos, siendo su objetivo atender mejor a los usuarios basándose en sus perfiles [MORADI08]. Los objetivos de la recuperación de contenido personalizada es mejorar el proceso de recuperación teniendo en cuenta los intereses particulares de usuarios individuales. Con este método, tanto páginas como perfiles de usuario se muestran como redes de concepto borrosas ampliadas. Una red de conceptos incluye nodos y enlaces dirigidos, donde cada nodo representa un concepto o un documento; cada enlace dirigido une dos conceptos o dirige de un concepto C_i a un documento d_i, estando etiquetado con un valor entre cero y uno (Figura 3.11).

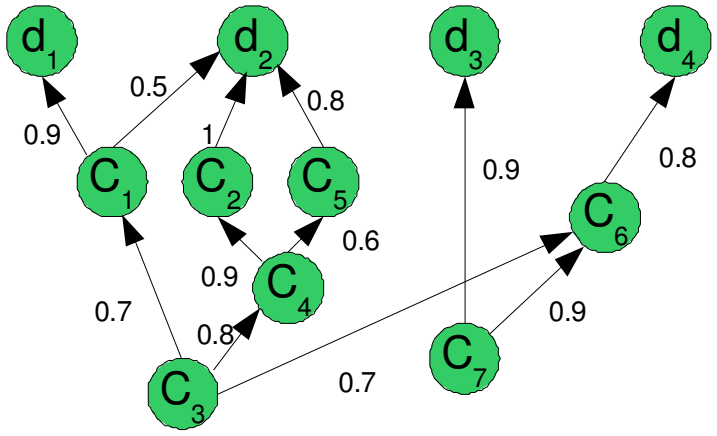


Figura 3.11. Red de conceptos borrosa.

Las redes de concepto borrosas ampliadas son más generales. Hay cuatro clases de relaciones borrosas entre conceptos:

- Asociación borrosa positiva: La asociación borrosa positiva relaciona los conceptos que en algunos contextos tienen un significado borroso similar. (p.e., persona → individuo).

- Asociación borrosa negativa: relaciona los conceptos que son antónimos borrosos complementarios, borrosos incompatibles o borrosos. (p.ej., macho → hembra, grande → pequeño).

- Generalización borrosa: generalización borrosa de otro concepto si este incluye dicho concepto (p.e., vehículo → coche).

- Especialización borrosa: es la inversa de la relación de generalización borrosa. Es decir, un concepto es considerado como una especificación borrosa de otro concepto si es una parte o una clase de este (p.e., coche → vehículo).

La figura 3.15 muestra el ejemplo de una red de concepto borrosa ampliada donde C_1, C_2..., C_7 son conceptos y d_1, d_2, d_3 y d_4 son documentos. De esta forma podemos ver que el documento d_2 posee 50 % del concepto c_1, el 80 % del concepto c_5 y es 100 % complementario con el concepto c_2. (P=Positive; N=Negative; G=Generalized; S=Specialized; Z=Zero - sin relación definida por parte del experto -).

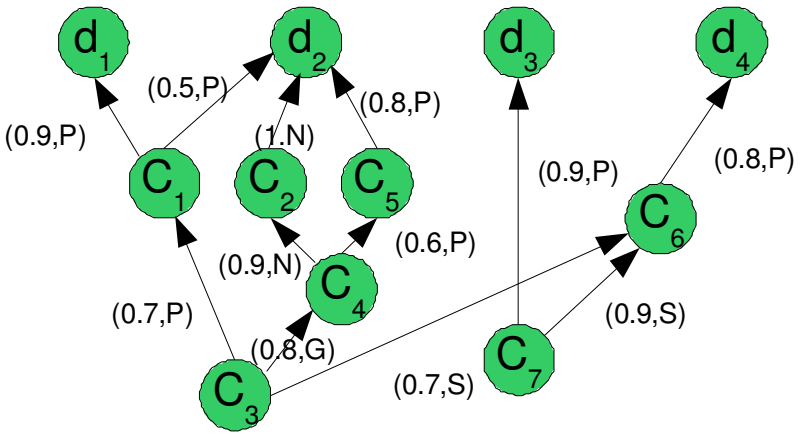


Figura 3.12. Red de conceptos borrosa.

Supongamos que hay dos usuarios A y B. Los intereses del usuario A incluyen la informática mientras que los intereses del usuario B están más centrados en la biología. Supongamos que buscan la misma palabra "java" sobre el sistema de recuperación de datos: para el usuario A, queremos recuperar documentos más relacionados con la informática mientras que para el usuario B, queremos recuperar documentos más relacionados con la biología. Básicamente, lo que se hace es modificar los pesos en la matriz con el fin de que el resultado dependa de cada perfil de usuario.

Aunque la mayor aplicación de la lógica borrosa para IR es la recuperación de textos y documentos, también existen otras aplicaciones, como la Recuperación de Imágenes Basada en Contenidos (*Content Based Image Retrieval*, CBIR) [ZHANG03]. Dada una imagen, el sistema

primero la incluye en un grupo de regiones basadas en un espacio de características que tiene en cuenta las características de color y textura usando un algoritmo k-means modificado, el cual consiste en agrupar los objetos de una colección en k grupos [DING04]. En cada región segmentada, aplicamos la lógica borrosa para definir los rasgos de color, textura y forma, respectivamente. La motivación para incorporar la lógica borrosa en los rasgos es localizar la imprecisión de una representación de características típica para mejorar la robustez y la eficacia del esquema de indexado, de manera que permita un cierto grado de variación de los valores de dichas características. Se usa un histograma de colores para una región segmentada. De esta forma, se asume que cualquier color *c* es un conjunto borroso, calculándose el parecido de cualquier color *c'* con el color *c*, mediante una función gaussiana que mide la distancia de color LAB (espacio de color, el más utilizado es RGB) [HARDEBERG01]. Este modelo de colores borrosos permite tener en cuenta la típica imprecisión de la característica de color, "cubriendo" sus colores vecinos en el espacio de color. Una función similar se usa para las características de textura y forma. La similitud entre dos imágenes (DI) está definida en la Ecuación 3.3.

$$DI = \frac{\sum_{i=1}^{M} w1i \cdot Min_{i=1}^{M}\{DIST(j,i)\} + \sum_{j=1}^{N} w2i \cdot Min_{i=1}^{N}\{DIST(i,j)\}}{2}$$

Ecuación 3.3. CBIR. Similitud entre dos imágenes

El peso para cada región al principio se fija en relación al área de la región de la imagen correspondiente. Estos pesos se actualizan adaptativamente mediante una realimentación de usuario.

FL basada en ontologías

La otra posible dirección en la que se puede utilizar la lógica borrosa para la búsqueda de conocimiento es el procesado de búsquedas guiado mediante ontologías. Este modelo de procesado puede ayudar a los usuarios a tener acceso a la información almacenada dentro de documentos de texto no estructurados o semiestructurados con cierta eficacia.

La ontología representa el conocimiento de un dominio en una forma estructurada y, cada vez más, está siendo aceptada como una tecnología clave cuando los conceptos más importantes y sus relaciones mutuas son almacenados para proporcionar un entendimiento compartido y común de un dominio a través de ciertos usos. La ontología es una conceptualización de un dominio en un formato humano comprensible, pero legible por una máquina. Consiste en clases, atributos, relaciones y axiomas y también en entidades [GUARINO95]. Ya que la ontología describe un dominio de interés de un modo inequívoco, los esquemas de IE basados en ontología pueden ayudar en la eliminación de ambigüedades presentes en textos en lenguaje natural y, además, realizar un análisis semántico de textos eficaz. Sin embargo, el procesado de textos basado en la ontología no ha sido aún explotado completamente.

En [ABULAISH05], se propone un mecanismo que utiliza la minería de texto con un conjunto de conceptos ontológicos extrayendo relaciones borrosas a través de dicha minería de

textos. Los valores de pertenencia de las relaciones son funciones de frecuencia de co-ocurrencia de conceptos y relaciones. Se trabajó sobre el corpus GENIA y se mostró como las relaciones borrosas pueden también usarse para la extracción de información guiada de documentos Medline. Debido a la falta de colecciones normalizadas para la evaluación de los sistemas de minería, se ha creado el corpus GENIA, consistente en 2000 resúmenes de la base de datos Medline, con más de 400000 palabras y alrededor de 100000 anotaciones, que han sido codificadas de forma manual para los términos biológicos [GÁLVEZ08].

Se propone un sistema de IE inteligente con dos objetivos principales. En primer lugar, el sistema usa una ontología biológica existente (GENIA en este caso) para la extracción de información de documentos de texto. En segundo lugar, la información extraída se usa para evolucionar desde las estructuras de ontología inflexibles ya existentes a estructuras de ontología borrosas, que pueden asumir relaciones interconceptuales con diferentes grados de relación. El sistema de minería de textos a base de ontologías emplea el llamado reconocimiento de entidades en conjunción con técnicas NLP para encontrar relaciones borrosas que unen conceptos ontológicos a partir de documentos de texto.

La arquitectura de sistema propuesta tiene cinco componentes principales:

- Procesador de Documentos: este módulo tiene como entradas documentos de texto etiquetados basados en una ontología y extrae frases de ellos como salidas. El módulo usa un *Part-of-Speech Tagger* (POS o POST, etiquetador de discurso), utilizado para evitar la posible ambigüedad en las distintas categorías en las que se agrupan las palabras. Básicamente es parecido a un analizador gramatical, asignando una etiqueta gramatical a cada palabra de una oración. Cada documento se convierte en un "árbol de oraciones". Este módulo no usa etiquetas de ontología.

- Extractor de Verbos Relacional: este módulo usa las etiquetas de ontología del documento de entrada y también la estructura arborescente generada por el Procesador de Documentos. Implementa los principios de la minería de textos basándose en técnicas NLP para encontrar todos los verbos relacionados y sus variantes morfológicas en el documento.

- Extractor de Asociación de Etiquetas Biológicas: este módulo explota las relaciones de taxonomía en un dominio ontológico en conjunción con la estructura de árbol para identificar asociaciones que ocurran frecuentemente entre los diferentes tipos de etiquetas de entidades biológicas.

- Extractor de Relaciones Biológicas Borroso: este módulo usa los verbos relacionados y sus variantes morfológicas extraídas por el Extractor de Verbos Relacional, y las parejas entidad-etiqueta extraídos por el Extractor de Asociación de Etiquetas Biológicas, con el fin de identificar relaciones biológicas borrosas factibles, considerando relaciones factibles a aquellas que tienen un valor mayor que un umbral especificado.

- Editor de ontología: este módulo se usa para incorporar las relaciones biológicas extraídas y sus valores de pertenencia en el dominio ontológico existente.

Así el sistema obtiene ternas de la forma <E1, V, E2> de una manera muy eficiente. Podemos ver un ejemplo de las conclusiones a las que llega el sistema en la Tabla 3.2.

Entidad 1 (E1)	Verbo Relacional (V)	Entidad 2 (E2)	Valor de pertenencia
Mono_cell	Induces in	Protein_molecule	1.00
Protein_molecule	Inhibit in	Protein_molecule	1.00
Protein_molecule	Expressed	Cell_type	1.00
Protein_molecule	Affects	Cell_type	0.67
Protein_subunit	Associates	Protein_subunit	0.50
Cell_type	Activated	Protein_molecule	0.40
Cell_type	Contained in	Cell_type	0.33
Nucleotide	Mediate	Lipid	0.00

Tabla 3.2: Instancias de Relaciones Biológicas borrosas extraídas de la recopilación GENIA (en inglés).

También está basado en ontologías el trabajo de Quan et al. [QUAN06]. En este caso, se trata de desarrollar un sistema de ayuda (*help-desk*) semántico para apoyar al servicio de atención al cliente en un entorno de webs semánticas. En particular, un análisis de conceptos formales borroso (FCA, *Formal Concept Analysis*) se desarrolla para la generación automática de ontologías borrosas que pueden tratar con información incierta. La técnica de generación borrosa automática de ontologías propuesta consta de los pasos siguientes: análisis de concepto formal borroso, *clustering* conceptual borroso, y generación de ontologías. En el ámbito de las webs semánticas, se adopta la ontología como un estándar para la representación de conocimiento. Los programas pueden usar el conocimiento de las webs semánticas para procesar la información semánticamente. Así, se permite que máquinas diferentes o modelos producidos por fabricantes diferentes puedan ser compartidos e integrados. Además, la ontología generada también puede usarse para proporcionar una interpretación sobre los defectos comunes.

Los datos pueden ser almacenados en un formato inestructurado, semiestructurado, o totalmente estructurado (p.e., documentos textuales o base de datos). Sin embargo, es una tarea muy difícil e incómoda realizar una ontología "a mano". Las investigaciones más recientes intentan abordar este problema mediante el denominado estudio de ontología de texto libre, datos semiestructurados (p.e., HTML O XML), o datos estructurados de una base de datos. Se han utilizado diferentes técnicas para generar ontologías, entre las cuales cabe destacar las basadas en el *clustering*. Sin embargo, el formalismo conceptual apoyado por una ontología estándar puede no ser suficiente para representar la información incierta que habitualmente se encuentra en la información almacenada en la mayor parte de bases de datos. Por ejemplo, en una base de datos de un servicio de atención al cliente, cada problema registrado se describe mediante una cadena de texto. Se pueden extraer en este caso una serie de palabras, utilizándose estas para la recuperación de información. Sin embargo, es inadecuado tratar todas las palabras clave de la misma forma dado que algunas palabras clave pueden ser más significativas que otras. Una posible forma de manejar este aspecto es la incorporación de la lógica borrosa a la ontología. Un concepto puede ser descrito por una serie de palabras clave, mientras que los conceptos borrosos formales son organizados entonces como un entramado de conceptos borrosos, que proporcionan relaciones jerárquicas entre los conceptos borrosos formales. Basándose en tales relaciones jerárquicas, los conceptos borrosos formales pueden ser representados como subconceptos y superconceptos unos de otros. Un concepto formal es un

modelo conceptual que potencialmente representa un verdadero concepto en el dominio del concepto borroso. Las relaciones jerárquicas entre los conceptos formales implican la relación entre los conceptos generalización y especificación. Un superconcepto es más general que su subconcepto y viceversa.

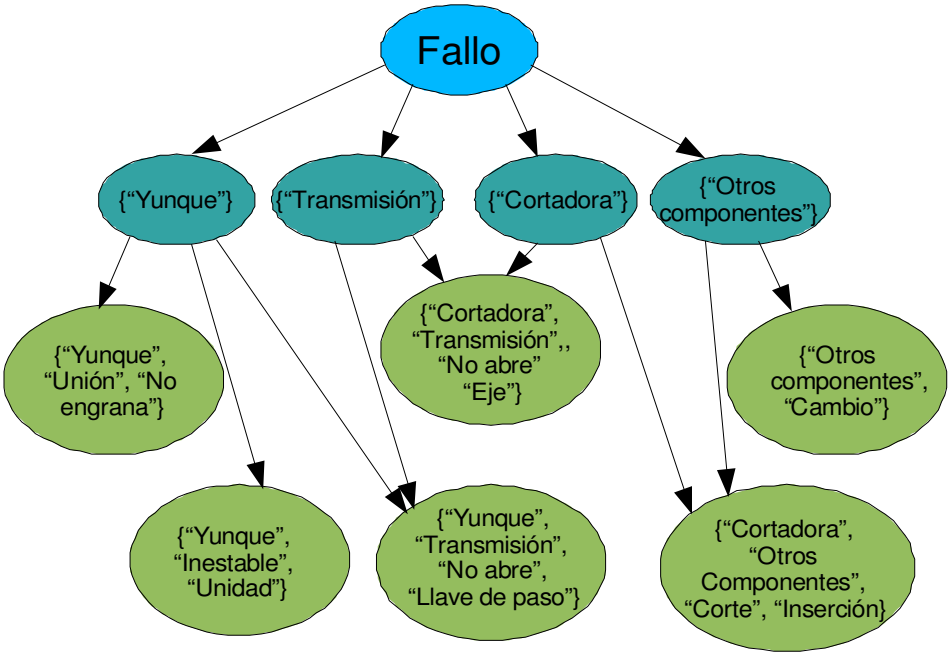


Fig. 3.13. Ejemplo. Jerarquía del concepto “Fallo” en la máquina AV_2011

Por ejemplo, la Figura 3.13 muestra la jerarquía de concepto “Fallo” en la máquina AV_2011, que muestra que los fallos de esta máquina a menudo pueden agruparse según el componente fallido, a saber: el yunque, la transmisión, la cortadora y otros componentes. Cada concepto está representado por sus palabras clave más frecuentes, mientras que cada componente está representado como un concepto, cuyos subconceptos indican los fallos comunes del componente correspondiente. Por ejemplo, el primer grupo indica los fallos que ocurren cuando la unión del yunque no puede engranar. En el segundo grupo, la unidad del yunque es inestable. El tercer grupo muestra los fallos que ocurren en el yunque y la transmisión cuando la llave de paso no se abre. En el algoritmo conceptual de *clustering* borroso, un concepto formal puede pertenecer a más de un *cluster* conceptual. Por ejemplo, el concepto C es similar al concepto A y al concepto B, pero los conceptos A y B no pasan el umbral de confianza de semejanza. En este caso, C queda agrupado con A y con B. Esto refleja el fenómeno real de que un objeto puede pertenecer a más de un concepto en la vida real.

Otro ejemplo de ontología es el de Barbancho et al. [BARBANCHO09], ya descrito en el capítulo 2 de esta tesis.

Un último ejemplo de aplicación de la FL a la búsqueda de información, en este caso para una cadena de producción es el de Zhai et al. [ZHAI08]. La IR es un punto importante en la Administración de la Cadena de Suministro (*Supply Chain Management*, SCM). Para conseguir una recuperación semántica borrosa, se aplica un entorno de trabajo de ontologías borroso al sistema de recuperación de datos en SCM. De esta manera, se busca recuperar información acerca de, por ejemplo, un producto, a partir de características como el precio del producto, los ingresos del cliente, sus condiciones o la zona de influencia de la empresa.

En todo caso, todas estas aplicaciones tienen en común con el trabajo que se propone dos aspectos primordiales:

- La gran cantidad de información a manejar.

- La estructura jerárquica o posibilidad de agrupamiento de esta información.

Por tanto, la idoneidad de la lógica borrosa y la posibilidad de profundizar en este campo, nos hace decidirnos por ella para el diseño del Agente Inteligente para la búsqueda de información en entornos ruidosos e imprecisos como, por ejemplo, puede ser un portal web. La forma en la que se utiliza la lógica borrosa para este diseño será el punto principal del capítulo 4 de esta tesis.

Capítulo 4. Método general para la búsqueda de conocimiento basado en FL.

El empleo eficaz de Internet por parte de las personas o de sistemas automáticos de toma de decisiones ha sido obstaculizado por algunas características dominantes de la WWW. En primer lugar, la información disponible de la red está desorganizada, es multimodal, y está distribuida en los servidores de todo el mundo. En segundo lugar, el número y la variedad de fuentes de datos y de servicios aumenta radicalmente día a día. Además, la disponibilidad, el tipo y la fiabilidad de los servicios de información cambian constantemente. Además, la información es ambigua e incluso a veces errónea debido a la naturaleza dinámica de las fuentes de la información, a la puesta al día de esta misma información y a problemas de mantenimiento. Por lo tanto, la información se hace cada vez más difícil de recoger, filtrar, evaluar, y usar [SYCARA96].

Con el fin de solucionar estos problemas, se ha propuesto el concepto de Agente Inteligente. Los agentes son entidades software programadas que llevan a cabo una serie de operaciones en nombre de un usuario o de otro programa, con algún grado de independencia o autonomía, empleando algún conocimiento o representación de los objetivos o deseos del usuario [MENGUAL01].

En este capítulo de la tesis, se propone el uso de un Agente Inteligente para la búsqueda general de conocimiento. En el capítulo siguiente, este método se aplica específicamente a entornos web, por la idoneidad del uso de la FL para la búsqueda de información en dichos entornos. En la sección 4.1 se aborda una introducción al uso de Agentes Inteligentes y se introducen algunos Agentes Inteligentes usados para la búsqueda de información, tanto en el ámbito de la investigación, como en el ámbito comercial. En la sección 4.2, se propone un Agente Inteligente basado en lógica borrosa, con la consecuente mejoría cosechada en términos de flexibilidad. Se describe el modo de funcionamiento del Agente, aprovechando la estructura jerárquica de la información contenida en los portales web. Por último, en la sección 4.3, se describe el sistema de lógica borrosa empleado, haciendo hincapié en la elección de las reglas y

los parámetros adecuados del sistema y analizando algunas herramientas disponibles para el diseño de estos sistemas.

4.1. Agentes inteligentes para la búsqueda de conocimiento.

Como se apuntó en la introducción a este capítulo, los agentes son entidades software programadas que llevan a cabo una serie de operaciones en nombre de un usuario o de otro programa, con algún grado de independencia o autonomía, empleando algún conocimiento o representación de los objetivos o deseos del usuario. Cuando hablamos de agentes, podemos distinguir tres dimensiones o coordenadas, las cuales usamos como medida de sus capacidades: representación o mediación, inteligencia y movilidad [MENGUAL01].

- La representación tiene que ver con el grado de autonomía que el agente software tiene al representar a otros agentes, aplicaciones o sistemas de ordenadores.

- La inteligencia se refiere a la capacidad del agente para capturar y aplicar el conocimiento y el procesamiento para solucionar problemas. Por lo tanto, los agentes pueden ser relativamente simples usando únicamente la lógica o pueden ser relativamente sofisticados usando complejos métodos basados en AI como inferencias y aprendizaje.

- Así mismo, los agentes pueden ser estáticos o móviles. Un agente estático es aquel que sólo puede ejecutarse en la máquina donde fue iniciado, mientras que un agente móvil es aquel que no está limitado al sistema donde se inició su ejecución, siendo capaz de transportarse de una máquina a otra de la red. Esta posibilidad le permite interactuar con el objeto deseado de forma directa sobre el sistema de agentes donde se halla dicho objeto.

En definitiva, un Agente Inteligente es un programa que ayuda al usuario en tareas rutinarias con máquinas. Como se trata de una tecnología relativamente nueva, se utilizan abundantes nombres para describirlos, tales como Agentes Software, *Wizards* (textualmente, “Magos”) y Sistemas Multi-Agente [TURBAN01]. En el caso de que mantengan algún tipo de conversación con el usuario, también se denominan Agentes Conversacionales, robots software (*bots*) o *chatbots* Algunas de las aplicaciones implementadas por Agentes Inteligentes incluyen sistemas tutoriales, diseño y análisis de sistemas, mantenimiento electrónico, representación del conocimiento, sistemas adaptativos, sistemas anti-contaminación, sistemas de apoyo a decisiones y un largo etcétera [LIAO05].

4.1.1. Agentes Inteligentes en investigación.

Aunque, como se ha dicho anteriormente, el concepto de Agente Inteligente aún no ha sido definido exactamente, la idea principal es que los Agentes Inteligentes son programas que actúan de parte de usuarios humanos u otros sistemas automáticos. Para el dominio de la búsqueda y extracción de conocimiento, se puede utilizar un Agente Inteligente para realizar funciones laboriosas de recolección de datos en tareas como las siguientes:

- Localización y acceso a la información de varias fuentes de información en línea.
- Resolución de inconsistencias en la información recuperada.
- Filtrado de información irrelevante o no deseada.
- Integración de la información en fuentes heterogéneas.
- Adaptación temporal a las necesidades de información de los usuarios humanos y a la forma de la WWW.

Recientemente ha aumentado el interés en los denominados agentes conversacionales, asistentes virtuales, robots software, *chatbots* o simplemente *bots*, los cuales actúan como proveedores eficaces y familiares de la información. Los agentes conversacionales son agentes inteligentes representativos que son capaces de responder de un modo inteligente (con un diálogo en lenguaje natural) a las preguntas de los usuarios, entendiendo además la intención de usuarios en la conversación y permitiendo la búsqueda y extracción de la información relevante. Los Agentes Inteligentes conversacionales constituyen una interfaz ideal para la búsqueda de conocimiento para cualquier usuario inexperto.

Una de las aplicaciones principales de los robots software (*bots*) es permitir a los usuarios mantener una conversación acerca de la información contenida en un sitio web dado. Cuando un usuario hace una pregunta al *bot*, el *bot* siempre responde con un trozo de información mediante el uso del emparejamiento de patrones y de los sinónimos. Los temas se enlazan con ciertos patrones de entrada del usuario que describen las diversas formas en las cuales el sistema espera que los usuarios realicen ciertas afirmaciones o hagan preguntas sobre rasgos particulares. La posterior evaluación de estos robots permitió llegar a la conclusión tan ampliamente aceptada hoy de que el tratamiento del Lenguaje Natural o la mera integración de medios de comunicación no son suficientes para construir sistemas que puedan llevar a cabo un diálogo realmente cooperativo con los usuarios [GARCÍA-SERRANO04]. Por tanto, existe un requisito fundamental: la inteligencia del sistema.

En la actualidad existen Agentes Inteligentes conversacionales para las más diversas aplicaciones, desde el comercio electrónico o *e-commerce* [AJAYI09, GARCÍA-SERRANO04] a la enseñanza virtual [KERLY07, WIK09], pasando por usos médicos [EISMAN09, BICKMORE09]. A continuación se describe la forma en la que algunos de estos autores han tratado diversos aspectos a tener en cuenta de estos Agentes Inteligentes.

Un aspecto fundamental en la creación de un Agente Inteligente es el diseño de su arquitectura. Por ejemplo, en el caso del proyecto ADVICE [GARCÍA-SERRANO04], el

objetivo final es diseñar e implementar un sistema que proporcione consejos para el comercio electrónico (*e-commerce*), haciendo que se pase del actual servicio al cliente basado en un catálogo a un servicio de atención al cliente con ayuda inteligente, emulando de algún modo el funcionamiento de un vendedor humano. Con este objetivo, los elementos principales de ADVICE incluyen una arquitectura basada en agentes con un Agente de Interfaz para manejar la presentación de multimedia, un Agente de Interacción que da soporte al intercambio de información entre el usuario y el sistema y un Agente Inteligente que incorpora un modelo basado en un conjunto de conocimiento del comercio electrónico para el dominio correspondiente y que soporta el razonamiento necesario para aconsejar al usuario según con sus necesidades y la evolución del diálogo.

Por tanto, el sistema consta de:

- Un cliente ADVICE, con comunicación directa a través de un navegador web o un teléfono wap.

- Un sistema de venta, que proporciona información sobre productos y que dirige el proceso de venta.

- Un servidor ADVICE, con tres componentes principales:

    - Agente de interfaz. Responsable de la comunicación multimedia con los usuarios, mediante una interfaz gráfica de usuario, incluyendo un personaje 3D, y un procesador de texto en Lenguaje Natural (NL). Como entradas, se recogen las expresiones del usuario (frases en inglés, clics en artículos - como iconos, menús, etc. -) y se transforman en estructuras semánticas. Como salidas, se recibe la información que debe mostrarse en forma de estructuras semánticas y se transforma en información presentable al usuario. Uno de los aspectos principales en las operaciones de compra-venta en la web es la capacidad del sitio web de generar algún tipo de sentimiento de confianza en el comprador, como haría un dependiente humano en una compra-venta cara a cara. Características tales como el entendimiento de las necesidades del comprador, la capacidad de dar un consejo técnico o la ayuda a la decisión final no son aspectos fáciles de alcanzar con un sitio web para el comercio electrónico. Las técnicas NLP pueden jugar un papel crucial para realizar estas mejoras. Otra buena motivación para integrar la tecnología NL en esta clase de sitios web es hacer que la interacción sea más fácil para aquellas personas con menos confianza en las nuevas tecnologías. Por otro lado, el intérprete de NL usa una aproximación basada en el reconocimiento de patrones mediante técnicas de extracción de mensaje útiles en dominios específicos, siendo estos mensajes formulados mediante estructuras semánticas que reflejan oraciones genéricas para distintos idiomas en el comercio electrónico, patrones para la jerga específica del dominio y palabras clave.

    - Agente de interacción. Responsable de la evolución coherente del diálogo entre sistema y usuario, actuando como intermediario entre las necesidades de usuario, interpretadas por el Agente de Interfaz, y la capacidad del sistema de satisfacerlos, representado en el Agente Inteligente. El Agente de Interacción tiene que: manejar la evolución de la conversación de un modo coherente; entregar al Agente Inteligente la consulta de usuario junto con la información relevante

acerca de este usuario que pueda influir en la selección de la respuesta apropiada; por último, debe enviar al Agente de Interfaz la información para que sea presentada al usuario en cada momento.

- Agente Inteligente. Tiene una responsabilidad doble: la generación del contenido de la respuesta según las expectativas de usuario, por ejemplo la identificación del producto o los productos que mejor encajan en las necesidades de usuario o la descripción de los pasos a realizar para solucionar un problema; y proporcionar ciertos criterios para dirigir la evolución del diálogo con el usuario de un modo inteligente. En el estado actual de la ingeniería del conocimiento, un modelo de conocimiento puede ser concebido como un modelo de resolución de problemas estructurado jerárquicamente. En este caso, se aplica el denominado principio de organización. Este principio establece que un modelo de conocimiento puede ser organizado como una jerarquía de áreas de conocimiento. Se establecen una serie de relaciones jerárquicas relacionadas mediante ontologías.

En [KERLY07], se presenta un sistema que permite a los estudiantes realizar una autoevaluación de sus capacidades en diferentes aspectos, y comparar estos con las opiniones del sistema, basándose en las preguntas que han contestado. También les permite contestar preguntas sobre cualquier evaluación, y actualizar su propia evaluación. Además, se ha desarrollado un agente conversacional, o *chatbot* para permitir al estudiante el uso del lenguaje natural. El sistema tiene el objetivo de apoyar las metas metacognitivas de autoevaluación y reflexión, que cada vez son más aceptadas como claves en el estudio y están siendo incorporadas en la política educativa británica. El *chatbot* permite a los estudiantes usar el lenguaje natural para preguntar por sus capacidades, pedir al sistema sus opiniones, justificar sus propios errores si el sistema discrepa, cambiar sus opiniones una vez reexaminadas sus capacidades, aceptar la creencia del sistema, tratar de encontrar un compromiso con el sistema, o hacer preguntas tipo test de forma remota. El objetivo es reducir el número de aspectos sobre los cuales el sistema y el estudiante discrepan, haciendo la autoevaluación del estudiante más precisa.

Con el fin de proporcionar una negociación basada en las diferentes opiniones sostenidas por el usuario y el sistema, es necesario almacenar unos "conjuntos de opiniones" de forma independiente. Cuando un usuario se conecta al sistema por primera vez, se requiere que este proporcione una posición de autoevaluación (uno de los cuatro niveles de confianza) para cada uno de los aspectos que sugiere el sistema. Siempre que los usuarios contestan una pregunta test, estos deben también replantearse su grado de confianza en su capacidad sobre el aspecto dado. La opinión del sistema para cada aspecto se calcula de modo similar siempre que el usuario contesta una pregunta, basándose en si el usuario contestó correctamente. Una vez que un usuario ha contestado un número dado de preguntas sobre un asunto, el sistema supone que tiene bastantes datos como para comenzar a mostrar su opinión. El sistema mantiene su modelo del conocimiento actual del usuario aplicando un algoritmo de recursivo ponderado con cada nueva respuesta u opinión proporcionada por un estudiante. Esto resulta en un modelo donde los datos más recientes tienen mayor peso, y los resultados más antiguos tienen un efecto cada vez más menor sobre la confianza del usuario o los valores de la opinión del sistema.

Por otra parte, el Agente Inteligente conversacional tiene dos funciones principales:

- Responder al usuario en una conversación iniciada por él sobre su conocimiento o la razón de las opiniones del sistema.

- Iniciar una discusión sobre las discrepancias entre los puntos de vista del sistema y del usuario.

Una vez iniciada la conversación, el *chatbot* entiende una amplia gama de entradas. La mayor parte de la base de conocimiento, como es de esperar, se relaciona con la discusión sobre el aprendizaje y el estudiante. Los usuarios pueden proporcionar entradas como "¿por qué pensamos de manera diferente?", "¿en qué voy bien?", "¿cuál es su opinión?", "Tengo dificultades con la Electricidad" o "¿qué es lo próximo que debería hacer?". Los usuarios simplemente pueden solicitar que el *chatbot* les diga sus opiniones, un apoyo del sistema para cualquiera de las asignaturas, o solamente que les haga preguntas test sobre una materia específica.

El entorno de modelado para el estudiante, incluyendo la parte del servidor encargada del procesado y la interfaz basada en un navegador, fue implementado mediante el uso de ASP.NET, *framework* para aplicaciones web desarrollado y comercializado por Microsoft. El sistema usa una base de datos SQL para almacenar todos los datos del modelo, incluyendo aspectos del dominio de las asignaturas, preguntas y respuestas, y conexiones al sistema. Para desarrollar el *chatbot*, se seleccionó la tecnología Lingubot comercial [CREATIVE09]. La tecnología Lingubot ha establecido una recopilación significativa de *scripts* externos y aplicaciones, proporcionando una funcionalidad relevante a las conversaciones. Tiene también la capacidad de generar y manipular variables e información referente a la conversación, y de recuperar y enviar la información a otras aplicaciones web como motores de búsqueda y bases de datos. La naturaleza comercial de la tecnología también asegura que la fiabilidad de Lingubot ha sido ampliamente testada (empresas británicas como O2, Lloyds TSB o NationalRail – ver Figura 4.1 – han usado esta tecnología para sus Asistentes Virtuales).

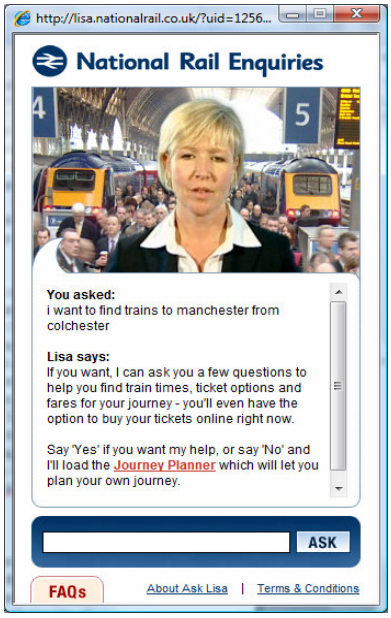


Figura 4.1. Asistente virtual de NationalRail (UK), realizado con la tecnologia Lingubot.

El motor de ejecución del *chatbot* fue instalado sobre el mismo servidor que el sistema .NET y la base de datos SQL. El *chatbot* se muestra en una ventana dedicada en la máquina del estudiante. Una conexión ODBC permite al *chatbot* extraer el material de la base de datos para realizarle preguntas, dar respuestas resueltas al usuario y pasar información al usuario.

También Wik et al. [WIK09] toca el mundo de la enseñanza virtual mediante un Agente Inteligente Conversacional. En este caso los autores lo denominan Agente conversacional integrado (ECA, *Embodied Conversational Agent*). El sistema, llamado Ville, es un profesor virtual de idiomas para el entrenamiento de la pronunciación y el vocabulario. Ville se pone en el papel del profesor, seleccionando las palabras los estudiantes deben decir. Esto es una gran ventaja en la etapa de análisis en Ville, puesto que facilita la tarea de corregir errores de pronunciación, por medio de la realización de hipótesis acerca de qué tipo de errores de pronunciación puede un estudiante cometer con mayor probabilidad. Ville también ayuda con el entrenamiento del vocabulario, el suministro de un modelo de pronunciación nuevas palabras, y la insistencia en ejercicios de memorización. La primera implementación de este profesor de idiomas virtual da clases de sueco a estudiantes de la universidad para alumnos extranjeros en el Instituto Tecnológico Real KTH de Estocolmo, aunque el objetivo final de los autores es crear un tutor de idiomas más general.

El Profesor de Idiomas Virtual (VLT, *Virtual Language Teacher*) puede dividirse en unidades funcionalmente separadas, pero que actúan en conjunto, como se observa en la Figura 4.2. A continuación se expone una descripción de como se prevé que sea el VLT.

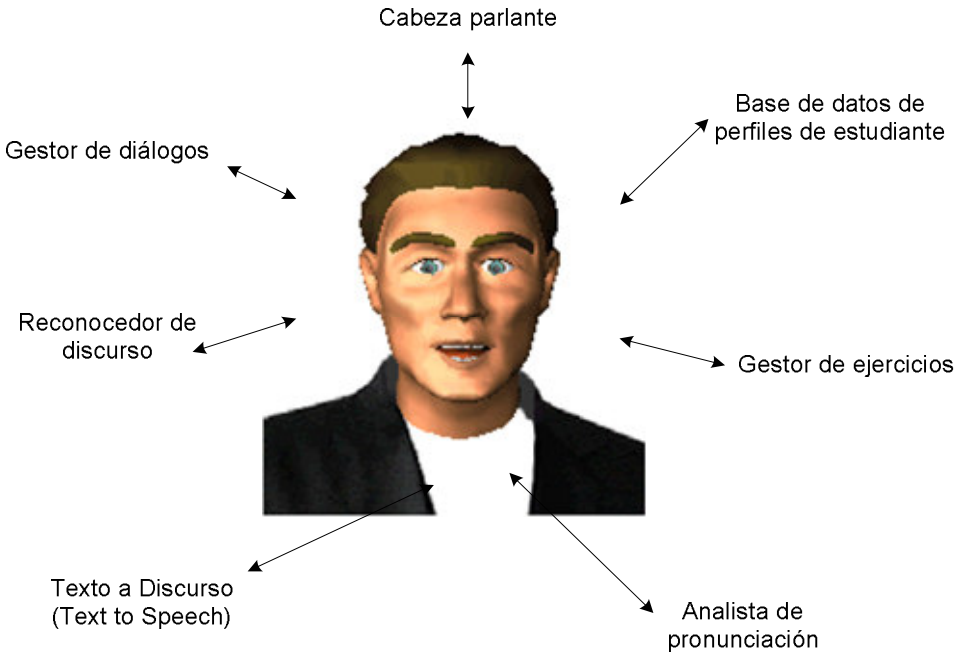


Figura 4.2. Arquitectura del Agente Inteligente Ville [VILLE09].

A continuación se describen algunos de los módulos del VLT:

- Cabeza Parlante: existen varias cabezas para escoger, con un API para hacerlos conversar y para mover la cabeza y las distintas partes de la cara, como la boca. Actualmente, su diseño es motivo en el CTT de Estocolmo.

- Text to Speech (TTS): Es el conversor de texto a diálogo. Aunque se han probado varios tipos de TTS en el VLT, la versión actual usa un conjunto de *prompts* pregrabado.

- Análisis de Pronunciación: el análisis de pronunciación automático constituye un área de investigación abierta. Actualmente, los autores se centran en problemas específicos de la lengua sueca. Estos incluyen entre otros la duración/cantidad, detección de acentos o la calidad vocal.

- Reconocimiento de discurso: la versión actual de Ville no usa reconocimiento de discurso automático.

- Gestión de Diálogos: la versión corriente de Ville no usa ningún sistema de gestión del diálogo. Un proyecto paralelo a Ville, llamado DEAL, incorpora un sistema de diálogo que se puede usar para la práctica de conversaciones.

- Base de datos de perfiles de estudiante: la versión actual de Ville no la usa.

- Gestión de ejercicios: Para adaptar las lecciones a las necesidades individuales de estudiantes, un gestor de ejercicios es necesario. La versión actual de Ville tampoco usa gestor de ejercicio.

Aunque muchos de estos módulos no hayan sido definidos, suponen un buen punto de partida para ilustrar como debe ser la arquitectura un Agente Inteligente conversacional.

Otros autores también pretenden buscar una mayor flexibilidad en los Agentes Inteligentes. En [KIM06] se propone un agente conversacional basado en Redes Bayesianas Semánticas (SeBN, *Semantic Bayesian Networks*). El objetivo son los sitios web móviles, a los que se puede acceder mediante teléfonos celulares, cámaras digitales, y reproductores MP3. La base de datos fue construida extrayendo la información de cinco sitios web: Naver.com (www.nshopping.naver.com), Samsung-mall (www.samsung-mall.co.kr), LG-eshop (www.gseshop.co.kr), Enuri.com (www.enuri.com) y DCinside.com (www.dcincide.com).

El agente propuesto en este artículo no sólo reduce la complejidad de la construcción, sino que también deduce las intenciones del usuario con más detalle. Dado que las conversaciones contienen a menudo expresiones ambiguas, la capacidad de manejar el contexto o la incertidumbre es muy importante en la construcción de agentes conversacionales flexibles. El método propuesto usa una interacción de iniciativa mixta para obtener la información que falta y clarificar para conceptos espurios para entender la intención de los usuarios correctamente. Este método no sólo reduce la complejidad de las redes, sino que también infiere la intención de los usuarios más activamente. De hecho, esta idea es una de las ideas claves de esta tesis, con la diferencia de que, en nuestro caso, se ha utiliza la lógica borrosa para dotar al sistema de esta flexibilidad.

El agente conversacional propuesto está compuesto de dos partes:

- Interfaz de diálogo multimodal. Proporciona una interfaz de usuario agradable, así como el tratamiento de preguntas generales basadas en el reconocimiento de patrones. Así, los desarrolladores del sistema pueden construir fácilmente *scripts* de respuesta independientes del dominio de aplicación.

- Módulo de inferencia. Está compuesto del motor de inferencia y el módulo de gestión del conocimiento, en el que el motor de inferencia analiza lo que el usuario quiere a partir de preguntas ambiguas. El módulo de gestión del conocimiento almacena la información sobre el dominio correspondiente mediante la extracción de contenido específico de páginas web e introduciéndolo en una base de conocimiento. Si no hay bastante información para inferir la intención del usuario, se recoge información adicional acerca del usuario de forma activa.

Para poder gestionar varias preguntas, es necesario dividir los diálogos en bloques y establecer una prioridad jerárquica según el tipo de diálogo. En la etapa de pre-procesado, las palabras clave se extraen de la consulta de entrada para encontrar dichas palabras clave en los scripts de respuesta. Estas respuestas pueden ser la salida cuando los scripts coinciden. El problema es que la correspondencia de patrones tradicional obtiene un gran resultado cuando hay muchas coincidencias, dado que solo se contempla el número de estas. Para obtener una inferencia eficiente, se diseñan redes Bayesianas semánticas compuestas de la inferencia

probabilística y una inferencia semántica. Este modelado ayuda a entender las intenciones de los usuarios basándose en la conversación. La red Bayesiana semántica propuesta tiene tres niveles según su función: palabras clave, conceptos, y objetivos, tal y como se observa en la Figura 4.3. La capa correspondiente a las palabras clave consiste en palabras relacionadas con la pregunta del usuario, mientras que la capa de concepto está compuesta de las entidades u objetos del dominio y sus relaciones semánticas. La capa objetivo representa la información objetivo (productos). La capa de concepto está dividida a su vez en tres componentes: objetos, atributos, y valores. Cada objeto es un conjunto de pares de valor-atributo, donde el nodo a_i es un atributo y el nodo el nodo v_k es un valor en el dominio. Las líneas continuas representan la relación probabilística entre nodos, mientras que las líneas de puntos representan la relación semántica entre ellos.

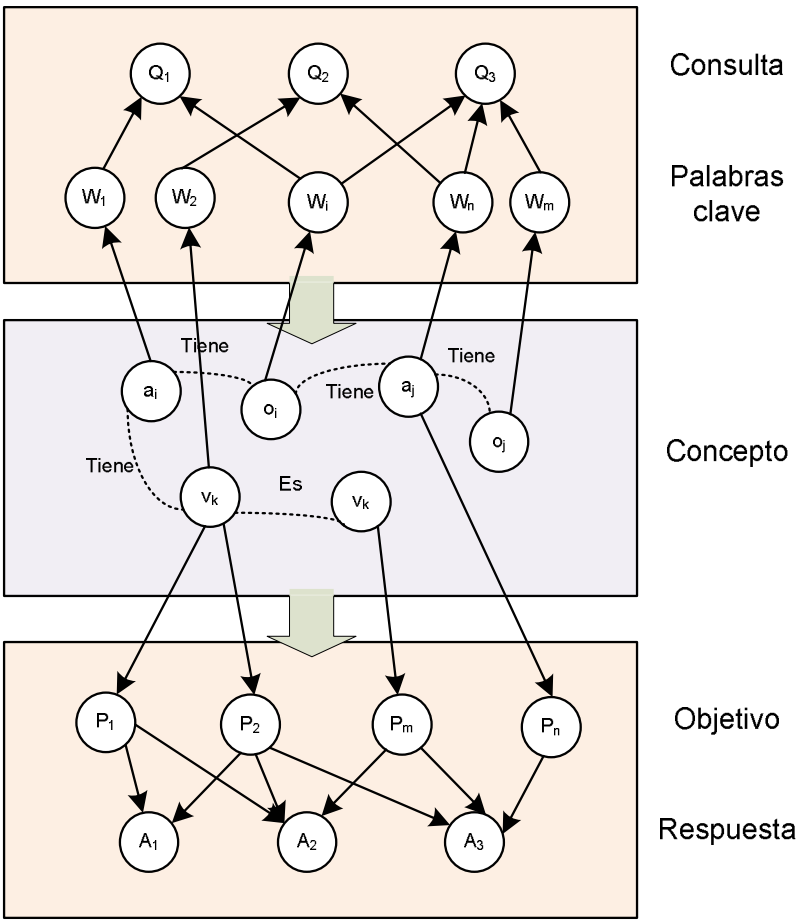


Figura 4.3: Arquitectura de una red bayesiana.

La relación probabilística en redes Bayesianas semánticas es similar a la del modelo de IR tradicional. Primero, se infiere probabilisticamente la relación entre la capa de palabras clave y la capa de conceptos. La consulta de usuario $U = \{k_1, k_2..., k_t\}$, donde la palabra clave k_i se interpreta como una palabra elemental en la capa de palabras clave. Se pone un nodo de palabra

clave a 1 cuando la palabra dada en la capa de palabras clave se encuentra en la búsqueda Q. En caso contrario, se pone a 0. Entonces el sistema infiere la probabilidad para cada nodo de la capa de conceptos cuando todas las variables de prueba asociadas a las palabras clave son puestas a 1, mediante reglas bayesianas, teniendo en cuenta el número de nodos, el conjunto de palabras clave, el de objetos, el de atributos y el de valores. Por último, se selecciona un nodo en la capa de objetivos cuya probabilidad es más alta que un cierto umbral después de la inferencia. Esto proporciona la información sobre el producto objetivo del usuario cuando se selecciona un número apropiado de nodos. La asignación de probabilidades está basada en el criterio del diseñador según la posible frecuencia de ocurrencia de una de palabra clave W y en los conceptos objetivo Ci. Si se identifica una palabra clave W como obligatoria para un concepto de objetivo Ci, se asignará una alta probabilidad (por ejemplo, del orden de 0.95 a 0.99). Por ejemplo, podría haber una correlación cercana entre la palabra clave "matiz" y el concepto "color". Así mismo, los valores asignados disminuyen a 0.7-0.8 para la palabra clave "rojo" y el concepto "color". Sin embargo, la palabra clave "azul" por lo general no está asociada con la palabra clave "rojo", por lo que los valores disminuyen a 0.2-0.3. Cuando no se selecciona ningún producto, se ejecuta la inferencia semántica de redes Bayesianas en la capa de concepto. Hay dos relaciones principales entre nodos (“Tiene”, y “Es”). Por ejemplo, una posible relación semántica para un teléfono móvil es: atributo “grande” – valor del precio “bajo”.

Un último aspecto de los Agentes Inteligentes conversacionales en el que se han centrado otros investigadores son la interfaz gráfica, la posibilidad de que el Agente Inteligente utilice una voz humana, el estudio de los gestos que este debe de realizar o incluso de las emociones de un Agente Inteligente utilizando la Lógica Borrosa.

Una posibilidad interesante para la creación de una interfaz gráfica es la creación de un avatar. Un ejemplo de programa para la creación de avatares de este tipo es Qavatar MA (Modeler and Animaker) [QAVATAR09], el cual es un potente instrumento de creación 3D de Qtelsoft que permite crear avatares propios sin conocimiento especializado alguno de gráficas en 3D. Se puede ver el avatar, la animación, y los datos de voz para el Qavatar MA sobre cualquier página web que use el Control ActiveX Q3DViewer, un visor total de Web3D. Igualmente se puede usar VRML97 Wizard. Qavatar MA permite registrar la voz y hacer una animación de sincronización de labios, permitiendo además la exportación del avatar, la animación, y los datos de voz a un archivo AVI. En la Figura 4.4 se observa un avatar creado por el autor de la tesis con la versión demo de este programa.

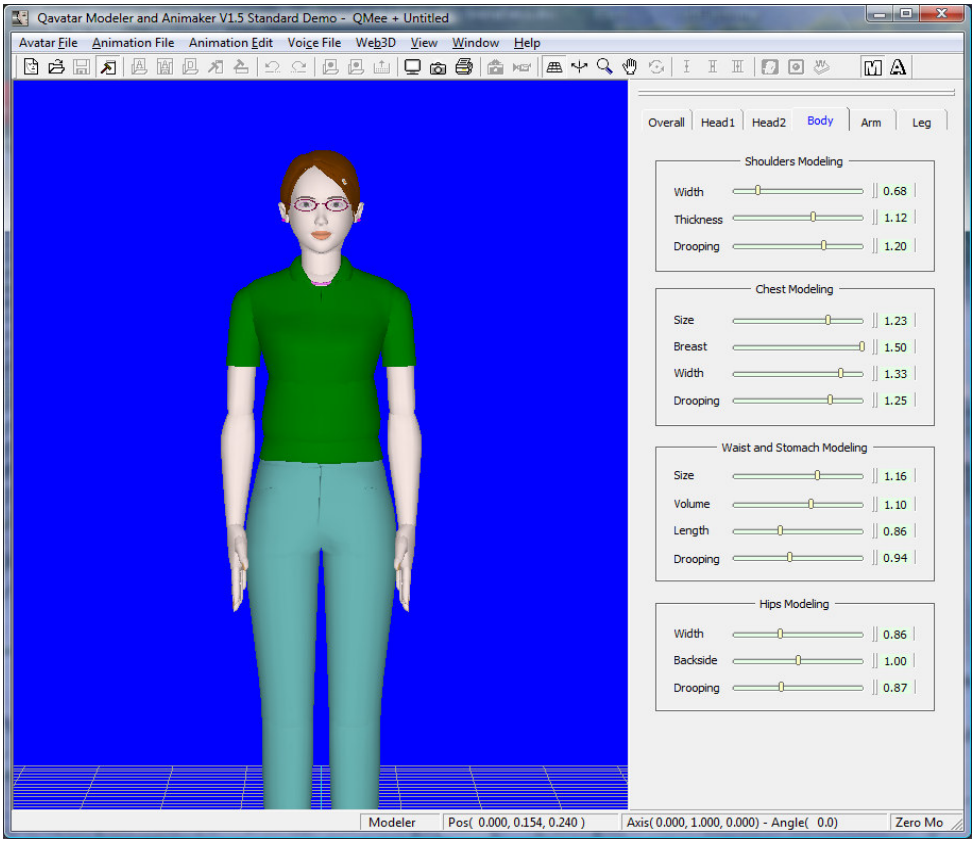


Figura 4.4. Avatar creado por el autor de la tesis con la demo del programa Q-avatar.

En cuanto a la emulación de la voz humana, también se pueden utilizar programas comerciales. El programa VoiceText [VOICETEXT09], por ejemplo, puede convertir datos de texto en voz natural humana por medio del análisis de la estructura gramatical del texto y realizando la entonación del diálogo más apropiada. VoiceText presenta una tecnología de diálogo natural y un pronunciación clara en distintas lenguas (inglés, español, coreano, chino y japonés, con la demo del programa en la web http://www.voiceware.co.kr/english/demo/demo_text.html). VoiceText está disponible para una amplia gama de dispositivos integrados, aplicaciones de red/servidores, telemática, lectura de páginas web, robots, domótica, PDAs, teléfonos móviles, educación y entretenimiento.

Por último, otros investigadores se han centrado en los gestos y emociones del Agente Inteligente. Pelachaud describe dicho comportamiento expresivo mediante la utilización de un conjunto de seis parámetros que actúan como modelo para la animación de dicho comportamiento [PELACHAUD08]. Estas seis dimensiones que caracterizan la expresividad son:

- Extensión espacial: espacio ocupado por las distintas partes del cuerpo.
- Extensión temporal: velocidad de ejecución de un movimiento.
- Fluidez: nivel de continuidad de los movimientos.

- Energía: dinamismo del movimiento.
- Activación general: cantidad de movimiento total.
- Repetición: repetición del ritmo de un movimiento.

Mención especial, por el uso de la lógica borrosa, merece el trabajo de Eisman et al [EISMAN09]. Hay varios modelos para representar el estado emocional de un agente conversacional. En esta sección se analizan dos aproximaciones, el modelo de emociones básico de Yanaru et al. [YANARU97] y el modelo circunflejo de afecto de Russell y Barrett. [RUSSELL99]

El modelo de emociones básico propone una representación n-dimensional con los siguientes postulados:

- Hay un pequeño número de emociones puras o primarias.
- El resto de emociones son emociones compuestas, es decir, pueden ser obtenidas combinando emociones primarias en niveles de intensidad diferentes.

El número de atributos emocionales varía mucho entre distintas publicaciones, aunque los autores por lo general eligen ocho atributos emocionales, de modo que los vectores correspondientes a los estados emocionales sean de ocho dimensiones. La elección de estos atributos también depende mucho del dominio específico. Yanaru et al. definen los siguientes atributos emocionales: alegría, tristeza, cólera, miedo, expectativas, sorpresa, odio y aceptación. Para cada vector, componentes de atributo emocionales pertenecen al intervalo [-1, 1], aunque el uso del intervalo [0,1] es también habitual. Russell y Barrett presentan una alternativa a este modelo. Según estos autores, el modelo de emociones básico está basado en una secuencia de categorías básicas discretas y mutuamente excluyentes, por lo que si un agente siente dos emociones simultáneamente, cada emoción pertenece a una y sólo a una categoría básica. Además, las categorías son borrosas, ya que no pueden dividirse abruptamente.

El sistema de control de estado emocional de un agente conversacional integrado está basado en dos conceptos esenciales como el estado emocional y la personalidad. Cada atributo, ocho en este sistema, corresponde a una dimensión del espacio en el cual el estado emocional está representado por vectores. Estas ocho dimensiones corresponden con los siguientes atributos emocionales: alegría, desdén, ira, miedo, preocupación, sorpresa, tristeza y vergüenza. El valor para cada atributo es un número real entre 0 (ausencia total de la emoción correspondiente) y 1 (presencia total).

Al asignar la personalidad de un agente en particular, existe una pequeña variación en el valor de los atributos emocionales asociados con dicha personalidad de manera aleatoria. De este modo, dos agentes que comparten la misma personalidad no tienen que ser exactamente los mismos. En la Tabla 4.1, se muestran dos ejemplos de estados emocionales diferentes. Por otro lado, en la Tabla 4.2 se observan distintos tipos de personalidad para cada agente.

Estado emocional	**Atributos emocionales**							
	Ale	Des	Ira	Mie	Pre	Sor	Tri	Ver
Estado 1	0.73	0.26	0.12	0.08	0.11	0.39	0.04	0.10
Estado 2	0.05	0.03	0.07	0.48	0.39	0.10	0.86	0.01
Leyenda: Ale(gría), Des(dén), Ira, Mie(do), Pre(ocupación), Sor(presa), Tri(steza), Ver(güenza)								

Tabla 4.1: Posibles estados emocionales de un Agente Inteligente.

Personalidad	**Atributos emocionales**							
	Ale	Des	Ira	Mie	Pre	Sor	Tri	Ver
Angustiado	0.00	0.00	0.00	0.60	0.60	0.00	0.00	0.00
Depresivo	0.00	0.00	0.00	0.00	0.30	0.00	0.60	0.40
Hipocondríaco	0.00	0.00	0.50	0.70	0.00	0.00	0.00	0.00
Maniaco	0.60	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Fóbico	0.00	0.00	0.00	0.60	0.30	0.00	0.00	0.00
Normal	0.10	0.00	0.00	0.10	0.10	0.10	0.10	0.20
Leyenda: Ale(gría), Des(dén), Ira, Mie(do), Pre(ocupación), Sor(presa), Tri(steza), Ver(güenza)								

Tabla 4.2: Posibles tipos de personalidad de un Agente Inteligente.

En general, el estado emocional de cualquier agente conversacional, independientemente del dominio de aplicación, dependerá de las variables siguientes:

- Personalidad: determina el modo en el que los agentes perciben todos los acontecimientos que pasan en el mundo y como sus acciones son reflejadas por emociones.

- Salud: el estado emocional de un agente está influenciado por de su salud, como ocurre con las personas. Por ejemplo, cuando estamos enfermos o cansados somos más propensos a enfadarnos con los demás. Por lo tanto, los agentes deben actuar de un modo similar.

- Ambiente: esta variable controlará la interacción del agente con todos los individuos del mundo (otros agentes o incluso personas) que podría influir sobre su estado emocional.

- Tipo de conversación: aunque la clase de conversación con el agente dependa del dominio específico, esta variable debe aparecer en todas las reglas borrosas de los sistemas. Comentarios positivos o negativos pueden variar, por ejemplo, los niveles de alegría o enfado del agente.

- Memoria (veces que cada pregunta se repite): uno de los componentes más importantes que un agente debe tener es el registro conversacional. Sería inútil que el agente reaccionara perfectamente al afrontar un estímulo y mostrara la misma emoción si después del repetir el mismo estímulo una y otra vez la respuesta fuera siempre la misma. Por eso, es necesario tomar el registro conversacional en consideración de modo que, por ejemplo, el agente pueda hacerse el enfadado si piensa que alguien está tomándole el pelo repitiéndole la misma pregunta una y otra vez.

- Gestor de diálogo: si la estructura del diálogo ha sido prefijada, el sistema debe controlar el flujo conversacional.

Basándose en estos atributos, en el trabajo de Eisman et al. [EISMAN09] se aplican al ejemplo de un Agente Inteligente que hace las veces de paciente virtual, de manera que se definen como variables de entrada al sistema de lógica borrosa las siguientes:

- Tipo de pregunta: Los tipos de pregunta que el doctor puede realizar al paciente están relacionados con la razón de la cita, un síntoma particular, preguntas de contenido sexual, comentarios positivos o negativos o incluso puede mantenerse callado.

- Mismo síntoma en otro lugar u otro síntoma en el mismo lugar: su valor puede ser sí o no.

- Intensidad del síntoma (entre 0 y 100), frecuencia (entre 0 y 10 veces al día), duración (de 0 a 30 días) e importancia (entre 0 y 10).

- Entorno: aspectos del paciente que pueden influir en su estado emocional, como si un pariente ha experimentado previamente la enfermedad o ha muerto de eso, lo que podría incrementar los niveles de atributos como “miedo” y “preocupación”

- Gestor de diálogos: controla el flujo de la conversación doctor-paciente sobre la base de un conjunto de estados: introducción, explicación de la información, diagnóstico de enfermedad, despedida y silencio. Esto permite al paciente, entre otras situaciones, enfadarse si el doctor hace un diagnóstico después de que el paciente se opere o si el doctor se despide sin dar un diagnóstico.

En cuanto a la salida del sistema, se determina mediante una expresión lingüística que contiene un nivel de variación específico para cada atributo emocional, en vez de emplear una fórmula. Esta variación está definida por una de las ocho etiquetas siguientes: negativo alto, negativo medio, negativo bajo, cero, positivo bajo, positivo medio, positivo alto, y personalidad.

Por último, para cada atributo, se definen reglas borrosas con conjuntos de salida trapezoidales. Por ejemplo, para el atributo emocional “miedo”, se puede definir la siguiente regla:

SI “Tipo de Pregunta” = Síntoma AND “Intensidad” = Alta, ENTONCES “Variación” = Positivo bajo.

4.1.2. Agentes Inteligentes comerciales en entornos web.

Además de los Agentes Inteligentes para investigación, para los cuales ya existen un buen número de aplicaciones comerciales, también existen ya algunos Agentes Inteligentes comerciales en páginas web, siendo el pionero de los existentes en español, y posiblemente uno de los mejores, el asistente virtual de Caja Madrid, llamado Bea. Fue creado en 2005 por la empresa Asistentes Virtuales. Este agente, aparte de tener una interfaz gráfica agradable, responde a muchas de las dudas correspondientes a su ámbito (e incluso a algunas que no lo

son) en forma de conversación y redireccionando al usuario a la página que considera más parecida a la consulta.

En la Figura 4.5 se observa la interfaz gráfica de Bea.

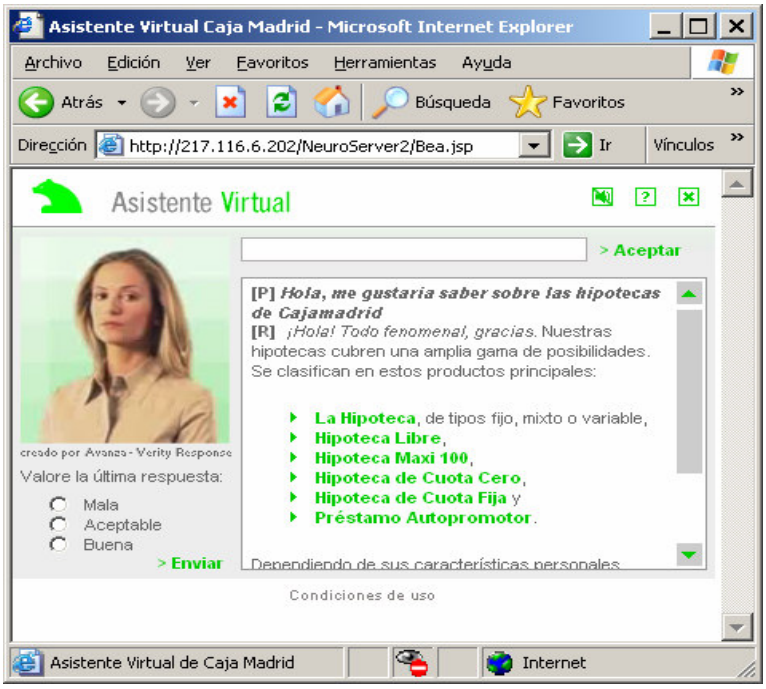


Figura 4.5. Interfaz gráfica de Bea, asistente virtual de Cajamadrid.

En la propia página web de Caja Madrid definen algunas características de Bea. Según Caja Madrid, el asistente entiende mejor preguntas realizadas en lenguaje natural con una sintaxis sencilla y siempre de una en una. Si en algún caso no conoce el dato solicitado, dirige al usuario a páginas del Grupo Caja Madrid en las que se contesta la pregunta. Si, por el contrario, es la pregunta lo que no entiende, indica al usuario que no ha entendido la pregunta [CAJAMADRID09].

Por otra parte, también según la misma página web, el asistente ha sido sometido a más de 150.000 conversaciones y 350.000 preguntas lo que, evidentemente, es un entrenamiento importante (y trabajoso). Así mismo, se ofrece al usuario la posibilidad de valorar la respuesta, lo que supone, a fin de cuentas, la realimentación del sistema.

La empresa Asistentes Virtuales también ha diseñado con la misma filosofía los Agentes Inteligentes de las compañías de seguros Groupama (cuyo Agente Inteligente se llama Beatriz) o Clickseguros (Clara), entre otras, y tiene varios agentes más previstos [ASISTENTES09]. Se pueden ver los dos últimos agentes mencionados en la Figura 4.6.

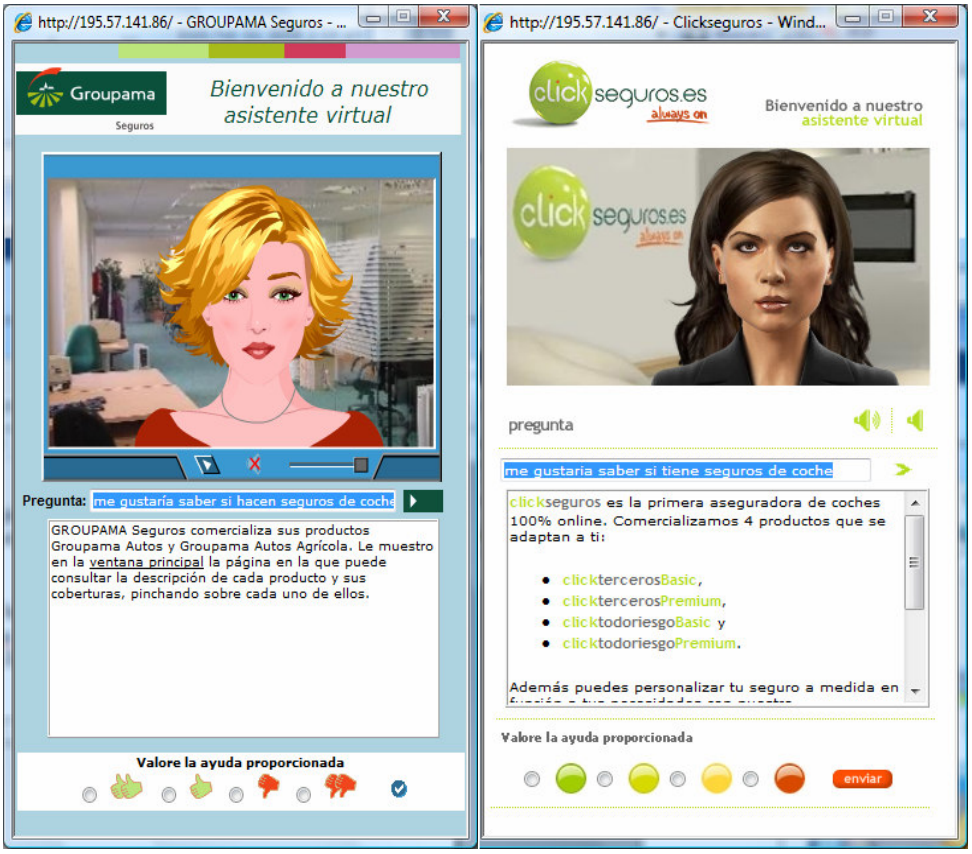


Figura 4.6. Interfaces gráficas de Beatriz y Clara (Agentes Inteligentes de Groupama y Clickseguros, respectivamente).

Además, existen otros Agentes Inteligentes comerciales en español en la WWW, entre los cuales podríamos destacar los de Telefónica e Ikea. Estos Agentes Inteligentes se muestran en la Figura 4.7.

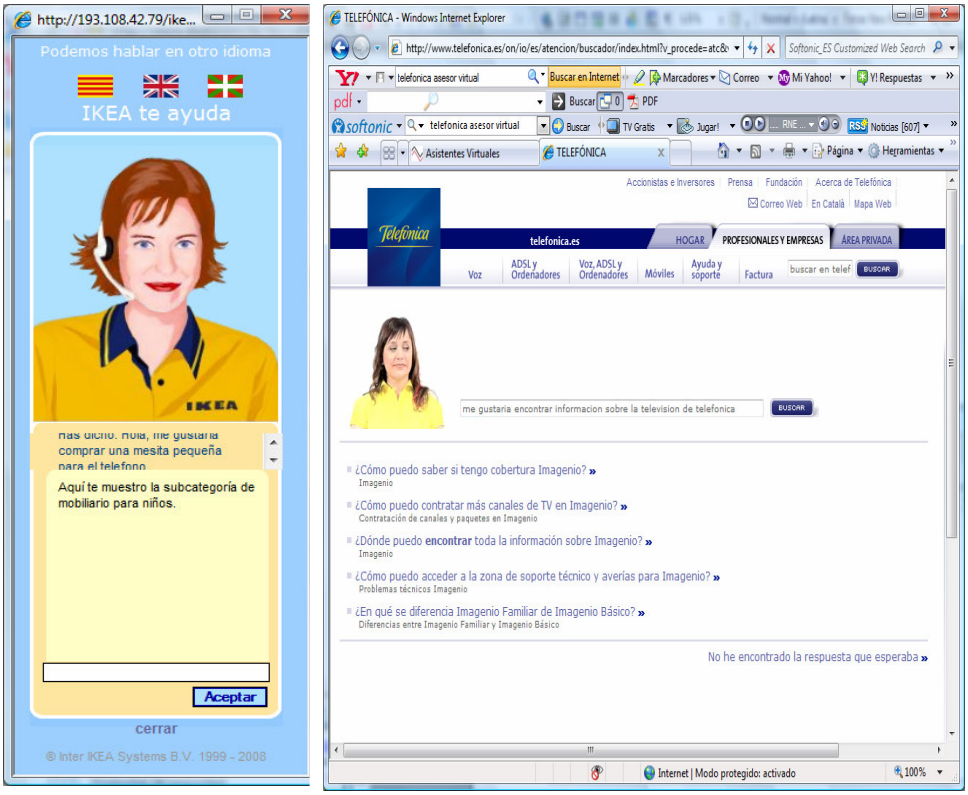


Figura 4.7. Interfaces gráficas de los Agentes Inteligentes de Ikea y Telefónica.

Por último, la empresa sevillana Indisys realiza Agentes Inteligentes para distintas empresas, además de implementar distintas soluciones relacionadas con el Lenguaje Natural, el diálogo inteligente o interfaces de voz [INDISYS09]. Cabe destacar la implementación de un Agente Inteligente para la página web de la cámara de comercio de Sevilla (http://www.camaradesevilla.com). En la Figura 4.8, se observa la demo de la interfaz gráfica del Agente Inteligente disponible en la web de Indisys.

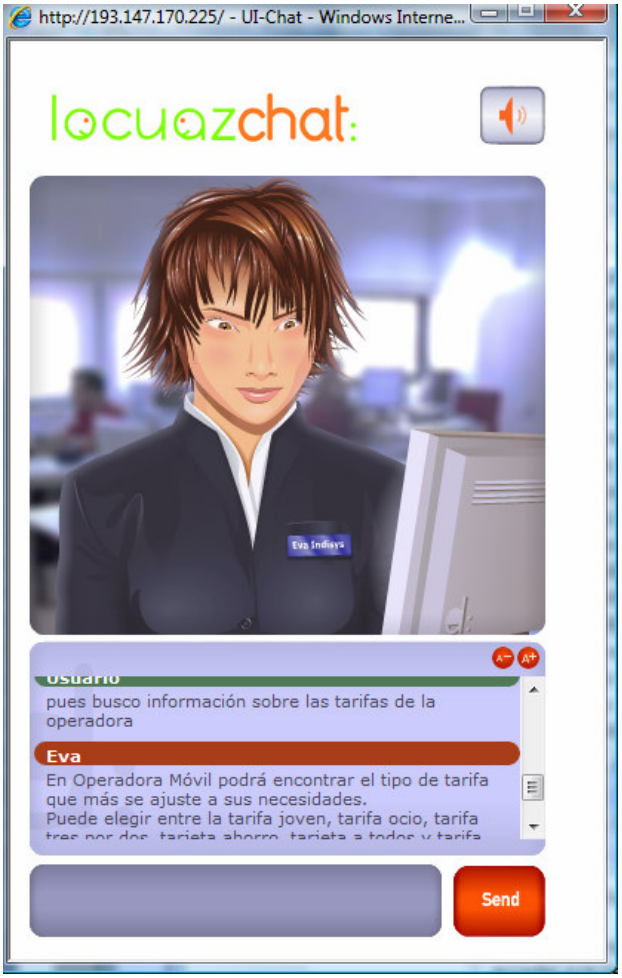


Figura 4.8. Interfaz gráfica de Bea, asistente virtual creada por Indisys.

El problema principal de la mayoría de estos agentes radica, en general, en su falta de flexibilidad. Reaccionan bien ante preguntas bien realizadas, pero su respuesta no suele ser demasiado satisfactoria ante variaciones en las consultas. Este aspecto representa un inconveniente importante cuando la información es abundante y el usuario no es un experto en la materia.

Por otra parte, y relacionado también con esta falta de flexibilidad, muchos de estos agentes no proporcionan al usuario más que una respuesta a su consulta. En opinión del autor de esta tesis, es interesante dar al usuario la posibilidad de que elija entre algunas opciones más, puesto que en los portales de internet existe mucha información que está relacionada con la información requerida y que también podría resultar de interés para el usuario.

4.2. Caracterización del Agente Inteligente propuesto.

Como se exponía en el punto anterior de esta tesis, los Agentes Inteligentes para IE e IR existentes adolecen de una importante falta de flexibilidad. En general, suelen responder bien ante preguntas muy concretas, pero cuando el usuario no es concreto en sus peticiones o consultas (en inglés, *queries*), lo cual ocurre a menudo cuando el usuario no es un experto en la materia o no conoce con precisión la terminología utilizada en el dominio de conocimiento, las respuestas no son tan acertadas. Además, estos sistemas suelen centrar sus respuestas, es decir, no ofrecen ninguna alternativa. Esto, según la opinión del autor de esta tesis, no es un planteamiento correcto en entornos tales como portales web, en los que existe una gran correlación entre la información contenida en ellos, por lo que varias respuestas podrían interesar a un usuario.

Para solucionar estos inconvenientes, en esta tesis se propone un método general de Extracción de Información mediante el uso de la Lógica Borrosa para un Agente Inteligente, con lo que se aprovecha la flexibilidad que proporciona esta herramienta matemática, el cual se describe en este capítulo 4, a continuación. En el capítulo 5 se particulariza este método para extraer la información de un portal web basándose en el Modelo de Espacio Vectorial (VSM) y en la creación de términos índice basados en palabras claves, ya descritos en el Capítulo 2. Hay que tener en cuenta que la información contenida en una página web es heterogénea e imprecisa en la mayoría de los casos, con lo que la FL es de gran utilidad para encontrar la información requerida. Así mismo, se propone un método de consulta basado en FL mediante una interfaz con la que se puede interactuar en Lenguaje Natural.

En esta sección se describe como debe ser el Agente Inteligente desde el punto de vista de su arquitectura y funcionamiento, sus objetivos y la forma en que debe operar, teniendo en cuenta la estructura jerárquica que posee la información contenida en diversos conjuntos de conocimiento (como, por ejemplo, en un portal web) y describiendo los métodos de elección de los términos índice y la manera en la que se produce la asignación de pesos (*Term Weighting*, TW) a estos términos índice. En el capítulo 6 también se introduce un nuevo esquema de introducción de pesos basado igualmente en la FL, el cual obtiene mejores resultados que el método clásico.

4.2.1. Objetivos del Agente Inteligente.

El objetivo principal debe ser el diseño de un Agente Inteligente que permita una fácil extracción de la información relevante de, en general, cualquier conjunto de conocimiento. Esto será posible mediante una interfaz que permita consultas de los usuarios en Lenguaje Natural acerca de los contenidos y que también responda a estas consultas en Lenguaje Natural. Así mismo se debe contemplar la posibilidad de que el Agente Inteligente redireccione al usuario a la dirección que estime más probable como respuesta a la consulta del usuario [ROPERO07b].

Por tanto, el sistema diseñado debe permitir a los usuarios encontrar posibles respuestas a sus búsquedas en un conjunto de conocimiento de gran tamaño. Con este fin, el conjunto de conocimiento completo debe ser clasificado en diferentes objetos. Estos objetos deben representar las posibles respuestas a las consultas de usuario (*user consultations*), organizadas en grupos jerárquicos. A cada uno de estos objetos se le asigna una o varias preguntas tipo o preguntas estándar (*standard questions*), cuyas respuestas son estos objetos. Según la naturaleza del objeto considerado, se pueden definir distintas clases de preguntas tipo:

- Una única pregunta tipo principal, que es la que mejor define al objeto en observación. Esta pregunta tipo siempre debe existir para cada objeto, siendo las otras clases de preguntas tipo opcionales.

- Preguntas tipo que tengan en cuenta sinónimos de algunas de las palabras utilizadas en la pregunta tipo principal, siempre y cuando estas tengan importancia en la consulta de usuario (por ejemplo, "password" como sinónimo de "clave").

- Preguntas tipo que estimen la posibilidad de que un usuario haga una consulta relacionada con un objeto, pero lo haga de manera imprecisa o que no conozca la jerga empleada (por ejemplo, "rotura de una mesa" por "servicio de mantenimiento").

- Preguntas tipo que tengan relación con la pregunta tipo principal asociada al objeto, pero sean más concretas, como podría ser un listado (por ejemplo, "me gustaría saber el plan de estudios de Ingeniería Informática" por "me gustaría saber el plan de estudios de una titulación de la Universidad de Sevilla").

- Preguntas tipo creadas por realimentación del sistema, es decir, considerar las consultas de usuario más habituales como preguntas tipo.

Posteriormente, de las preguntas tipo definidas se extraen una serie de términos índice (*index terms*), con el fin de diferenciar unos objetos de otros. Finalmente es necesario asignar una serie de pesos a cada uno de estos términos para cada nivel de jerarquía de acuerdo con su importancia en cada una de las preguntas tipo. Esta asignación se hace de acuerdo con un esquema basado en el VSM, siendo estos pesos las entradas a un sistema de lógica borrosa. Este sistema de FL debe devolver al usuario los objetos cuya/s pregunta/s tipo correspondientes sean más parecidas a la consulta de usuario. El proceso completo, junto con otros conceptos que se definen a continuación, están ilustrados en la Figura 4.14.

Por otra parte, una vez que el contenido del conjunto de conocimiento (por ejemplo, un portal web, como se muestra en el capítulo 5 de esta tesis) sea gestionado por el Agente Inteligente, el Agente Inteligente puede realizar otras funciones:

- Humanización de la interfaz. Una vez obtenido un funcionamiento satisfactorio en la gestión de la información del portal, se podría proceder a "humanizar" la interfaz trabajando en los aspectos conversacionales del Agente Inteligente. Se puede proceder al tratamiento de la parte de la consulta del usuario que no corresponda a contenidos de la web creando un modelo conversacional capaz de simular las respuestas que daría una persona. Por ejemplo, ante las frases "hola", "buenos días"

o “buenas tardes” se podría responder “Hola. Bienvenido a la web de la Universidad de Sevilla. ¿En qué puedo ayudarle?”. Una manera de realizar esta tarea es considerar también estas posibles respuestas como objetos.

- Gestión de errores: un “bunos días” puede ser entendido como un “buenos días”.

- Gestión de otros aspectos del modelo conversacional, como por ejemplo el uso de palabras obscenas, pudiendo considerar también estas como palabras clave que den lugar a respuestas estimadas como objetos.

- Acceso a otros contenidos relacionados con la información del portal web, pero que se encuentren fuera de este. Posible redireccionamiento a estas páginas.

Además, se puede crear un modelo conversacional capaz de simular las respuestas que daría una persona, aunque dada la complejidad de la tarea, puede ser necesario recurrir al asesoramiento de un lingüista para el desarrollo de esta.

4.2.2. Estructura jerárquica.

Dado que el objetivo del sistema es encontrar las posibles respuestas a las consultas de usuario, devolviendo no solo la que mejor se adecue a sus necesidades, sino también aquellas que estén relacionadas (recordemos que las consultas están sujetas a posibles imprecisiones), parece lógico establecer una clasificación basada en un cierto criterio o grupo de criterios. De esta forma, el usuario podría obtener no solo el objeto que más se adecuara a su consulta sino aquellos que estén más íntimamente relacionados [ROPERO07a].

Por ejemplo, en el caso que particular que contemplamos en el capítulo 5 de esta tesis, el de la IE en un portal web, una clasificación jerárquica es completamente apropiada, puesto que un portal web también tiene una estructura jerárquica. Por consiguiente, es necesario identificar página web y objeto, es decir, cada página web del portal es considerada un objeto, siendo estos objetos agrupados en la estructura jerárquica mencionada. También es posible asignar varios objetos a una misma página web si la información que contiene es lo suficientemente variada.

Un portal se divide pues en N niveles, siendo el tamaño de cada uno de estos variable (dado que así ocurre en un portal web, en el que cada sección tiene también un tamaño variable). Cada objeto pertenece a un conjunto de nivel de N, mientras que varios conjuntos de nivel N con ciertos rasgos comunes se agrupan en un conjunto de nivel N-1.

Análogamente, varios conjuntos de nivel de N-1 con características comunes se juntan en otros grupos que forman un conjunto de nivel de N-2 y así sucesivamente hasta llegar al nivel 0 (N-N), que representa la totalidad del conocimiento acumulado, que es toda la información de la que se dispone, relevante o no para un usuario. El valor de la información específica no es fácil de separar del conocimiento acumulado, dado que, mucho del conocimiento que los individuos acumulan y almacenan no está especialmente dirigido a un problema en particular [GREENES06]. La clasificación propuesta se muestra en la Figura 4.9 formando una estructura arborescente.

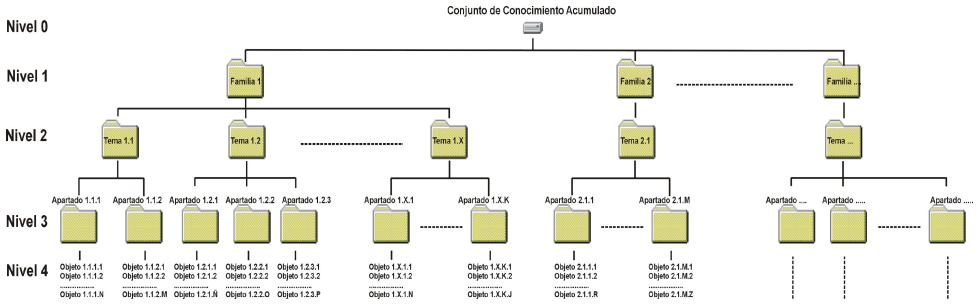


Figura 4.9. Estructura jerárquica arborescente.

La decisión acerca de la asignación de grupos debe ser adoptada por un Ingeniero del Conocimiento, basándose en sus criterios de experiencia respecto al manejo del conjunto de conocimiento acumulado. En el caso de un portal web, podría ser el Administrador o Desarrollador del portal.

Así mismo, es necesario almacenar tanto los Objetos (es decir, las posibles respuestas a las consultas de usuario, Figura 4.10) como la estructura jerárquica del conjunto de conocimiento en bases de datos (Figura 4.12).

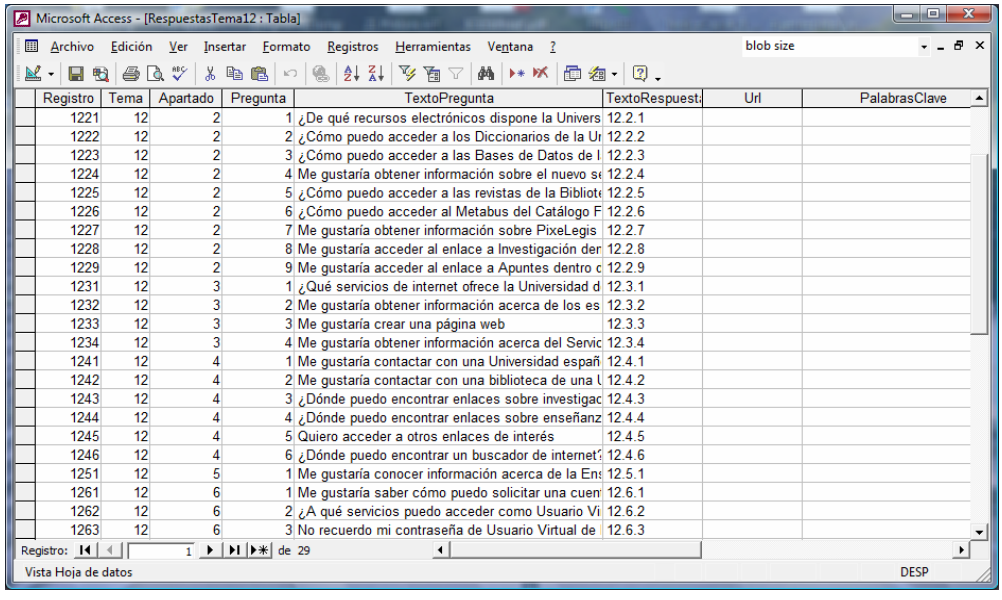


Figura 4.10. Base de datos con las respuestas a las consultas de usuario.

En la Figura 4.10 se puede observar que cada Objeto tiene una referencia a un texto que servirá como respuesta a la consulta de usuario. Cada Objeto o respuesta está asociado a una o

varias preguntas tipo. La respuesta a la pregunta tipo que más se parezca a la consulta de usuario será la respuesta que se dará al usuario del sistema, pudiéndose así mismo proporcionar otras respuestas similares y que también pudieran interesar igualmente al usuario. Este esquema se refleja en la Figura 4.11.

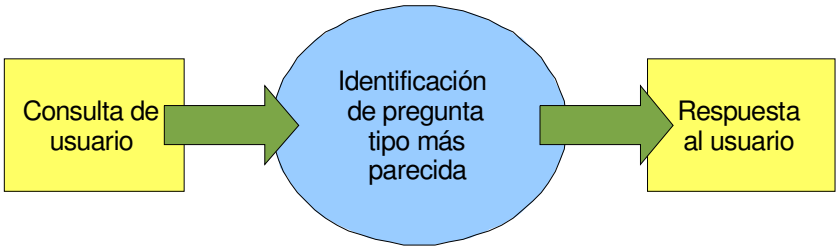
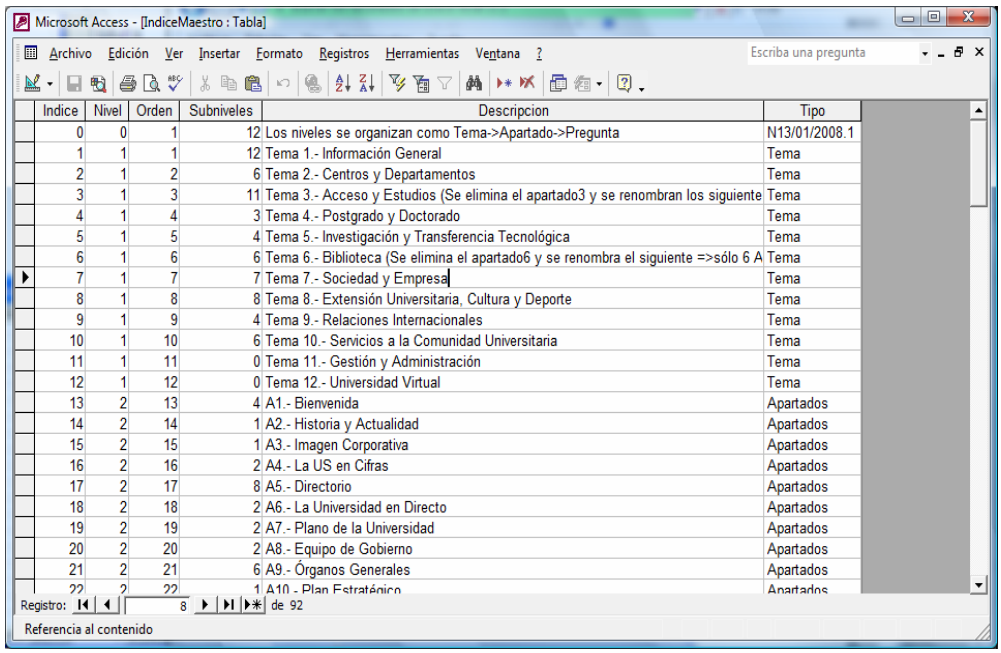


Figura 4.11. Esquema de la respuesta a una consulta de usuario.

La Figura 4.12 muestra la organización del conjunto de conocimiento en distintos niveles jerárquicos (en este caso tres niveles), en los que cada subconjunto de conocimiento de nivel superior está dividido en varios subconjuntos de conocimiento de nivel inferior.



Indice	Nivel	Orden	Subniveles	Descripcion	Tipo
0	0	1	12	Los niveles se organizan como Tema->Apartado->Pregunta	N13/01/2008.1
1	1	1	12	Tema 1.- Información General	Tema
2	1	2	6	Tema 2.- Centros y Departamentos	Tema
3	1	3	11	Tema 3.- Acceso y Estudios (Se elimina el apartado3 y se renombran los siguiente	Tema
4	1	4	3	Tema 4.- Postgrado y Doctorado	Tema
5	1	5	4	Tema 5.- Investigación y Transferencia Tecnológica	Tema
6	1	6	6	Tema 6.- Biblioteca (Se elimina el apartado6 y se renombra el siguiente =>sólo 6 A	Tema
7	1	7	7	Tema 7.- Sociedad y Empresa	Tema
8	1	8	8	Tema 8.- Extensión Universitaria, Cultura y Deporte	Tema
9	1	9	4	Tema 9.- Relaciones Internacionales	Tema
10	1	10	6	Tema 10.- Servicios a la Comunidad Universitaria	Tema
11	1	11	0	Tema 11.- Gestión y Administración	Tema
12	1	12	0	Tema 12.- Universidad Virtual	Tema
13	2	13	4	A1.- Bienvenida	Apartados
14	2	14	1	A2.- Historia y Actualidad	Apartados
15	2	15	1	A3.- Imagen Corporativa	Apartados
16	2	16	2	A4.- La US en Cifras	Apartados
17	2	17	8	A5.- Directorio	Apartados
18	2	18	2	A6.- La Universidad en Directo	Apartados
19	2	19	2	A7.- Plano de la Universidad	Apartados
20	2	20	2	A8.- Equipo de Gobierno	Apartados
21	2	21	6	A9.- Órganos Generales	Apartados
22	2	22	1	A10.- Plan Estratégico	Apartados

Figura 4.12. Base de datos que contiene la estructura jerárquica del conjunto de conocimiento.

4.2.3. Construcción del Agente Inteligente.

Para construir el Agente Inteligente, en primer lugar hay que tener en cuenta que las consultas de usuario se realizan en Lenguaje Natural (*Natural Language*, NL), por lo que, como se ha mencionado antes, se aprovechará esta particularidad para representar cada objeto mediante una o varias preguntas tipo, formuladas en NL. Posteriormente, es necesario extraer una serie de términos índice de dichas preguntas tipo y, por último, asignar pesos a estos términos índice en relación con la importancia de estos en el objeto al que representan [GÓMEZ06].

Representación en Lenguaje Natural (*Natural Language*, NL).

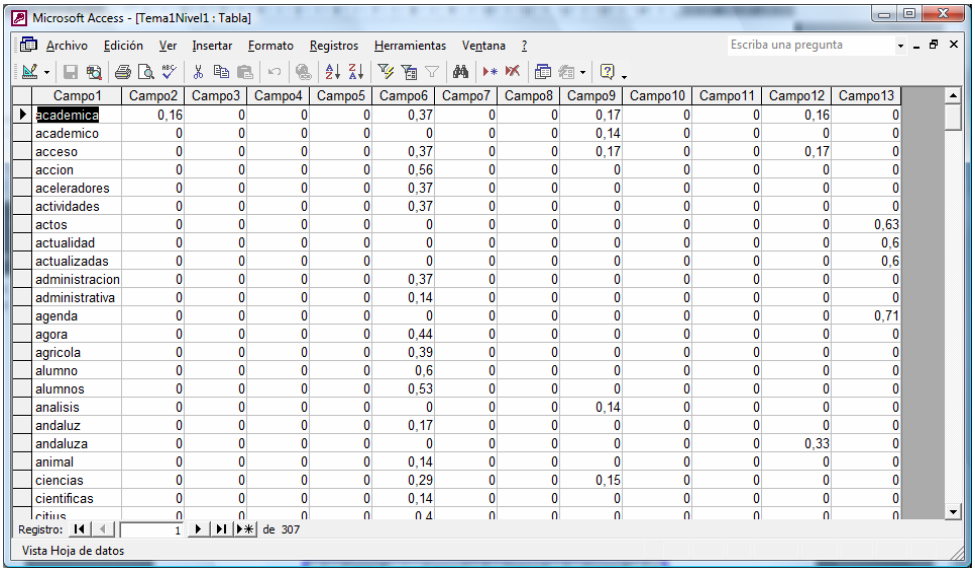
Como se ha indicado anteriormente, es necesario asociar una representación en NL a cada objeto del conjunto de conocimiento acumulado. El proceso consiste en dos pasos:

- El primer paso debe ser dividir el conocimiento en objetos. A cada objeto se le asignan una o varias preguntas en NL (a las cuales se ha denominado anteriormente preguntas tipo), cuya respuesta debe representar al objeto deseado. A estas preguntas se les ha denominado preguntas estándar o preguntas tipo. La experiencia del Ingeniero de Conocimiento que define estas preguntas tipo con respeto a la jerga en el dominio del conjunto de conocimiento es importante dado que mientras mayor sea su conocimiento, mayor será la fiabilidad de las preguntas tipo propuestas para la representación del objeto. Esto es debido a que estas son más parecidas a las posibles consultas de usuario. Sin embargo, es posible redefinir representaciones de objeto o añadir nuevas definiciones que analicen futuras consultas de usuario y estudiar su sintaxis y su vocabulario. Por consiguiente, el sistema puede refinar su conocimiento mediante el mencionado Ingeniero de Conocimiento. Además, el hecho de trabajar con FL proporcionará igualmente una mayor flexibilidad.

- El segundo paso es la selección de términos índice, los cuales son extraídos de las preguntas tipo y que representan los términos más relacionados de dichas preguntas tipo con el objeto representado.

Elección de términos índice.

Una vez que se han definido las preguntas tipo, se extraen de ellas los denominados términos índice. Se han identificado estos términos índice con palabras clave (*keywords*), aunque también se podría trabajar con términos compuestos (*joint terms*). Los términos índice extraídos son los que mejor representan a una pregunta tipo. Estas palabras deben ser almacenadas en una base de datos junto sus correspondientes pesos, correspondientes a cada uno de los niveles jerárquicos. Se puede ver un ejemplo en la Figura 4.13, en la que diferentes palabras clave tienen asociadas un peso para cada uno de los subconjuntos del nivel jerárquico considerado.



Campo1	Campo2	Campo3	Campo4	Campo5	Campo6	Campo7	Campo8	Campo9	Campo10	Campo11	Campo12	Campo13
academica	0,16	0	0	0	0,37	0	0	0,17	0	0	0,16	0
academico	0	0	0	0	0	0	0	0,14	0	0	0	0
acceso	0	0	0	0	0,37	0	0	0,17	0	0	0,17	0
accion	0	0	0	0	0,56	0	0	0	0	0	0	0
aceleradores	0	0	0	0	0,37	0	0	0	0	0	0	0
actividades	0	0	0	0	0,37	0	0	0	0	0	0	0
actos	0	0	0	0	0	0	0	0	0	0	0	0,63
actualidad	0	0	0	0	0	0	0	0	0	0	0	0,6
actualizadas	0	0	0	0	0	0	0	0	0	0	0	0,6
administracion	0	0	0	0	0,37	0	0	0	0	0	0	0
administrativa	0	0	0	0	0,14	0	0	0	0	0	0	0
agenda	0	0	0	0	0	0	0	0	0	0	0	0,71
agora	0	0	0	0	0,44	0	0	0	0	0	0	0
agricola	0	0	0	0	0,39	0	0	0	0	0	0	0
alumno	0	0	0	0	0,6	0	0	0	0	0	0	0
alumnos	0	0	0	0	0,53	0	0	0	0	0	0	0
analisis	0	0	0	0	0	0	0	0,14	0	0	0	0
andaluz	0	0	0	0	0,17	0	0	0	0	0	0	0
andaluza	0	0	0	0	0	0	0	0	0	0	0,33	0
animal	0	0	0	0	0,14	0	0	0	0	0	0	0
ciencias	0	0	0	0	0,29	0	0	0,15	0	0	0	0
cientificas	0	0	0	0	0,14	0	0	0	0	0	0	0
citius	0	0	0	0	0,4	0	0	0	0	0	0	0

Figura 4.13. Base de datos de términos índice.

Asignación de pesos (*Term Weighting*, TW).

En apartados anteriores de esta tesis, se comentaba la necesidad de que existan una serie de coeficientes o pesos asociados a cada palabra clave o término índice y que los valores de estos debían estar relacionados de alguna manera con la importancia que tuviera el término índice en el conjunto de conocimiento al que representara (es decir, a la importancia del término en cada nivel de la estructura jerárquica). Podemos considerar principalmente dos métodos para la asignación de pesos.

- Dejar que un experto en la materia evalúe intuitivamente la importancia de los coeficientes. Este método es sencillo, pero tiene la desventaja de que depende exclusivamente del ingeniero del conocimiento, es muy subjetivo y no se puede automatizar.

- Automatizar la generación de coeficientes mediante una serie de reglas.

Habida cuenta de la ingente cantidad de información que puede haber en un portal web, nos inclinamos en esta tesis por la segunda opción y se propone un método basado en el VSM. El método más ampliamente usado para TW es el método TF-IDF, ya explicado en el capítulo 2 de esta tesis. Sin embargo, en esta tesis se propone una modificación del método basada en el uso de FL, la cual proporciona una serie de ventajas que serán descritas junto con el propio método en el capítulo 6 de la tesis.

Cada término índice se asocia con su peso correspondiente. Este peso tiene un valor entre 0 y 1 y depende de la importancia del término en cada nivel jerárquico. Mientras mayor es la importancia del término en un nivel, mayor es el peso de dicho término. Además, hay que tener en cuenta que el peso de término no tiene por qué ser el mismo para cada nivel jerárquico, dado

que la importancia de una palabra para distinguir, por ejemplo, una sección de las demás puede ser muy diferente de su importancia para distinguir entre dos objetos.

En resumen, el proceso completo de constitución del Agente Inteligente se puede resumir en la Figura 4.14.

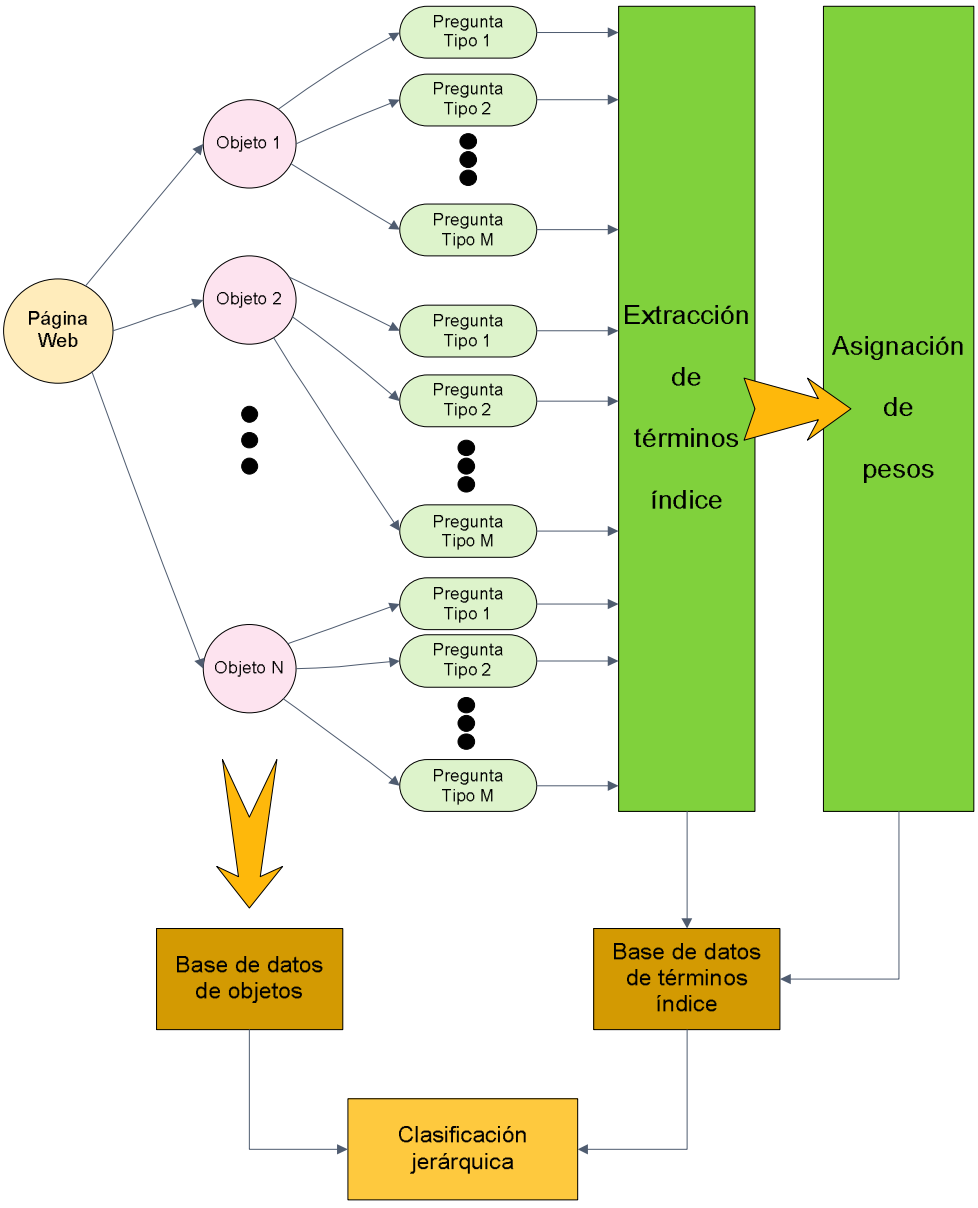


Figura 4.14. Constitución del Agente Inteligente.

Cada página web (o, en general, cualquier porción de información) se divide en uno o varios objetos según la cantidad de información que contengan. En realidad, cada Objeto es una respuesta a cada posible consulta de usuario. Como es posible que varias preguntas conduzcan a una misma respuesta, se pueden definir también una o varias preguntas tipo para un mismo Objeto.

Cuando se han definido las preguntas tipo, es necesario extraer los términos índice o palabras clave y asignarles un peso. Tanto los términos índice con sus pesos correspondientes como los objetos deben ser almacenados en respectivas bases de datos que, a su vez, constituyen una estructura jerárquica.

4.2.4. Modo de operación del Agente Inteligente.

Una vez constituido el Agente Inteligente, es necesario saber cuál es su modo de operación, es decir, como funciona una vez que recibe una consulta de usuario. En primer lugar, las palabras clave o términos índice se extraen por comparación con las contenidas en la base de datos correspondiente a estas. Los pesos de estas palabras clave para cada nivel constituyen las entradas a un sistema de lógica borrosa. El funcionamiento de este sistema de lógica borrosa está descrito en el apartado 3 de este capítulo 4 de la tesis.

En este momento es donde entra en juego la estructura jerárquica del conjunto de conocimiento. El conjunto de conocimiento completo, que constituye el nivel jerárquico 0, se divide en subconjuntos de nivel 1, para cada uno de los cuales los términos índice tienen definido un determinado peso en consonancia con su relación con dichos subconjuntos. Estos pesos utilizados como entradas en el motor de inferencia de lógica borrosa proporcionan una salida para cada subconjunto, a la que llamaremos nivel de certeza. Si dicho nivel de certeza es menor que un valor prefijado, denominado umbral, el contenido del subconjunto correspondiente es rechazado. El hecho de empezar por el primer nivel y usar una estructura jerárquica posibilita el rechazo de una gran cantidad de contenido, que ya no deberá ser considerado en futuras búsquedas. A todos los efectos, este hecho proporciona dos ventajas importantes:

- Ahorra tiempo de computación.

- Elimina gran cantidad de “ruido”, es decir, se elimina información que ya no es necesaria y que no será presentada al usuario con toda seguridad.

Para cada subconjunto que supere el umbral mínimo de certeza, se repite el proceso, analizándose los coeficientes de pertenencia correspondientes al nivel 2. Los conjuntos en los que el grado de certeza devuelto por el motor borroso no supera un umbral mínimo son nuevamente rechazados. Si sobrepasan dicho umbral, el proceso se repite y así sucesivamente hasta el último nivel. Las respuestas definitivas corresponden a los elementos de último nivel, es decir, los objetos, cuya certeza supere el umbral definitivo, existiendo la posibilidad de que haya varias respuestas. Mientras más vagas sean las preguntas, más respuestas se obtienen. En la Figura 4.15, se puede observar como sería el proceso completo para una estructura jerárquica de dos niveles, es decir, el conjunto de conocimiento completo (nivel 0), se agrupa en subconjuntos de nivel 1 y estos a su vez se agrupan en subconjuntos de nivel 2. Al ser este el último nivel, estos subconjuntos son los propios objetos. El autor de esta tesis ya reflejó este

proceso en [ROPERO07b].

En el caso plasmado en la Figura 4.15, con dos niveles jerárquicos, el orden en el tendrían lugar los pasos sería el siguiente:

- **Paso 1**: Se produce una consulta de usuario.

- **Paso 2**: Se extraen los términos índice para el nivel jerárquico 1 por comparación con la base de datos de términos índice.

- **Paso 3**: Se extraen los pesos de los términos índice para cada subconjunto correspondiente a este nivel jerárquico. Estos pesos son las entradas de un motor de inferencia borroso, que dará lugar a una salida por cada subconjunto.

- **Paso 4**: Si la salida del motor borroso correspondiente a un determinado subconjunto de conocimiento es menor que un umbral, se descarta el subconjunto completo. En caso contrario, se repite el proceso anterior para los subconjuntos del nivel jerárquico 2 que pertenezcan a los subconjuntos de nivel 1 que superen el umbral de certeza.

- **Paso 5**: Se repite el proceso para todos los niveles jerárquicos hasta llegar al nivel de Objeto. Los Objetos que sobrepasen el último umbral son seleccionados por el sistema como respuesta a la consulta de usuario, presentándose en primer lugar los que presenten un valor de salida más alto y, por tanto, obtengan una mayor certeza.

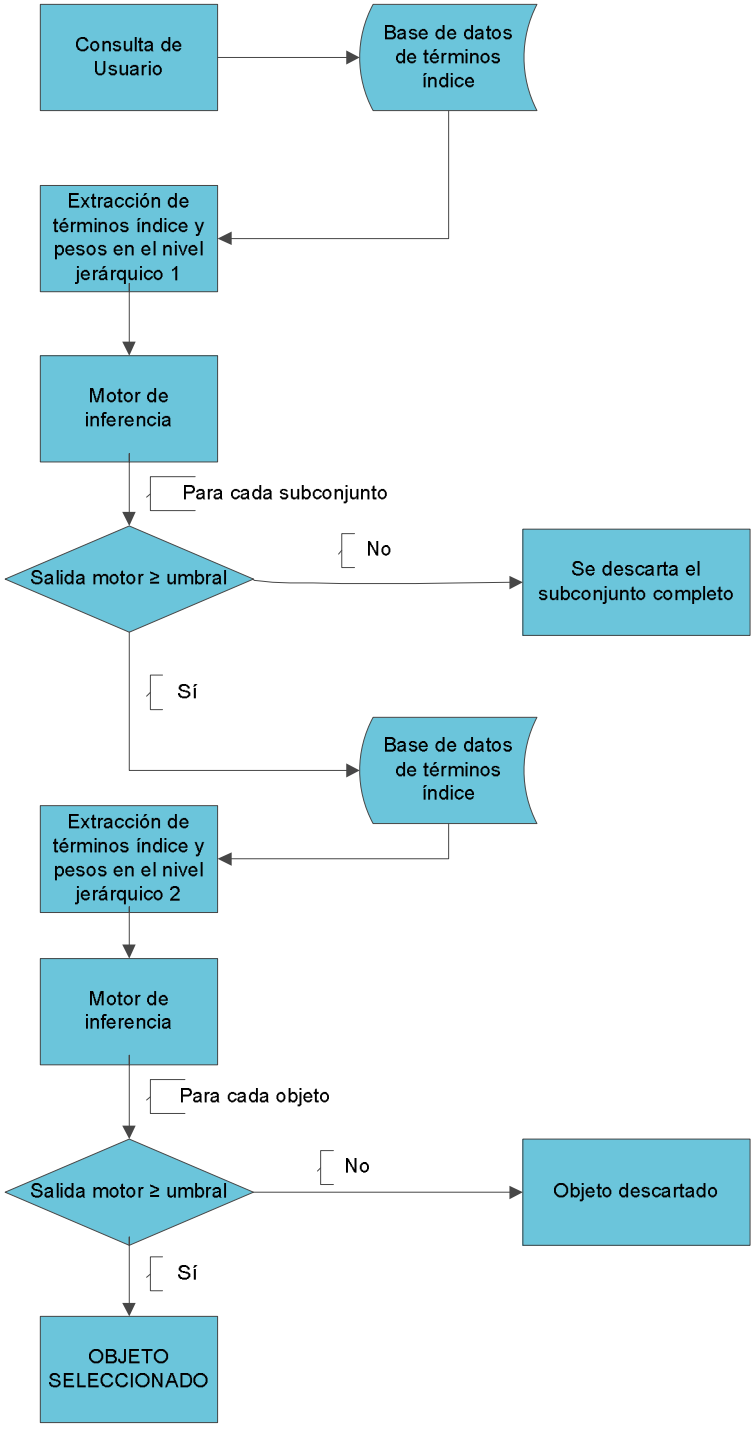


Figura 4.15. Modo de operación del Agente Inteligente con dos niveles jerárquicos.

Un ejemplo real del funcionamiento de este esquema es mostrado en el siguiente capítulo de esta tesis, concretamente en el Apartado 5.2.3., tratándose de un ejemplo realizado para el portal web de la Universidad de Sevilla.

4.3. Sistema de Lógica Borrosa.

El elemento del Agente Inteligente que determina el grado de certeza acerca de que determinadas palabras clave correspondan a cada uno de los posibles subconjuntos del conjunto de conocimiento completo es el motor de inferencia borroso.

El motor de inferencia recibe una serie de entradas y devuelve una salida, la cual corresponde con el anteriormente mencionado nivel de certeza. Como se explicó en el capítulo 3, para que esto ocurra es necesario definir:

- El número de entradas que van al motor de inferencia. Este número depende del número de palabras clave extraídas, por lo que es variable. Así mismo, es conveniente definir un número de entradas máximo para evitar consultas demasiado vagas y que, por tanto, devuelvan demasiados objetos. Estas entradas deben corresponder a los términos índice que tengan un mayor peso para cada uno de los niveles jerárquicos.

- Los conjuntos borrosos de entrada: rango de valores del universo de discurso, número de conjuntos borrosos, y forma y rango de valores de estos.

- Los conjuntos borrosos de salida: rango de valores de la salida, número de conjuntos borrosos, y forma y rango de valores de estos.

- Las reglas por las que se rige el sistema borroso. Estas reglas son del tipo SI … ENTONCES. Ejemplos de estas reglas son:

    - Si todas las entradas son BAJO, la salida es BAJO.
    - Si una entrada es MEDIO y las otras son BAJO, la salida es MEDIO-BAJO.
    - Si dos entradas son MEDIO y las otras son BAJO, la salida es MEDIO-ALTO.
    - Si todas las entradas son MEDIO o una entrada es ALTO, la salida es ALTO.

- Los métodos de realización de las operaciones AND y OR y el método de desborrosificación utilizado.

Una definición más formal de estos parámetros fue realizada en el capítulo 3 de esta tesis.

Existen una serie de herramientas para diseñar sistemas de lógica borrosa. Por ejemplo, es posible utilizar el programa MATLAB, con su toolbox de lógica borrosa [MATHWORKS02], debido a su facilidad de uso, la posibilidad de integrar otros toolboxes y la potencia de la herramienta. El problema principal de esta aplicación es que no exporta código C de la totalidad de lo implementado, cuestión importante para integrar el desarrollo en una aplicación independiente, como es nuestra intención con el Agente Inteligente. Este problema de

exportación de código puede ser resuelto con la herramienta Un-Fuzzy, que es un software para diseño de sistemas de lógica difusa de la Universidad de Colombia, de libre distribución y que exporta código C. Actualmente se encuentra en la versión 1.2 y es una herramienta para el análisis, diseño, simulación e implementación de Sistemas de Lógica Borrosa creado por Oscar G. Duarte para el departamento de Tecnología Eléctrica de la Universidad de Colombia, distribuyéndose gratuitamente a través de la Web de dicho departamento [DUARTE98]. Es destacable de este programa la poca cantidad de recursos que necesita para funcionar, sin necesitar ni siquiera una instalación previa, y que, tanto el programa como el manual y la ayuda en línea, se encuentran en español, algo no muy común en este tipo de herramientas. El funcionamiento del programa Un-Fuzzy está descrito en el Apéndice A de esta tesis.

Sin embargo, por su carácter más intuitivo, se expone a continuación el modo en el que se definirían todos los parámetros correspondientes al motor de inferencia borrosa en MATLAB mediante el uso de su toolbox de lógica borrosa. Dicho toolbox consta de una interfaz gráfica sencilla de usar y que nos permite configurar todos los parámetros de un sistema de lógica borrosa de una manera fácil y cómoda.

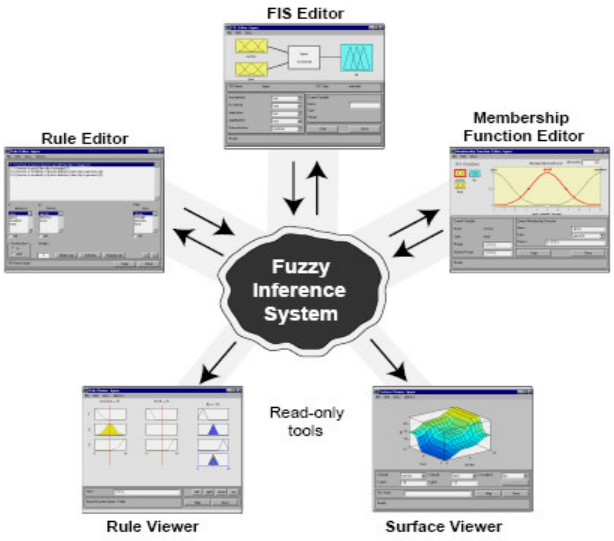


Figura 4.16. Esquema del editor de sistemas de inferencia borrosos.

La interfaz gráfica, como se puede ver en la Figura 4.16 se divide en cinco interfaces, que se describen a continuación, entre los que destacan un editor del sistema, en el que se definen los principales parámetros del sistema de inferencia borroso, un editor de funciones de pertenencia y un editor de reglas borrosas.

La interfaz principal es el editor FIS (*Fuzzy Inference System*, Sistema de Inferencia Borrosa). En este editor se definen los parámetros principales para comenzar la implementación de un sistema de lógica borrosa. Se puede ver su interfaz gráfica en la Figura 4.17.

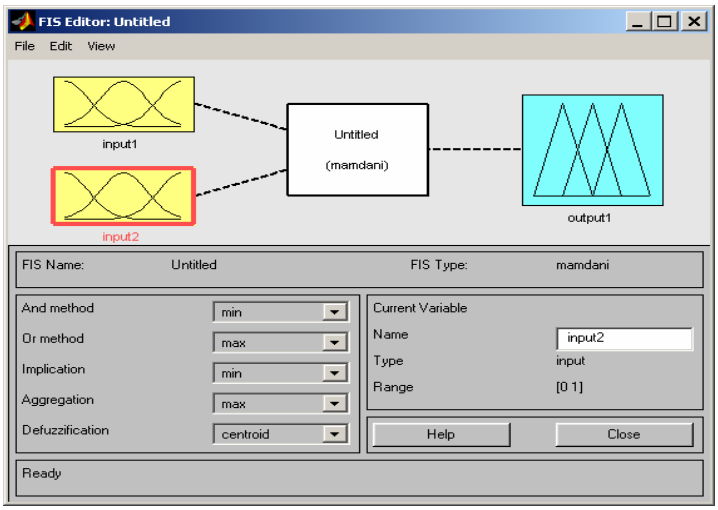


Figura 4.17. Editor FIS de MATLAB.

El editor FIS permite definir:

- El número y nombre de las entradas.
- El número y nombre de las salidas.
- Otros parámetros del sistema, tales como las funciones de inferencia borrosa y el método de desborrosificación.

El editor de funciones de pertenencia o pertenencia comparte algunas características con el editor FIS. De hecho es accesible desde el editor simplemente pinchando en la entrada o salida correspondiente. Este editor de funciones de pertenencia (Figura 4.18) permite:

- Definir el número de conjuntos lógicos para cada variable (entrada o salida).
- Definir el nombre de cada conjunto.
- Definir la función de cada conjunto lógico (triángulo, gaussiana, sigmoidea, etc.).
- Definir el rango de valores de cada función.

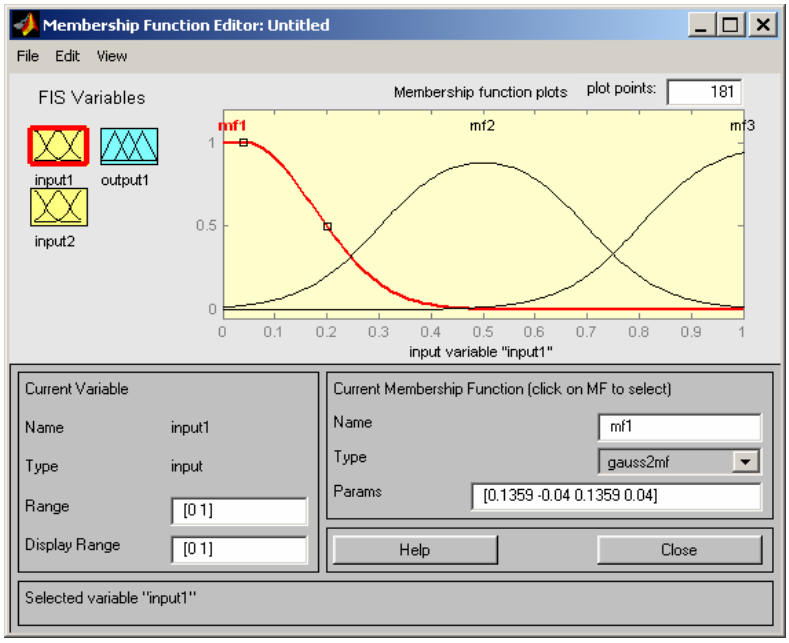


Figura 4.18. Editor de funciones de pertenencia.

En cuanto al editor de reglas, la construcción de reglas de inferencia borrosa usando el editor de reglas gráfico es trivial. No hay más que ver la propia interfaz (Figura 4.19) para comprender esta facilidad.

Se pueden usar operaciones AND y OR para construir reglas del tipo SI… ENTONCES. Además, existe la posibilidad de asignar pesos diferentes a cada regla.

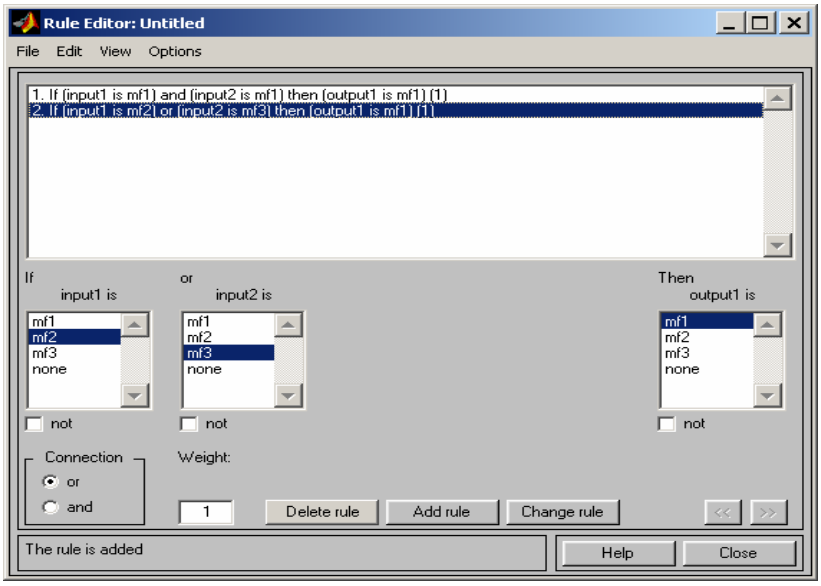


Figura 4.19. Editor de reglas.

Por lo tanto, todos estos parámetros deberán ser tenidos en cuenta a la hora de encontrar una configuración óptima del motor de inferencia, núcleo del Agente Inteligente. El estudio de los parámetros idóneos para el caso de un portal web centrará la atención del Capítulo 5 de esta tesis.

Capítulo 5. Aplicación: diseño de un Agente Inteligente para el portal web de la Universidad de Sevilla.

Hasta ahora, se ha expuesto un método general para la búsqueda y extracción de información, aunque resaltando las aplicaciones que podría tener en entornos web. En este capítulo se remarca la idoneidad de utilizar este método para IE en portales web, realizándose como ejemplo de aplicación particular del método propuesto el diseño de un Agente Inteligente para el portal web de la Universidad de Sevilla.

En el apartado 5.1 se describe brevemente dicho portal de la Universidad de Sevilla, muy adecuado para probar el método de IE propuesto en esta tesis por la gran cantidad de información contenida en él. El apartado 5.2 muestra la estructura del conocimiento en dicho portal y como esta es aprovechada por el Agente Inteligente. A continuación, en el apartado 5.3 se describen los parámetros del sistema de FL, núcleo del Agente Inteligente, y las diferentes pruebas realizadas para encontrar los mejores parámetros del sistema.

Se deja para el Capítulo 6 de esta tesis, por tener la entidad suficiente como para constituir un nuevo capítulo, el nuevo esquema de asignación de pesos propuesto, basado también en lógica borrosa, que mejora al método clásico TF-IDF, utilizado habitualmente para la asignación automática de pesos.

5.1. El portal web de la Universidad de Sevilla.

La Universidad de Sevilla, con 500 años de antigüedad, es la segunda universidad española en número de alumnos, con las siguientes cifras significativas [US09]:

- 57.200 alumnos de 1º y 2º ciclo.
- 7.200 alumnos de 3º ciclo y postgrado.
- 4.400 miembros del Personal Docente e Investigador (PDI).
- 2.300 miembros del Personal de Administración y Servicios (PAS).
- 5 campus en diferentes zonas de la ciudad.
- 25 Centros propios.
- 124 Departamentos.
- 78 titulaciones de 1º y 2º ciclo.
- 102 programas de doctorado.
- 4.400 asignaturas de grado y postgrado.

En cuanto al portal web de la Universidad de Sevilla, en octubre de 2009 ocupaba el puesto 223 de entre las más de 4000 Universidades mundiales censadas en el ranking de Webometrics (séptima de España y entre las 100 mejores de Europa), elaborado por el Laboratorio de Cibermetría del CSIC, el cual mide el impacto web de las Universidades en el mundo, basándose en criterios de páginas recuperadas desde los cuatro grandes motores de búsqueda (Google, Yahoo, Live Search y Exalead), los enlaces externos al sitio, el número de ficheros de información accesibles y el número de artículos y citas de investigación [WEBOMETRICS09].

Por tanto, el portal web de la Universidad de Sevilla se clasifica entre el 10 % de los portales web de universidades más importantes. Además, el portal web de la Universidad de Sevilla recibe 1.346.000 visitas mensuales, lo que supone aproximadamente unas 45.000 visitas diarias [US08]. En la Figura 5.1 se muestra la página de bienvenida del portal de la Universidad de Sevilla (www.us.es).

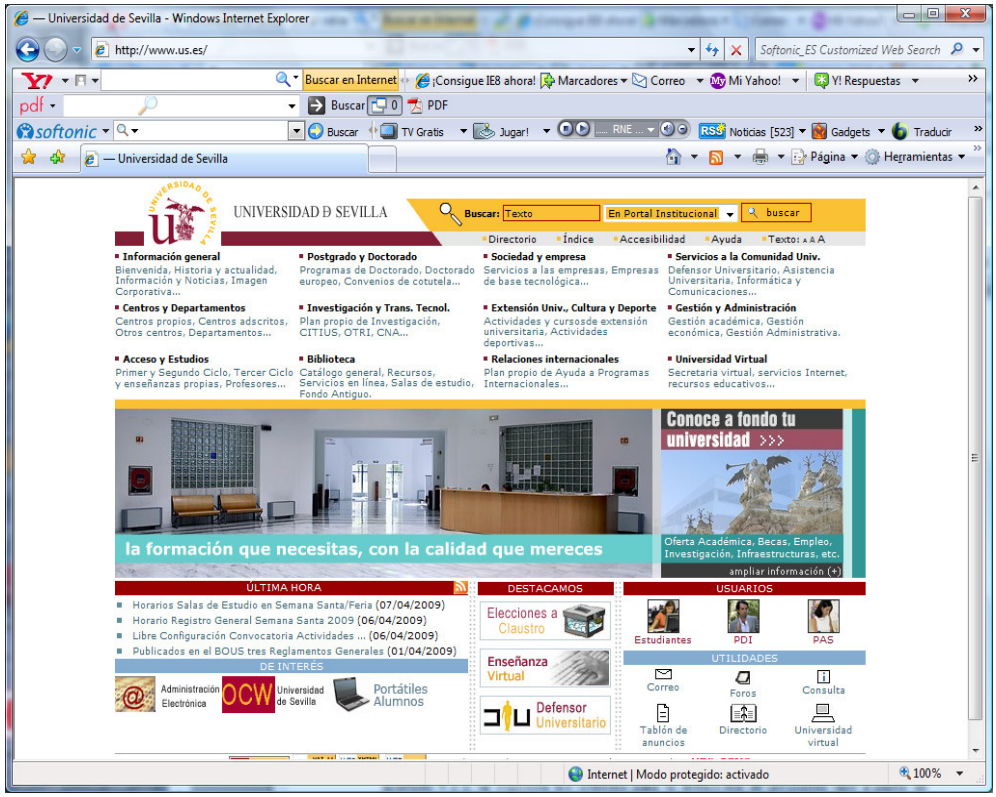


Figura 5.1. Página de bienvenida del portal web de la US.

5.2. Estructura del conjunto de conocimiento.

Dado que en el portal de la Universidad de Sevilla existe una gran cantidad de información, se definieron un total de 253 Objetos agrupados en 12 Temas. Todos estos grupos están compuestos de un número variable de Apartados y Objetos. Además, se definieron 2.107 preguntas tipo a partir de estos 253 Objetos.

Todo esto fue realizado de la forma en la que se expone en los dos siguientes subapartados. En el apartado 5.2.1, se muestra la estructura jerárquica empleada. En el apartado 5.2.2, se exponen los criterios para la definición de preguntas tipo a partir de cada uno de los Objetos y la forma de extraer los términos índice, que en este caso corresponden con palabras clave. Por último, en la sección 5.2.3, se describe el proceso que sigue el Agente Inteligente ante una consulta de usuario.

5.2.1. Estructura jerárquica.

Para llevar a cabo la Extracción de la Información, como se apuntó en el capítulo anterior, lo primero que hay que hacer es identificar página web con Objeto, es decir, cada página del portal es considerada como uno o varios Objetos, dependiendo de la cantidad de información

existente en cada página. Estos Objetos se agrupan en una estructura jerárquica. Cada Objeto sólo es accesible a través de un único camino del árbol de clasificación, es decir, clasificándolo bajo un único criterio (o grupo de criterios). En nuestro caso, se ha aprovechado la estructura jerárquica de la página para dividir el portal en tres niveles, a los que se ha denominado, en orden decreciente jerárquico decreciente, Tema, Apartado y Objeto. El tamaño de cada uno de los niveles es variable: existen 12 Temas, cada uno de ellos con un número de Apartados que está comprendido entre los 4 y los 12. Así mismo, cada uno de los Apartados tiene un número variable de Objetos, desde 1 a 42. Se muestra una tabla del conjunto de conocimiento completo, es decir, de la información contenida en el portal web de la Universidad de Sevilla, en la Tabla 5.1.

Tema	*Número de Apartados*	*Número de Objetos para cada Apartado*
Tema 1.- Información General	12	4, 1, 1, 2, 8, 2, 2, 2, 6, 1, 7, 8
Tema 2.- Centros y Departamentos	6	2, 1, 1, 1, 1, 2
Tema 3.- Acceso y Estudios	11	7, 7, 8, 1, 1, 3, 1, 5, 1, 5, 1
Tema 4.- Postgrado y Doctorado	3	4, 1, 1
Tema 5.- Investigación y Transferencia Tecnológica	4	4, 1, 1, 1
Tema 6.- Biblioteca	6	1, 1, 4, 2, 1, 1
Tema 7.- Sociedad y Empresa	7	1, 1, 4, 1, 1, 3, 2
Tema 8.- Extensión Universitaria, Cultura y Deporte	8	1, 3, 6, 1, 2, 2, 1, 3
Tema 9.- Relaciones Internacionales	4	1, 1, 1, 1
Tema 10.- Servicios a la Comunidad Universitaria	6	1, 2, 2, 42, 1, 7
Tema 11.- Gestión y Administración	6	5, 4, 4, 3, 1, 1
Tema 12.- Universidad Virtual	6	6, 9, 4, 6, 1, 3

Tabla 5.1. Estructura jerárquica del portal web de la US.

Se puede observar que la información es muy heterogénea, en el sentido de que hay temas con una gran cantidad de información y otros que contienen bastante menos. Además, el

número de subconjuntos es muy variable. Sin embargo, el hecho de manejar la información de esta manera nos proporciona la ventaja de agrupar la información en *clusters* con contenidos relacionados. Esto permite que el Agente Inteligente pueda ofrecer al usuario no solo el Objeto más cercano a su petición, sino también Objetos que estén relacionados y que le pueden resultar interesantes. La lógica borrosa proporciona la flexibilidad necesaria para manejar este tipo de información.

Como ejemplo de la estructura jerárquica, se analiza a continuación el lugar que le corresponde en esa estructura a uno de los subconjuntos de conocimiento, concretamente al Tema 12, denominado Universidad Virtual. El Tema 12 está dividido en 6 Apartados:

- Apartado 12.1: Secretaría Virtual.
- Apartado 12.2: Recursos Electrónicos.
- Apartado 12.3: Servicios de internet.
- Apartado 12.4: Enlaces de interés.
- Apartado 12.5: Enseñanza Virtual.
- Apartado 12.6: Usuario Virtual.

A su vez, el Apartado 12.6, denominado Usuario Virtual, está dividido en 3 Objetos, a saber:

- Objeto 12.6.1: Solicitud de cuenta en Usuario Virtual.
- Objeto 12.6.2: Servicios de Usuario Virtual.
- Objeto 12.6.3: Problemas con Usuario Virtual.

5.2.2. Definición de preguntas tipo y extracción de palabras clave.

Una vez definida la estructura jerárquica del conjunto de conocimiento completo, que, en el caso de un entorno web, hacemos coincidir con la estructura jerárquica del portal web por comodidad, simplicidad y facilidad de uso, es necesario extraer las preguntas tipo asociadas a cada Objeto.

Como se ha explicado en apartados anteriores, a partir de un Objeto es posible extraer una o varias preguntas tipo. En el primer caso, la información es bastante concreta por lo que una sola pregunta basta para definir el Objeto. Cuando es necesario utilizar varias preguntas tipo, esto es debido a imprecisión o dispersión en la información o bien al posible uso de sinónimos por parte de los usuarios al realizar sus búsquedas.

Siguiendo con el ejemplo planteado en el subapartado 5.2.1, se muestran en la Tabla 5.2 las preguntas tipo definidas para el Apartado 6 del Tema 12:

Objeto	*Preguntas tipo*
12.6.1: Solicitud de cuenta en Usuario Virtual.	**1. Me gustaría saber cómo puedo solicitar una cuenta como usuario virtual de la Universidad de Sevilla**
12.6.2: Servicios de Usuario Virtual.	**1. ¿A qué servicios puedo acceder como Usuario Virtual de la Universidad de Sevilla?**
12.6.3: Problemas con Usuario Virtual.	**1. No recuerdo mi contraseña de Usuario Virtual de la Universidad de Sevilla.**
	2. No recuerdo mi clave de Usuario Virtual de la Universidad de Sevilla.
	3. No recuerdo mi password de Usuario Virtual de la Universidad de Sevilla.
	4. Tengo un problema para entrar en mi cuenta de Usuario Virtual de la Universidad de Sevilla.
	5. Tengo un problema con el password de mi cuenta de Usuario Virtual de la Universidad de Sevilla.
	6. Tengo un problema con la clave de mi cuenta de Usuario Virtual de la Universidad de Sevilla.
	7. Tengo un problema con la contraseña de mi cuenta de Usuario Virtual de la Universidad de Sevilla.

Tabla 5.2. Ejemplo de definición de preguntas tipo.

Entre los objetivos para el Agente Inteligente en un entorno web definidos en al apartado 4.2.1 de esta tesis, se mencionaba la posibilidad de crear distintas clases de preguntas tipo, entre las que se encuentran, resumiendo:

- La pregunta tipo principal.
- Preguntas tipo que consideran sinónimos.
- Preguntas tipo imprecisas.
- Preguntas tipo concretas dentro de objetos más generales.
- Preguntas tipo creadas por realimentación del sistema.

Objetos con información más concreta, como son los dos primeros (Objetos 12.6.1 y 12.6.2), únicamente necesitan una pregunta tipo (la pregunta tipo principal), mientras que algo más difuso como es un problema (Objeto 12.6.3), necesita de una mayor casuística y, por tanto, de más preguntas tipo.

Por ejemplo, el Objeto 12.6.2 está definido por la pregunta tipo:

¿A qué servicios puedo acceder como Usuario Virtual de la Universidad de Sevilla?

En este caso, los conceptos "servicios" y "Usuario Virtual" quedan muy claros y no es necesaria la definición de más preguntas tipo para este Objeto. Distinto es el caso del Objeto 12.6.3, que queda definido en un primer momento por la pregunta tipo principal:

Tengo un problema para entrar en mi cuenta de Usuario Virtual de la Universidad de Sevilla

Para este Objeto, sin embargo, hay que tener en cuenta dos cosas: que podemos considerar sinónimos de contraseña (por ejemplo, clave) y que en esa página web en concreto también se dan soluciones en el caso de no poder entrar en la cuenta de Usuario Virtual.

Por tanto se añaden preguntas tipo para consultas más concretas:

Tengo un problema con la contraseña de mi cuenta de Usuario Virtual de la Universidad de Sevilla.

Y, además, se añaden preguntas tipo sinónimas, como por ejemplo:

Tengo un problema con la clave de mi cuenta de Usuario Virtual de la Universidad de Sevilla.

No recuerdo mi clave de Usuario Virtual de la Universidad de Sevilla

No recuerdo mi contraseña de Usuario Virtual de la Universidad de Sevilla.

Lógicamente, el conocimiento del administrador del sistema respecto a la jerga del campo relacionado con los Objetos (y respecto al propio campo) es importante, dado que cuanto mayor sea su conocimiento, mayor será la fiabilidad de las preguntas tipo propuestas para la representación de ese Objeto (porque serán más parecidas a las consultas que harán los usuarios para recuperar la información). En nuestro caso, la definición de las preguntas tipo se ha basado tanto en el estudio de las propias páginas web como en consultas realizadas anteriormente por los usuarios del portal (banco de preguntas-respuestas de la Universidad de Sevilla).

Una vez que se han definido las preguntas tipo, de estas se extraen los denominados términos índice (*index terms*). Estos términos índice se han definido en nuestro caso como

palabras clave (*keywords*), aunque también podrían haberse definido como términos compuestos (*joint terms*). Los términos índice son las palabras que mejor definen un Objeto. Por ejemplo, en el caso de una de las preguntas anteriores, "¿A qué servicios puedo acceder como Usuario Virtual de la Universidad de Sevilla?", las palabras que en principio podrían ser seleccionadas como términos índice serían "servicios" "usuario", "virtual", "Universidad" y "Sevilla". Llegados a este punto, no hay que olvidar el concepto de *stop word*, definido en el capítulo 2 de esta tesis y, según el cual, las palabras que aparezcan muchas veces no aportan ninguna información y, por lo tanto, no deben ser consideradas como términos índice. Todos los artículos, determinantes, conjunciones y, en general, cualquier palabra susceptible de no aportar información o añadir confusión a la consulta será considerada una *stop word*. De hecho, y proporcionando un adelanto al apartado de realización de pruebas, la consideración de las palabras "Universidad" y "Sevilla" como *stop words* constituyó una mejor sustancial en los resultados.

Así mismo, cabría tener en cuenta la utilización de términos compuestos. Por ejemplo, "Usuario virtual" podría constituir un término compuesto. Sin embargo, en esta tesis consideramos que es preferible utilizar únicamente términos simples, que también llamaremos palabras clave, debido a la mayor flexibilidad que permite la utilización de los términos definidos de esta forma (puede ser que el usuario esté interesado no solo en respuestas que contengan el término compuesto, sino también en respuestas que contengan los términos simples asociados correspondientes).

Por último, es necesario añadir un peso o coeficiente a estas palabras clave. El valor de este peso está comprendido entre 0 y 1, es mayor mientras mayor sea la relación de la palabra clave con el subconjunto del nivel jerárquico correspondiente y debe ser definido para todos los niveles jerárquicos. Una mayor descripción de la forma en que se definen y asignan estos pesos a las palabras clave se puede encontrar en el siguiente capítulo de la tesis.

Se muestra un resumen de la metodología seguida en el ejemplo de este apartado en la Tabla 5.3.

Etapa	*Ejemplo*
Etapa 1: Identificación de información y Objeto.	**Página web: www.us.es/univirtual/internet**
Etapa 2: Localización del Objeto en la estructura jerárquica.	**Tema 12: Universidad Virtual.** **Apartado 6: Usuario Virtual.** **Objeto 2: Servicios de Usuario Virtual.**
Etapa 3: Definición de pregunta(s) tipo.	**Pregunta tipo: ¿A qué servicios puedo acceder como Usuario Virtual de la Universidad de Sevilla?**
Etapa 4: Extracción de términos índice.	**Términos índice: "servicios", "usuario", "virtual".**
Etapa 5: Asignación de pesos.	**Ver Capítulo 6 de esta tesis.**

Tabla 5.3. Resumen de la metodología empleada.

Hasta aquí se ha visto de forma general como se debe de configurar el Agente Inteligente, aunque se entrará en más detalle más adelante. Sin embargo, en el siguiente apartado se describe el comportamiento del Agente Inteligente ante una hipotética consulta de usuario.

5.2.3. Respuesta ante una consulta de usuario.

El objetivo final del Agente Inteligente debe ser el de encontrar el Objeto u Objetos cuya información sea más parecida a la consulta (*query*) que haya realizado un usuario.

Para clarificar un poco más el proceso que sigue el Agente Inteligente para encontrar la información relacionada con la consulta de usuario, y que ya fue descrito de forma general en el apartado 4.2.4 de esta tesis, se plasma en el ejemplo definido en la Tabla 5.3 del apartado anterior de este capítulo.

Paso 1: Realización de la consulta de usuario en Lenguaje Natural (NL).

¿A qué servicios puedo acceder como Usuario Virtual de la Universidad de Sevilla?

Paso 2: Extracción de términos índice o palabras clave. Se extraen todos los términos índice que aparezcan con un coeficiente de peso distinto de cero en alguno de los subconjuntos del nivel más bajo del conjunto de conocimiento (en este caso, el nivel 1, al que hemos denominado Nivel de Tema). Recordemos que cuando un término índice tiene una máxima relación con el subconjunto al que pertenece, el coeficiente de peso correspondiente a dicho subconjunto tiene un valor 1, mientras que cuando no pertenece a un subconjunto se le adjudica

un peso igual a 0. Una descripción detallada del proceso de asignación de estos coeficientes de peso está especificada en el Capítulo 6 de la tesis.

En la Tabla 5.4 se observan los coeficientes de peso de los términos índice extraídos de la consulta de usuario realizada en el paso 1 para cada subconjunto del nivel jerárquico 1 o Nivel de Tema.

Término Índice	**WT1**	**WT2**	**WT3**	**WT4**	**WT5**	**WT6**
Servicios	0.14	0	0	0	0	0.16
Usuario	0	0	0	0	0	0
Virtual	0	0	0.16	0	0	0
Término Índice	**WT7**	**WT8**	**WT9**	**WT10**	**WT11**	**WT12**
Servicios	0.16	0	0	0.14	0.16	0.15
Usuario	0	0	0	0.29	0	0.6
Virtual	0	0	0	0	0.16	0.53

Tabla 5.4. Coeficientes de peso para el Nivel de Tema para distintos términos índice.

WTi = Coeficiente de peso del Tema i. Los términos índice que no aparecen en el Tema correspondiente tienen un coeficiente con un valor igual a 0.

Los términos índice “Universidad” y “Sevilla” fueron considerados en principio como términos índice, observándose sin embargo que los resultados eran mejores si eran considerados como *stop words*.

Paso 3: Paso del vector de coeficientes por los motores lógicos correspondientes a cada Tema. Aunque se ahonda en los parámetros usados por el motor de inferencia borroso en el apartado siguiente de la tesis (5.3), cabe decir que se usan vectores de tres entradas al motor en este caso. Por ejemplo, y de acuerdo con la Tabla 5.4, el vector de entrada 1 es [0.14 0 0] mientras que el vector de entrada 12 es [0.15 0.6 0.53].

Con todo esto, los resultados obtenidos a la salida del motor lógico correspondiente a cada Tema, descrito así mismo en el apartado 5.3 de la tesis, vienen dados en la Tabla 5.5.

SMT1	**SMT2**	**SMT3**	**SMT4**	**SMT5**	**SMT6**
0.29	0.13	0.30	0.13	0.13	0.30
SMT7	**SMT8**	**SMT9**	**SMT10**	**SMT11**	**SMT12**
0.30	0.13	0.13	**0.43**	0.39	**0.62**

Tabla 5.5. Salidas del motor de inferencia borroso para el Nivel de Tema para los distintos subconjuntos (Temas).

SMTi = Salida del motor borroso correspondiente al Tema i.

Dado que es necesario establecer un determinado umbral, todas aquellas salidas que no lo superen provocan que se descarten todos los Apartados (y por consiguiente, también los Objetos de esos Apartados) de los Temas correspondientes. En nuestro caso, si fijamos un umbral de

0.4, este solo sería superado por las salidas correspondientes a los Temas 10 y 12 (sombreados en verde). Todos los Objetos pertenecientes a otros Temas quedan, por tanto, automáticamente eliminados, con el ahorro computacional correspondiente y el descarte de posibles "Objetos ruidosos" correspondientes a algunos de estos Temas.

Paso 4: Repetición del paso 3 para el nivel 2 de la estructura jerárquica, es decir, los Apartados de los Temas seleccionados. En la Tabla 5.6, se pueden observar los diversos coeficientes de peso para los Apartados del Tema 10, mientras que la Tabla 5.7 muestra lo propio para los Apartados del Tema 12.

Término Índice	WT10A1	WT10A2	WT10A3	WT10A4	WT10A5	WT10A6
Servicios	0	0	0	0.16	0	0.33
Usuario	0	0	0	0.4	0	0
Virtual	0	0	0	0	0	0

Tabla 5.6. Coeficientes de peso para el Nivel de Apartado para los distintos Apartados del Tema 10.

Término Índice	WT12A1	WT12A2	WT12A3	WT12A4	WT12A5	WT12A6
Servicios	0.37	0	0.16	0	0	0.12
Usuario	0	0	0	0	0	0.6
Virtual	0.33	0	0	0.16	0.16	0.45

Tabla 5.7. Coeficientes de peso para el Nivel de Apartado para los distintos Apartados del Tema 12.

WTiAj = Coeficiente correspondiente al Tema i, Apartado j.

Las salidas de los motores borrosos correspondientes a los vectores de entrada obtenidos de acuerdo con las Tablas 5.6 y 5.7 se pueden ver en las Tablas 5.8 y 5.9, correspondientes a las salidas del motor para los Apartados de los Temas 10 y 12, respectivamente.

SMT10A1	SMT10A2	SMT10A3	SMT10A4	SMT10A5	SMT10A6
0.13	0.13	0.13	**0.46**	0.13	0.38

Tabla 5.8. Salidas del motor de inferencia borroso para el Nivel de Apartado para los distintos Apartados del Tema 10.

SMT12A1	SMT12A2	SMT12A3	SMT12A4	SMT12A5	SMT12A6
0.51	0.13	0.30	0.30	0.30	**0.59**

Tabla 5.9. Salidas del motor de inferencia borroso para el Nivel de Apartado para los distintos Apartados del Tema 12.

SMTiAj = Salida del motor del Tema i, Apartado j.

Estableciendo nuevamente un valor umbral de salida del motor borroso de 0.4, se puede ver que se seleccionan nada más que el Apartado 4 del Tema 10, el Apartado 1 del Tema 12 y el Apartado 6 del Tema 12 (sombreados en verde).

Paso 5: Repetición del paso 3 para los Objetos de los Apartados seleccionados.

En este caso, se repite la misma operación para cada Objeto de los Apartados seleccionados. Dado que el Apartado 4 del Tema 10 contiene 42 Objetos (es el Apartado con más Objetos de todo el conjunto de conocimiento), nos ceñiremos en este ejemplo al Apartado 6 del Tema 12 y nos limitaremos a dar los resultados del caso anterior y del Apartado 1 del Tema 12 al final.

En la Tabla 5.10 se pueden ver los coeficientes para los Objetos del Apartado 6 del Tema 12, mientras que en la Tabla 5.11 se encuentran las salidas definitivas del Agente Inteligente para estos objetos.

Término Índice	WT12A6O1	WT12A6O2	WT12A6O3
Servicios	0	0.4	0
Usuario	0.57	0.52	0.52
Virtual	0.57	0.52	0.52

Tabla 5.10. Coeficientes de peso para el Nivel de Objeto para los distintos Objetos del Apartado 6 del Tema 12.

WTiAjPk = Coeficiente de peso correspondiente al Tema i, Apartado j, Objeto k.

SMT12A6P1	SMT12A6P2	SMT12A6P3
0.6045	0.7413	0.6005

Tabla 5.11. Salidas del motor de inferencia borroso para el Nivel de Objeto para los distintos Objetos del Apartado 6 del Tema 12.

SMTiAj = Salida del motor del Tema i, Apartado j, Pregunta k.

Se puede apreciar que, en este caso, los tres objetos están por encima del umbral de 0.4, por lo que entran dentro de la categoría de posibles respuestas a la consulta. Además de estos tres, superaron este umbral otros 4 Objetos, correspondientes a otros Apartados, que quedaron ordenados de la forma mostrada en la Tabla 5.12.

Posición	Tema	Apartado	Pregunta	Notación abreviada	Certeza (%)
1	12	6	2	12.6.2	74.13
2	12	6	1	12.6.1	60.45
3	12	6	3	12.6.3	60.05
4	12	1	5	12.1.5	54.07
5	12	1	6	12.1.6	54.07
6	10	4	9	10.4.9	48.96
7	12	1	1	12.1.1	41.04

Tabla 5.12. Objetos del conjunto de conocimiento devueltos por el Agente Inteligente.

En este caso, se ha logrado devolver la información demandada, puesto que la consulta realizada (**¿A qué servicios puedo acceder como Usuario Virtual de la Universidad de Sevilla?**) corresponde efectivamente a una pregunta tipo que hace referencia al Objeto 12.6.2 (notación abreviada para el Objeto correspondiente al Tema 12, Apartado 6, Objeto 2). Además, superan el umbral otros Objetos, a los que se han asignado, entre otras, las preguntas tipo mostradas en la Tabla 5.13.

Posición	Objeto	Certeza (%)	Pregunta tipo asociada
1	12.6.2	74.13	**¿A qué servicios puedo acceder como Usuario Virtual de la Universidad de Sevilla?**
2	12.6.1	60.45	**Me gustaría saber cómo puedo solicitar una cuenta como usuario virtual de la Universidad de Sevilla**
3	12.6.3	60.05	**No recuerdo mi contraseña de Usuario Virtual de la Universidad de Sevilla**
4	12.1.5	54.07	**Deseo tener acceso a los servicios económicos de la Secretaria Virtual de la Universidad de Sevilla**
5	12.1.6	54.07	**Deseo tener acceso a los servicios administrativos de la Secretaria Virtual de la Universidad de Sevilla**
6	10.4.9	48.96	**¿Qué servicios ofrece el Servicio de Informática y Comunicaciones?**
7	12.1.1	41.04	**¿Cómo puedo acceder a la Secretaria Virtual de la Universidad de Sevilla?**

Tabla 5.13. Preguntas tipo asociadas a los Objetos devueltos por el Agente Inteligente.

Como se ha expuesto antes, la primera pregunta tipo corresponde al Objeto buscado, pero, además, se obtiene una ventaja importante: las dos siguientes preguntas tipo devueltas están muy relacionadas con dicho Objeto (mencionan también al Usuario Virtual) e, igualmente, pudieran interesar a la persona que realiza la consulta. Las siguientes preguntas tipo, aun no siendo tan parecidas, sí podrían tener alguna relación con la búsqueda del consultante. Nuestra sugerencia es dirigir a dicho consultante a una página web asociada con la pregunta que aparece en primer lugar y presentar en otra ventana entre tres y cinco de las opciones devueltas.

Además, el hecho de presentar otras respuestas muy relacionadas con el Objeto de la consulta da pie a pensar que cuando la consulta no corresponda exactamente con ninguno de los Objetos buscados, el sistema tratará de encontrar los Objetos más parecidos, siendo esta flexibilidad una de las ventajas más importantes que aporta la lógica borrosa.

5.3 Sistema de Lógica Borrosa.

Como se comentó en el capítulo 4 de esta tesis, el núcleo inteligente del Agente está constituido por el sistema de lógica borrosa, para el cual tenemos que tener en cuenta parámetros como el número de entradas y salidas, los conjuntos borrosos que se definen para estas y, por supuesto, las reglas borrosas especificadas. La sección 5.3.1 describe el número óptimo de entradas al motor de inferencia borroso y los umbrales de certeza fijados para la eliminación de los distintos subconjuntos de conocimiento de cada nivel jerárquico. En el apartado 5.3.2 se define el número y la forma de los conjuntos borrosos que conforman el motor de inferencia, tanto de entrada como de salida. Por último, el punto 5.3.3 trata sobre la definición de las reglas borrosas que rigen el comportamiento del sistema. Estas cuestiones ya fueron introducidas y discutidas en [GÓMEZ06], [ROPERO07a] y [ROPERO07b].

Para poder probar la eficacia del sistema propuesto y mejorar las prestaciones, era necesario realizar pruebas para definir los parámetros idóneos del sistema de lógica borrosa sobre un conjunto de conocimiento acumulado. Dado que cuando se realizaron estas pruebas aún no estaba en funcionamiento el nuevo portal web de la Universidad de Sevilla, se recurrió al banco de preguntas-respuestas más frecuentes de la Universidad de Sevilla realizadas al administrador del portal. Este banco de preguntas-respuestas se constituyó en un principio como el conjunto de conocimiento, consistente en un total de 117 preguntas tipo agrupadas en 5 Temas, cada uno de los cuales contiene un número variable de Apartados y Objetos. Los resultados obtenidos a partir de él, debido a la generalidad del método, son aplicables a cualquier conjunto de conocimiento y, en particular, a cualquier portal web. En todo caso, se realizaron más pruebas a la hora de comparar los métodos de asignación de pesos a las palabras clave (las cuales se describen en el capítulo 6 de esta tesis) con el nuevo portal web de la Universidad de Sevilla ya en funcionamiento, y se comprobó la mencionada universalidad del método.

La primera meta de estas pruebas es comprobar que el sistema realiza una identificación correcta de las preguntas tipo proporcionadas al sistema con un índice de certeza mayor que un cierto umbral. El hecho de utilizar la lógica borrosa posibilita la identificación, no solo de la pregunta tipo correspondiente, sino de otras relacionadas. Este concepto está relacionado con el de memoria (*recall*), mencionado en el capítulo 2 de esta tesis, aunque no corresponde a su definición exacta.

El segundo objetivo es comprobar si la respuesta a la pregunta tipo requerida está entre las tres respuestas con mayor grado de certeza. Estas tres respuestas son las que podrían ser presentadas al usuario, siendo deseable que la respuesta correcta esté entre estas tres posibilidades. Este aspecto está relacionado con el concepto de precisión (*precision*), aunque de nuevo no corresponde a su definición exacta.

Para realizar estas pruebas, se construyó una herramienta con la posibilidad de realizar consultas de usuario y utilizar el motor borroso para obtener las respuestas deseadas. Para ello se introducirán las denominadas preguntas tipo como consultas en lenguaje natural.

Para construir la herramienta se propuso utilizar Borland C++ Builder, debido a la facilidad de construcción de la interfaz visual, la robustez de la aplicación generada y la posibilidad de compilar el código para Sistemas Operativos Windows o UNIX, lo que permite integrar la aplicación en el Sistema Operativo deseado. El programa Un-Fuzzy, mencionado en el capítulo 4 de esta tesis y descrito más en profundidad en el Apéndice A, fue el elegido para la implementación del motor de inferencia borroso, por su capacidad de exportar código C. La Figura 5.2 muestra la herramienta que fue utilizada para las pruebas de los parámetros del sistema de lógica borrosa.

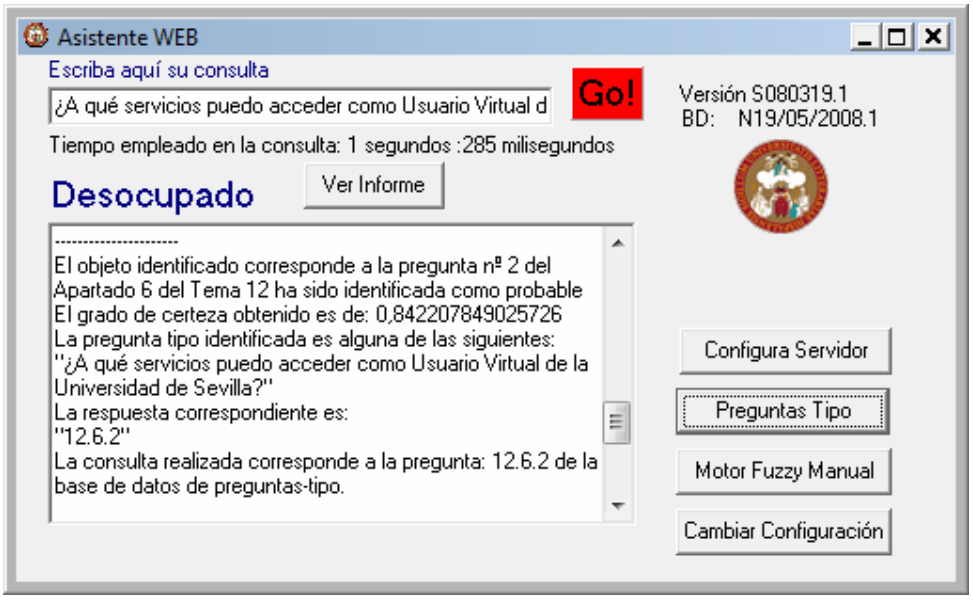


Figura 5.2. Herramienta utilizada para las pruebas de los parámetros de lógica borrosa.

Esta herramienta fue mejorada posteriormente y permite:

- La posibilidad de realizar cualquier consulta de usuario (*user query*).

- La introducción de las distintas preguntas tipo definidas por el sistema como consultas de usuario (*user queries*), con la finalidad de realizar pruebas. Estas preguntas se encuentran en una base de datos (Figura 5.3).

- El cambio de los parámetros de E/S (número de entradas y umbrales de certeza para cada nivel jerárquico) y las características del motor de inferencia borroso (difusor, concresor y tipo de conjuntos borrosos).

- La utilización de un motor borroso manual con el fin de realizar comprobaciones "sobre el terreno".

- La posibilidad de configurar la herramienta como servidor, con la consiguiente posibilidad de realizar consultas remotas.

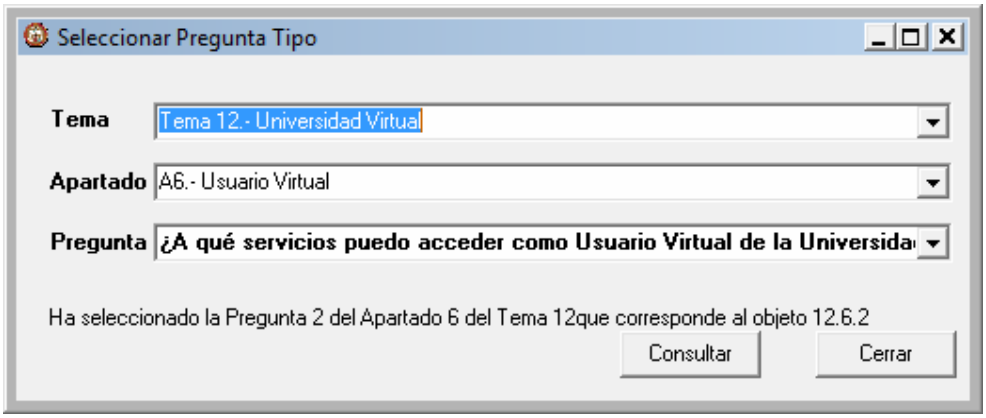


Figura 5.3. Selección de una pregunta tipo con la herramienta para pruebas.

Para realizar las pruebas, lo primero que se debe hacer es introducir cada una de las preguntas tipo en la herramienta. Dentro de cada pregunta tipo se determinan las palabras claves que servirán para identificar el Objeto relacionado con dicha pregunta. Los resultados de las pruebas de reconocimiento de preguntas tipo se han agrupado en cinco categorías, a saber:

1.- El Objeto que corresponde a la pregunta tipo realizada es el único devuelto o el que tiene mayor certeza.

2.- El Objeto que corresponde a la pregunta tipo realizada es devuelto con la segunda mayor certeza de todos los Objetos devueltos.

3.- El Objeto que corresponde a la pregunta tipo realizada es devuelto con la tercera mayor certeza de todos los Objetos devueltos.

4.- El Objeto que corresponde a la pregunta tipo realizada es devuelto, pero no entre los tres con mayor certeza de entre todos los Objetos devueltos.

5.- El Objeto que corresponde a la pregunta tipo realizada no aparece entre los

identificados.

Estas pruebas sirven para determinar los parámetros óptimos del sistema de lógica borrosa para IE y que se describen en las siguientes secciones.

5.3.1. Variables de E/S.

Como se ha visto en los apartados anteriores, el Agente Inteligente debe ocuparse de seleccionar las palabras clave introducidas por el usuario en su consulta en lenguaje natural. De entre las palabras clave seleccionadas, se eligen como entradas aquellas N palabras con mayor índice de significación usándose los pesos correspondientes a los diferentes subconjuntos de conocimiento como entradas a un motor de inferencia borrosa. Por tanto, lo primero que se debe hacer es determinar cuál es el número idóneo de entradas al sistema.

Además, dadas las entradas al motor de inferencia, los conjuntos borrosos y las reglas, los cuales serán definidos en apartados posteriores, el motor de inferencia debe encargarse de calcular la salida del sistema borroso para cada subconjunto de cada nivel jerárquico. El hecho de estructurar el contenido jerárquicamente posibilita que no haya que consultar todos los Objetos uno por uno, dado que se van descartando subconjuntos de conocimiento. En primer lugar, con los coeficientes de los términos índice que aparecen en la consulta pertenecientes al Nivel de Tema, se descartan aquellos temas cuya certeza (salida del motor borroso) no sea mayor que un determinado umbral. De esta forma, solo habrá que buscar en los Apartados de los Temas seleccionados, eliminando una gran cantidad de contenido, ahorrando carga computacional y evitando posibles fuentes de ruido derivadas de otros Objetos cuyos coeficientes de niveles inferiores para los términos índice seleccionados tengan un valor importante. El mismo proceso se repite para el nivel de Apartado, con lo que la búsqueda se limita a los Objetos de los Apartados seleccionados de entre los Temas filtrados anteriormente.

En las pruebas se han considerado umbrales de 0.5 para todos los niveles, aunque estos umbrales pueden ser modificados con el fin de obtener mejores resultados. Por otra parte, se definieron los conjuntos borrosos para las entradas y salidas y las reglas correspondientes al sistema de lógica borrosa. Por coherencia con la estructura de esta tesis, la forma en la que tanto unos como otras se definieron está explicada en apartados posteriores.

En cuanto a las entradas al sistema de lógica borrosa, se pueden extraer de una a cinco palabras clave o términos índice de una consulta. Se considera que más de cinco términos índice no deben ser relevantes para la extracción de información. Se pueden extrapolar las definiciones de memoria y precisión dadas en el capítulo 2 de esta tesis a nuestro caso, teniendo en cuenta que la memoria está tradicionalmente relacionada con el número de documentos relevantes recuperados (en nuestro caso, objetos) con respecto al número de documentos relevantes en total, mientras que la precisión relaciona dicho número de documentos relevantes recuperados con el número total de documentos recuperados. Por tanto, para que la memoria sea alta, es importante que se recuperen la mayoría de los objetos, pero para que se obtenga una alta precisión es necesario que no se recuperen muchos más objetos que los que sean relevantes, es decir, evitar introducir una gran cantidad de objetos irrelevantes entre los objetos devueltos. En la Figura 5.4 se representa este hecho.

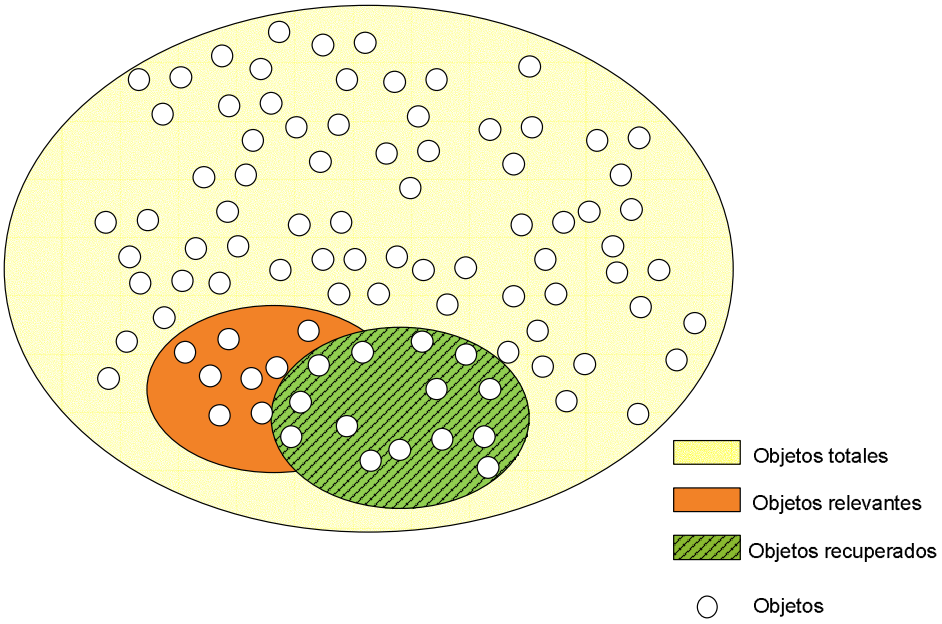


Figura 5.4. Memoria y precisión en el sistema.

Lo ideal es que las zonas naranja y verde rayada coincidan, es decir, que se recuperen todos los objetos relevantes (memoria alta) y que solo los objetos relevantes sean recuperados (alta precisión). Sin embargo, esta situación es difícil de conseguir y mientras más objetos se recuperan (zona verde rayada), también se recuperan más objetos irrelevantes, repercutiendo esto en la precisión, mientras que a medida que se recuperan menos objetos, también se recuperan menos objetos relevantes, redundando esto en una baja memoria. Es necesario, pues, encontrar una solución de compromiso.

Definir un motor con pocas entradas provoca la rápida saturación del sistema. Este es un gran inconveniente para la precisión: el 90% de las respuestas correctas son detectadas pero solo la mitad de ellas lo hacen como primera opción, tal y como se puede ver en la Figura 5.5, en la que se representan los resultados obtenidos para el caso en el que se consideran tres entradas al motor de inferencia borrosa.

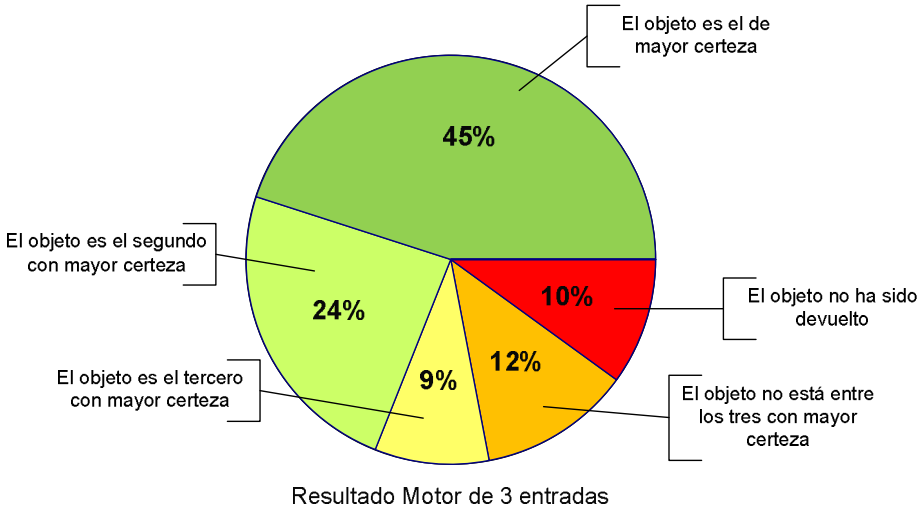


Figura 5.5. Resultados con tres entradas al motor de inferencia.

Por otra parte, si se definen cinco entradas al motor de inferencia, se producen valores muy bajos en el grado de certeza, como es lógico por otra parte, dado que un mayor número de palabras clave produce una mayor ambigüedad en la consulta. La precisión sube, detectándose en este caso el 55% de los objetos en primer lugar. Sin embargo, la memoria baja, no detectándose el 29 % de los objetos, como se puede observar en la Figura 5.6.

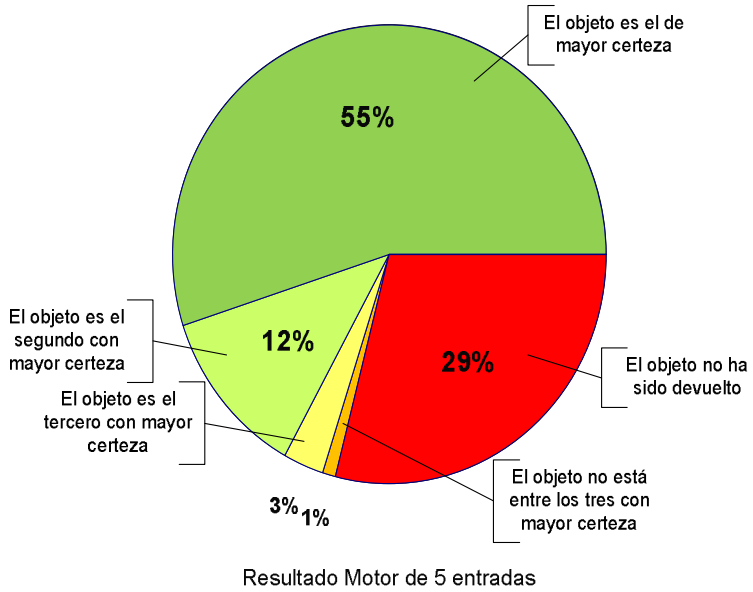


Figura 5.6. Resultados con cinco entradas al motor de inferencia.

Por lo tanto, concluimos que un número pequeño de entradas influye negativamente en la precisión mientras que un gran número de entradas lo hace en la memoria. No obstante, se pueden efectuar mejoras a este respecto si nos basamos en la utilización de un número de entradas variable, como se expondrá más adelante.

Además, del análisis de los resultados fallidos se observó que en buena parte de ellos no se obtenía el Objeto deseado porque la certeza fijada como umbral es mayor que la mínima obtenida. Esto plantea la posibilidad de bajar los umbrales de certeza para aceptar el resultado como correcto. No obstante, al aplicar esta modificación se dan por válidas muchas respuestas erróneas estropeando parte de los resultados anteriores.

La solución propuesta es modificar el procedimiento de manera que, sólo en el caso de que ningún resultado supere el umbral fijado, el Agente Inteligente baje automáticamente el umbral de aceptación del resultado. Aplicando este método, los resultados mejoran notablemente, como se puede observar en la Figura 5.7.

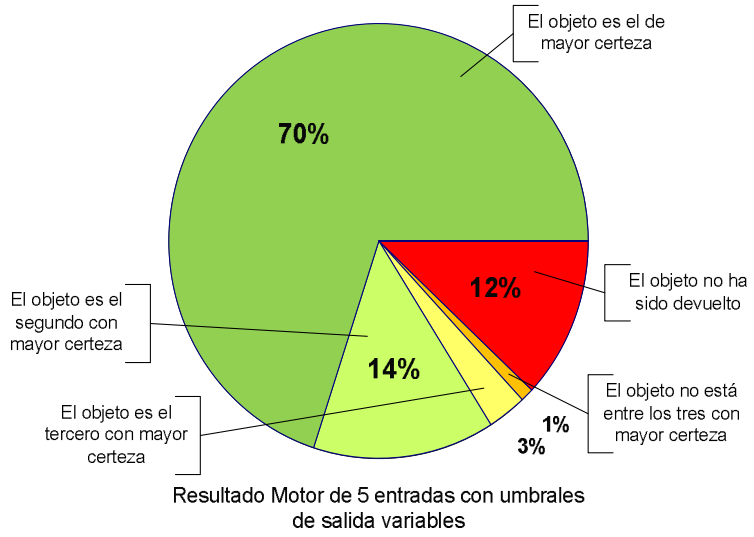


Figura 5.7. Resultados con cinco entradas al motor de inferencia utilizando umbrales de salida variables.

Viendo los buenos resultados obtenidos mediante la modificación anterior, cabe la tentación de subir mucho los umbrales de los índices de certeza para filtrar aún mejor los resultados. Observando los informes de resultados generados por la herramienta desarrollada para las pruebas, se apreciaba que no siempre el resultado correcto es el que tiene mayor certeza (hay que recordar que se trata de aplicar lógica difusa), por lo que si se suben mucho los umbrales es posible que no se devuelva el Objeto correcto. Además, es preciso recordar a estas alturas que al Agente Inteligente no se le formulará una consulta tipo sino una consulta parecida por lo que elevar mucho el umbral de exigencia podría inducir a repuestas erróneas.

En resumen, tomando como base las pruebas realizadas, se concluye que si se ofrecen al usuario los tres posibles resultados con mayor índice de certeza, el Objeto correcto es devuelto

el 88% de las veces, y es la primera opción el 70% de las veces, lo que, a todas luces, constituye un buen resultado.

En la herramienta para las pruebas se incluye pues la posibilidad de seleccionar el número de entradas y los umbrales de certeza para los distintos niveles jerárquicos tal y como se puede observar en la Figura 5.8.

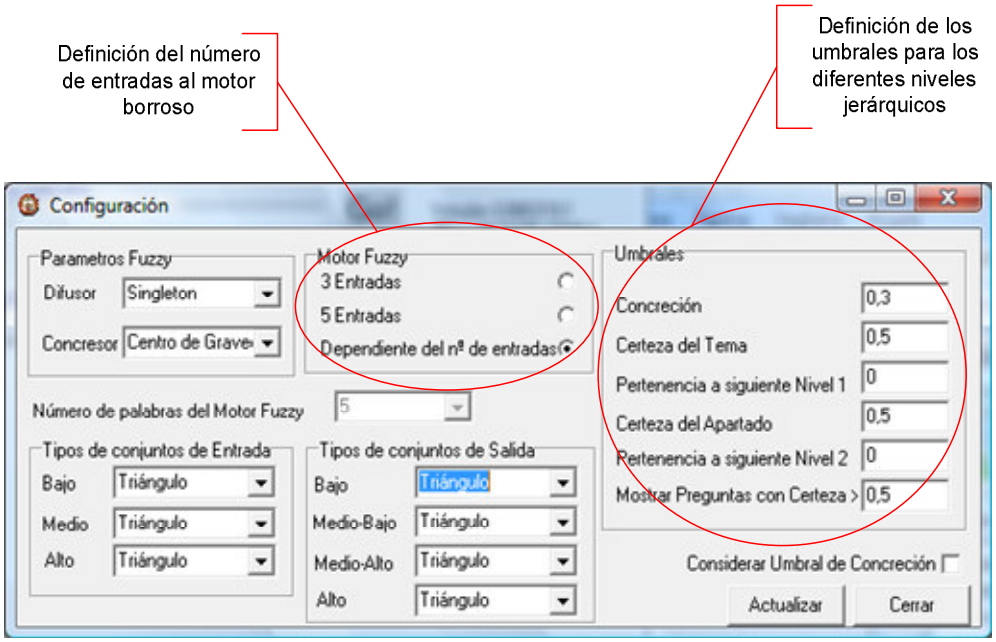


Figura 5.8. Definición del número de entradas al motor borroso y de los umbrales de certeza en la herramienta de pruebas.

En cuanto al número de entradas y, teniendo en cuenta que hay ocasiones en las que es mejor utilizar el motor difuso de tres entradas, mientras que en otras ocasiones es mejor utilizar el de cinco entradas, se propone un compromiso por el que se utiliza un motor variable dependiente del número de palabras clave identificadas en la consulta. En el caso de que se identifiquen entre una y tres palabras clave, se usa el motor de tres entradas, mientras que si se identifican más de tres palabras, es el motor de cinco entradas el utilizado. Los resultados obtenidos, y que tomamos como definitivos al respecto del número de entradas al motor de inferencia borroso y umbrales de salida del sistema, se pueden observar la Figura 5.9.

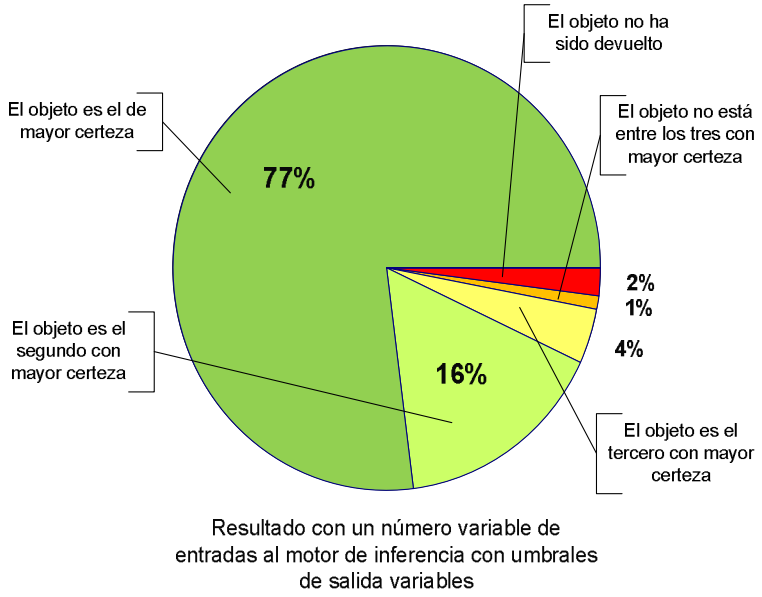


Figura 5.9. Resultados con un número variable de entradas al motor de inferencia utilizando umbrales de salida variables.

Se puede ver que ofreciendo al usuario los tres Objetos identificados como probables con mayor certeza, el usuario obtiene el Objeto correcto el 97 % de las veces y el 77 % de las veces es la primera opción, por lo que consideramos que utilizar un motor que utilice umbrales de salida variables y un número de entradas variable en función de los términos índice extraídos de la consulta es la mejor elección posible para los parámetros de E/S.

5.3.2. Definición de los conjuntos borrosos de E/S.

Tal y como se determinaba en la sección anterior, el universo de discurso de las entradas corresponde al rango de los pesos definidos para cada palabra clave, por lo que tiene un rango entre 0.0 y 1.0. En principio, se consideran tres conjuntos borrosos representados por los valores BAJO, MEDIO y ALTO. Así mismo, por simplicidad, todos son de tipo triangular, aunque posteriormente se modificarán los conjuntos con el fin de encontrar su forma óptima.

La salida, que indica el grado de certeza, también se define en un rango numérico entre 0 (mínima certeza) y 1 (certeza máxima) y tiene un universo lingüístico que puede tomar los valores BAJO, MEDIO-BAJO, MEDIO-ALTO, o ALTO. Estos valores corresponden a los conjuntos borrosos de salida.

El hecho de que las entradas tomen estos tres valores BAJO, MEDIO y ALTO es debido a que se trata del número suficiente como para que los resultados puedan ser coherentes y no son tantas opciones como para que el número de reglas aumente de manera alarmante (en la siguiente sección se comenta este aspecto, pero parece claro que, a mayor número de valores lingüísticos posibles en las entradas, es necesario definir un mayor número de reglas). De hecho, las salidas también se definieron en un principio de esta manera (tres posibles conjuntos

borrosos de salida). Tras un primer acercamiento, se consideró que introducir un cuarto conjunto borroso de salida, introducía una mejora considerable en los resultados.

Las entradas numéricas, que corresponden a los valores contenidos en los vectores de pesos, se encuentran por tanto acotadas entre 0 y 1. Si la entrada es 0, es evidente que el valor es BAJO con un grado de certeza 1, mientras que si la entrada es 1, el valor es ALTO con certeza 1. Los valores de entrada intermedios toman valores entre 0 y 1 para los conjuntos borrosos BAJO, MEDIO y ALTO. En la Figura 5.10 se observa cómo se definen estos conjuntos borrosos con el programa Un-Fuzzy.

El rango de valores de cada conjunto difuso de entrada es el siguiente:

- BAJO, de 0.0 a 0.4 con centro en 0.0.
- MEDIO de 0.2 a 0.8 con centro en 0.5.
- ALTO de 0.6 a 1.0 con centro en 1.0.

Todas las entradas que van al motor de inferencia borroso tienen una configuración idéntica.

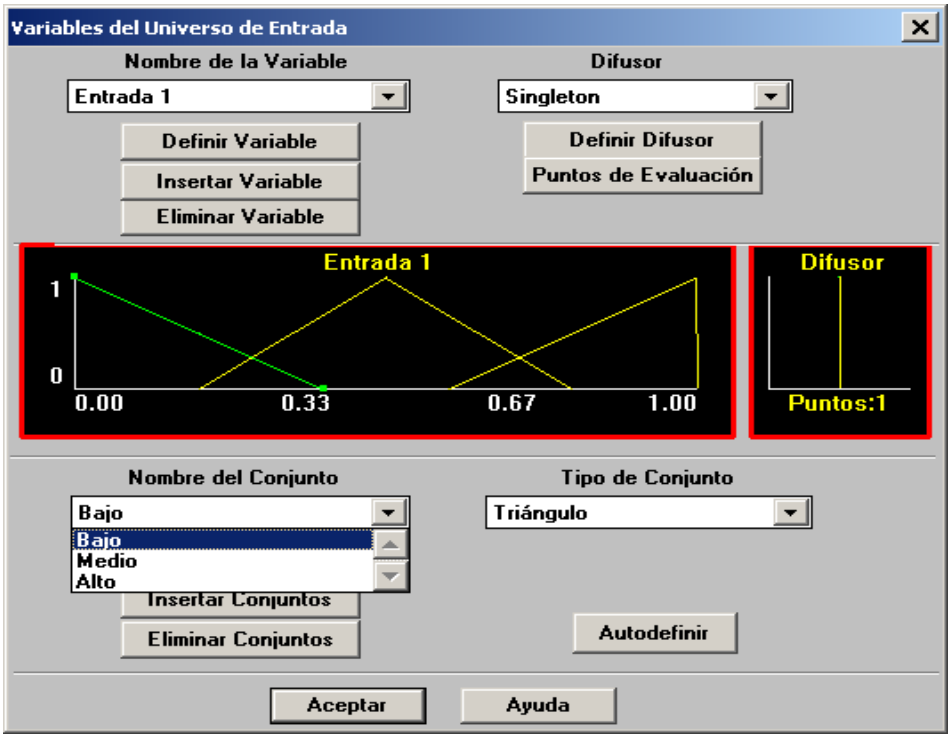


Figura 5.10. Definición de los conjuntos borrosos de entrada con el programa Un-fuzzy.

En cuanto a los conjuntos borrosos de salida, se definen de forma análoga, aunque con la diferencia de que, como se ha comentado antes, existe un conjunto borroso más. Se puede ver igualmente la definición de los conjuntos borrosos de salida con el programa Un-Fuzzy en la Figura 5.11.

El rango de valores de los conjuntos difusos de salida es el siguiente:

- BAJO de 0.0 a 0.4, con centro en 0.0.
- MEDIO-BAJO de 0.1 a 0.7, con centro en 0.4.
- MEDIO-ALTO de 0.3 a 0.9, con centro en 0.6.
- ALTO de 0.6 a 1.0, con centro en 1.0.

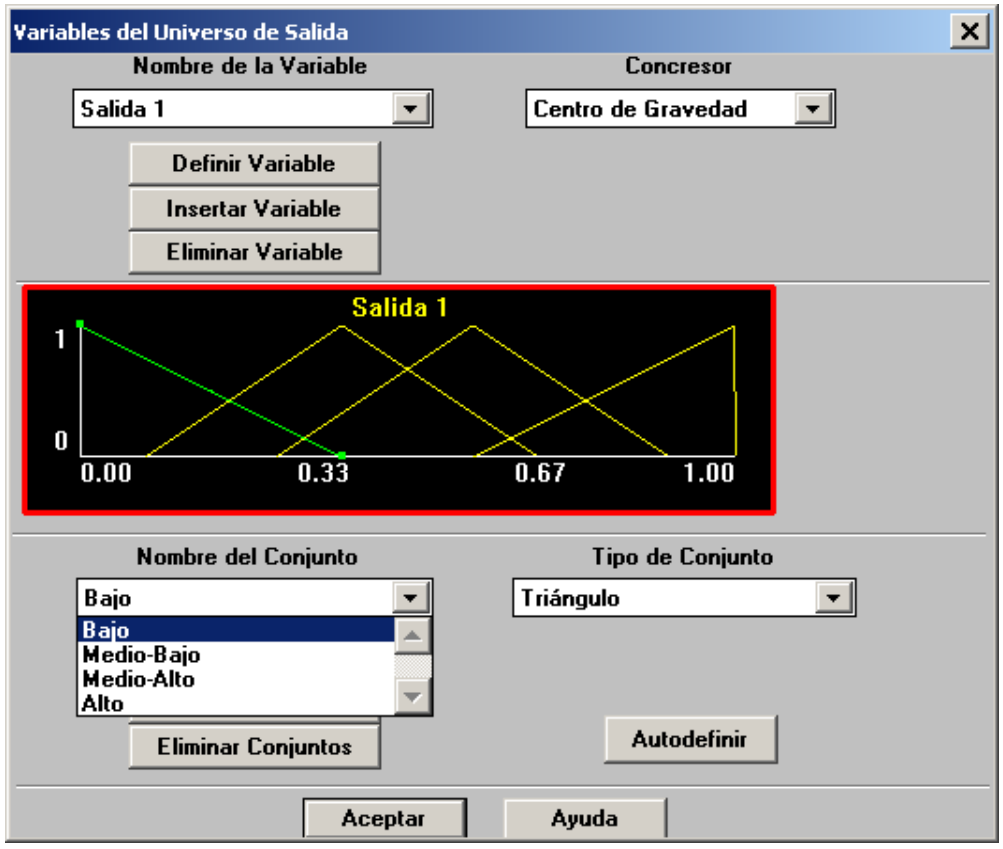


Figura 5.11. Definición de los conjuntos borrosos de salida con el programa Un-fuzzy.

Estos son los conjuntos borrosos con los que se realizaron las pruebas para determinar el número de entradas al motor de inferencia y los umbrales de salida descritas en la sección 5.3.1. En todo caso, es necesario comprobar los resultados que se obtendrían al utilizar otros conjuntos de entrada y salida.

Además, para la elección del difusor y el concresor de los conjuntos de entrada y salida respectivamente, se partió de la configuración más general usando un difusor singleton y un concresor centro de gravedad, pero se realizaron baterías de pruebas variando el difusor y concresor utilizados.

Se han considerado dos caracterizaciones del universo de discurso:

- 1ª caracterización.- Universo Recto, con todos los conjuntos borrosos del tipo triángulo.
- 2ª caracterización**.-** Universo Curvo, con los conjuntos borrosos BAJO y ALTO de tipo S y los conjuntos borrosos intermedios (MEDIO-BAJO y MEDIO-ALTO) de tipo Campana.

La herramienta para pruebas permite la utilización de distintos conjuntos borrosos para entrada y salida, así como el uso de distintos difusores y concresores, tal y como se muestra en la Figura 5.12.

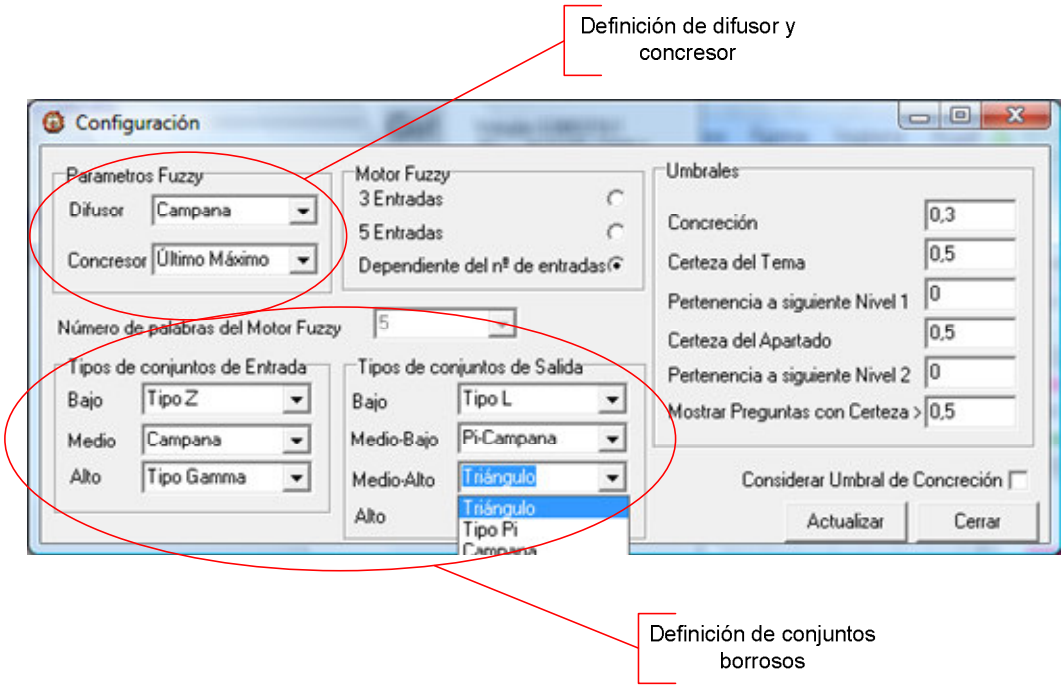


Figura 5.12. Definición del número de parámetros de borrosificación y desborrosificación y de los conjuntos borrosos de entrada y salida.

Para cada caracterización del universo de discurso se prueban tres configuraciones del motor de inferencia borroso. Estas tres configuraciones se describen en la Tabla 5.14 y se clasifican según los siguientes criterios:

- Según si el universo de discurso es recto o curvo (de acuerdo con la definición realizada más arriba).

- Según si los tipos de difusor y concresor utilizados.

Para una mayor información acerca de la forma de los conjuntos borrosos, concresores y difusores, se puede consultar el capítulo 3 de esta tesis.

Configuración	Universo	Difusor	Concresor
R1	Recto	Singleton	Centro de Gravedad
R2	Recto	Triangular	Centro de Gravedad
R3	Recto	Singleton	Media de Máximos
C1	Curvo	Singleton	Centro de Gravedad
C2	Curvo	Triangular	Centro de Gravedad
C3	Curvo	Singleton	Media de Máximos

Tabla 5.14. Configuraciones para el motor de inferencia borroso.

Los criterios seguidos para determinar qué configuración del motor difuso se ajusta mejor a las características del universo tratado se basan nuevamente en la introducción de las Preguntas Tipo que representan el conjunto de conocimiento como consultas de usuario. El resultado de la consulta se clasifica en las 5 categorías señaladas al comienzo del apartado 5.3 y que se denominaron Cat1, Cat2, Cat3, Cat4 y Cat5, en función de la posición en la que aparezca el Objeto entre los Objetos devueltos por el sistema en cuanto a su nivel de certeza:

- Categoría Cat1.- El Objeto que corresponde a la pregunta tipo es el único devuelto o el que tiene mayor certeza.

- Categoría Cat2.- El Objeto que corresponde a la pregunta tipo es devuelto con la segunda mayor certeza de todos los Objetos devueltos.

- Categoría Cat3.- El Objeto que corresponde a la pregunta tipo es devuelto con la tercera mayor certeza de todos los Objetos devueltos.

- Categoría Cat4.- El Objeto que corresponde a la pregunta tipo es devuelto, pero no entre los tres con mayor certeza de entre todos los Objetos devueltos.

- Categoría Cat5.- El Objeto que corresponde a la pregunta tipo no aparece entre los identificados.

Los resultados obtenidos se pueden observar en la Tabla 5.15.

Configuración	Cat1	Cat2	Cat3	Cat4	Cat5
R1	77.44%	15.79%	4.51%	0.75%	1.50%
R2	69.17%	18.05%	3.76%	5.26%	3.76%
R3	68.42%	15.04%	6.77%	7.52%	2.26%
C1	75.94%	15.79%	4.51%	1.50%	1.50%
C2	84.21%	8.27%	1.50%	2.26%	3.76%
C3	65.41%	18.80%	6.02%	8.27%	1.50%

Tabla 5.15. Resultados obtenidos para cada configuración según las distintas categorías.

La configuración que mejores resultados obtiene en cuanto a los Objetos devueltos en primer lugar es la configuración C2 (Universo Curvo con difusor triangular y concresor por Centro de Gravedad). El problema es que también tiene un porcentaje relativamente alto de Objetos no devueltos. Por tanto, también debemos seguir teniendo en cuenta las configuraciones C1 y R1 (Difusor singleton y concresor por Centro de Gravedad para Universos Curvo y Recto, respectivamente). Aunque no devuelven tantos Objetos en primera posición como la configuración anterior, sí devuelven la gran mayoría de los Objetos, lo que les otorga una mayor flexibilidad, clave para detectar Objetos en consultas vagas e imprecisas. En todo caso, los resultados obtenidos son bastante buenos, con el 97.25 % de los Objetos devueltos entre los tres primeros Objetos en el caso de la configuración R1.

Esta mayor flexibilidad del sistema unida a la mayor simplicidad de los conjuntos borrosos y el difusor, nos hace decantarnos por la configuración R1, y, por tanto, el sistema de FL elegido tiene finalmente las siguientes características:

- Conjuntos borrosos triangulares.
- Difusor singleton.
- Concresor por Centro de Gravedad.

La elección de esta configuración no quiere decir que otras configuraciones no puedan resultar igualmente válidas.

5.3.3. Definición de reglas.

Una vez que se han definido el número de entradas y los conjuntos borrosos, es necesario definir las reglas que rigen el funcionamiento del motor de inferencia borroso. En la sección 5.3.1 se había establecido la idoneidad de implementar un motor que aceptara un número de entradas variable según el número de palabras clave extraídas de la consulta de usuario. Esto, en la práctica, obliga a la implementación de dos motores de inferencia distintos:

- Si hay tres o menos palabras clave extraídas, se utiliza un motor de tres entradas.

- Si hay más de tres palabras clave extraídas, se utiliza un motor de cinco entradas (si hay más de cinco palabras clave extraídas, solo se consideran las cinco de mayor peso en cada subconjunto).

Por otra parte, en la sección 5.3.2, se habían definido tres conjuntos borrosos para cada entrada. Para el motor de tres entradas, esto supone la definición de $3^3 = 27$ reglas borrosas, mientras que para el motor de cinco entradas es necesario definir $3^5 = 243$ reglas. He ahí una de las razones por las que no se definieron más conjuntos borrosos: con un solo conjunto borroso más, para el motor de cinco entradas habría que definir $4^5 = 1024$ reglas.

Las reglas borrosas definidas se pueden observar en la Tabla 5.16 (motor de tres entradas) y en la Tabla 5.17 (motor de cinco entradas).

Número de regla	*Definición de regla*	*Salida*
R1	SI una o más entradas = ALTO	ALTO
R2	SI tres entradas = MEDIO	ALTO
R3	SI dos entradas = MEDIO y una entrada = BAJO	MEDIO-ALTO
R4	SI una entrada = MEDIO y dos entradas = BAJO	MEDIO-BAJO
R5	SI todas las entradas = BAJO	BAJO

Tabla 5.16. Reglas borrosas para el motor de tres entradas.

El desarrollo de las reglas borrosas para el motor de tres entradas mostrado en la Tabla 5.16 da lugar a 27 reglas distintas. Por otra parte, el criterio de definición de estas reglas se basa en los siguientes postulados:

- Si al menos una palabra clave tiene un coeficiente de peso alta es porque tiene una importancia capital para la detección de ese objeto, y por tanto, la salida es alta.

- Si todas las entradas tienen un coeficiente de peso medio, quiere decir que existen hasta tres palabras con un grado de significación lo suficientemente importante como

para ser tenidas en cuenta. Consideramos que tres palabras clave definen suficientemente un objeto como para asignar a la salida un valor alto.

- Por el mismo motivo, se considera que si una o dos entradas tienen un coeficiente de peso medio, las salida tiene que tener un valor medio, dado que, aunque puede no se defina al objeto claramente, sí es cierto que no hay que desdeñar estos objetos, que también pueden interesar al usuario. Como se ha mencionado antes, los valores intermedios se descompusieron en dos conjuntos borrosos, MEDIO-ALTO y MEDIO-BAJO, con lo que la salida toma el primero de estos valores cuando hay dos entradas con un coeficiente de peso medio, mientras que toma el segundo de ellos cuando solo hay una entrada a nivel medio.

- Lógicamente, si todas las entradas tienen un coeficiente de peso bajo, la salida también es baja.

Número de regla	*Definición de regla*	*Salida*
R1	SI dos o más entradas = ALTO	ALTO
R2	SI cuatro o más entradas = MEDIO	ALTO
R3	SI tres entradas = MEDIO	Al menos MEDIO-ALTO
R4	SI una entrada = ALTO y el resto de entrada ≠ BAJO	Al menos MEDIO-ALTO
R5	SI dos entradas = MEDIO y tres entradas = BAJO	MEDIO-BAJO
R6	SI una entrada= ALTO y cuatro entradas = BAJO	MEDIO-BAJO
R7	SI una entrada= MEDIO y cuatro entradas = BAJO	BAJO
R8	Si todas las entradas = BAJO	BAJO

Tabla 5.17. Reglas borrosas para el motor de cinco entradas.

El desarrollo de las reglas borrosas del motor de inferencia de cinco entradas mostrado en la Tabla 5.17 da lugar a 243 reglas distintas. El criterio de definición de las reglas borrosas es muy similar al definido para el motor de tres entradas, pero hay que tener en cuenta que al existir más palabras clave en la consulta del usuario, la consulta es más dispersa y, por tanto, los criterios deben de ser más restrictivos. Por ejemplo, una sola palabra clave no tiene por qué definir la consulta y, por tanto, el objeto de salida correspondiente.

El desarrollo completo de las reglas borrosas para ambos motores de inferencia se encuentra en el Apéndice B de esta tesis.

Capítulo 6. Nuevo método para la asignación de pesos basado en FL.

La asignación de pesos de los términos índice es uno de los mayores retos en IE e IR. El modelo más extendido para la recuperación de información, tal y como se mencionó en el capítulo 2, es el Modelo de Espacio Vectorial (VSM). En VSM, la importancia de un término en el subconjunto de conocimiento al que pertenece está representada por un determinado peso asociado [LEE97].

En un principio, se pensó en incluir la asignación de pesos en el capítulo 5 de esta tesis por formar parte del diseño del Agente Inteligente propuesto para la recuperación de información en entornos web, pero el tema tiene la entidad suficiente como para constituir un capítulo por sí mismo, dada la novedad que introducimos al presentar un esquema de asignación automática de pesos basado en la Lógica Borrosa.

En el apartado 6.1 se realiza una breve introducción a la asignación de pesos, la cual da paso a los apartados 6.2, en el que se presenta el método clásico de asignación de pesos, denominado TF-IDF, y al apartado 6.3, en el que lo enfrentamos al nuevo método propuesto basado en FL.

El apartado 6.4 trata sobre la implementación de ambos métodos. Finalmente, en el apartado 6.5 se realiza la comparación entre los resultados obtenidos con ambos métodos de asignación de pesos.

6.1. Introducción

En capítulos anteriores de esta tesis, se señaló la existencia de una serie de coeficientes de peso asociados a cada término índice. Los valores de los pesos deben estar relacionados de algún modo con la importancia de un término índice en el correspondiente conjunto de conocimiento (en nuestro caso Tema, Apartado y Objeto). Se comentaban, así mismo, las dos opciones posibles para definir estos pesos:

- La evaluación de los pesos por parte de un experto en la materia. Esta evaluación está basada en su percepción acerca de la importancia de los términos índice. Este método es simple, pero esto tiene la desventaja de depender exclusivamente del criterio del ingeniero de conocimiento, es muy subjetivo y no es susceptible de ser automatizado.

- La generación de pesos automatizados mediante un conjunto de reglas. El método más ampliamente usado para TW (*Term Weighting*, Asignación de Pesos) es el método TF-IDF, descrito en el capítulo 2 de esta tesis. En esta tesis, se propone un método novedoso para TW basado en FL y que obtiene mejores resultados para IE.

La gran cantidad de información contenida en un portal web hace inviable la primera opción por tediosa, intrincada, y por el alto nivel de maestría necesario por parte del Ingeniero de Conocimiento encargado de esta tarea para poder manejar el alto número de palabras clave generadas y, a su vez, asignar coeficientes de peso adecuados a dichas palabras clave en todos los niveles jerárquicos. Es necesario, pues, automatizar esta tarea.

En este capítulo se describe, en primer lugar, el funcionamiento del método de TW más extendido, el método TF-IDF. A continuación se propone un nuevo método de asignación de pesos basado en FL y por último, se implementan ambos métodos para el caso particular del portal web de la Universidad de Sevilla, obteniéndose resultados concretos del funcionamiento de ambos.

6.2. Descripción del método de asignación de pesos TF-IDF.

Aunque fue a finales de los años 50 cuando surgió la idea recuperar textos (concepto que posteriormente se ha extendido a la recuperación de información en general) mediante sistemas automáticos basados en la búsqueda de contenido textual mediante una serie de identificadores, fue Gerard Salton a finales de los años 70 y durante la década de los 80 quien sentó las bases para relacionar estos identificadores y los textos que representan [SALTON96].

Salton sugirió que cada documento podía ser representado por vectores de términos de la forma:

$D = (t_i, t_j, \ldots, t_p)$

Donde cada t_k identifica un término asignado a un documento D.

Una representación más formal del vector D nos llevaría no solo a considerar los términos que aparecen en dicho vector, sino que se añaden una serie de pesos w_{dk} representando el peso del término t_k en el documento D, es decir su importancia en dicho documento.

Un sistema para TW debe mejorar la eficacia en dos factores principales, la memoria y la precisión. La memoria tiene en cuenta el hecho de que los objetos más relevantes para el usuario deben ser recuperados. La precisión tiene en cuenta que los objetos no deseados por el usuario deben ser rechazados [RUIZ98]. Las definiciones exactas de memoria y precisión se encuentran en el apartado 2.3 de esta tesis.

En principio, es preferible un sistema que premie tanto una alta memoria, recuperando todo lo que sea relevante, como también una alta precisión, rechazando todos los objetos no deseados por el usuario. La memoria mejora si se usan términos índice de alta frecuencia, es decir, términos que ocurren en muchos documentos de la colección. Se puede esperar que tales términos devuelvan muchos documentos, incluyendo muchos de los documentos relevantes. El factor de precisión, sin embargo, mejora si se usan términos índice sumamente específicos que son capaces de aislar los pocos artículos relevantes de la masa de los no relevantes. En la práctica, se utilizan soluciones de compromiso, usando términos índice lo bastante frecuentes como para alcanzar un nivel de memoria razonable sin provocar una precisión demasiado baja.

Por lo tanto, y tal y como se comentó en el capítulo 2 de esta tesis, en primer lugar, los términos que se mencionan con frecuencia en documentos individuales, o en extractos de un documento, parecen ser útiles para mejorar la memoria. Esto sugiere la utilización de un factor denominado frecuencia de término (*Term Frequency*, TF) como parte del sistema de TW, midiéndose la frecuencia con la que se presentan los términos en el documento. Los pesos TF se han usado durante años en entornos de indexado automático.

En segundo lugar, el factor TF por sí solo no asegura una recuperación aceptable. En particular, cuando los términos de alta frecuencia no están concentrados en documentos concretos, pero en cambio son frecuentes en la colección entera, todos los documentos tienden a ser recuperados, y esto afecta a la precisión de búsqueda. De ahí que se deba introducir un nuevo factor dependiente de la colección que favorezca los términos que estén concentrados en pocos documentos de la colección. La frecuencia de documento inversa (*Inverse Document Frequency*, IDF), o frecuencia de colección inversa, es el factor que realiza esta función. El factor IDF es inversamente proporcional al número de documentos (*n*) a los cual un término es asignado en una colección de documentos *N*. Un factor típico IDF es *log (N/n)* [SALTON96].

Las consideraciones de discriminación de términos sugieren que los mejores términos para la identificación del contenido de un documento son aquellos capaces de distinguir ciertos documentos individuales del resto de la colección. Esto implica que los mejores términos deben tener altas frecuencias de término, pero frecuencias de colección totales bajas. Una medida razonable de la importancia de un término se puede obtener, por tanto, mediante el producto de la frecuencia de término y la frecuencia de documento inversa (TF x IDF), siendo usual describir el peso de un término i en un documento j de la forma expresada en la Ecuación 6.1.

$$w_{ij} = tf_{ij} \times idf_j$$

Ecuación 6.1. Fórmula para el cálculo de pesos.

Esta fórmula, inicialmente pensada para la búsqueda y extracción de documentos en una colección y que, eventualmente, ha sido utilizada también para buscar y extraer cualquier objeto de un conjunto de conocimiento acumulado, ha sido revisada y mejorada por otros autores, con el fin de obtener mejores resultados en IE e IR [LEE97; LIU01; ZHAO02; LERTNATTEE02; XU03].

6.3. Descripción del método de asignación de pesos con FL.

El método TF-IDF funciona razonablemente bien, pero tiene el inconveniente de no considerar dos aspectos que en esta tesis consideramos clave [ROPERO09]:

- El primer parámetro es el grado de identificación del objeto si sólo se usa en una consulta la palabra clave considerada. Este parámetro tiene una fuerte influencia sobre el valor final de un peso de término si el grado de identificación es alto. Mientras más identifica un objeto una palabra clave, un valor más alto tiene el peso correspondiente a dicha palabra clave. Por ejemplo, en la consulta de usuario 'Me gustaría obtener información acerca del SARUS', la palabra SARUS (el Servicio de Asistencia Religiosa de la Universidad de Sevilla) identifica la consulta en un alto grado.

  Sin embargo, este parámetro crea dos desventajas en términos de aspectos prácticos cuando hay que llevar a cabo una asignación automatizada y sistemática de pesos. Por una parte, el grado de identificación no es deducible de ninguna característica de una palabra clave, por lo que debe ser especificado por el Administrador de Sistema. Los valores asignados pudieran ser, pues, subjetivos, no unívocos y no sistemáticos. Por otra parte, la misma palabra clave puede tener una relación diferente con cada objeto.

- El segundo parámetro está relacionado con los términos índice compuestos, es decir, palabras clave ligadas unas a otras. En el ejemplo de la sección 5.2.2, correspondiente al Objeto 12.6.2, 'usuario virtual' constituiría un conjunto de dos palabras clave ligadas. Las palabras clave ligadas tienen pesos inferiores, puesto que el hecho de que estas palabras clave estén ligadas es lo que realmente determina el objeto con la certeza principal mientras que la aparición de una sola de estas las palabras puede referirse a otro objeto.

La consideración de estos dos parámetros junto a los parámetros clásicos TF e IDF determina el peso de un término índice para cada subconjunto en cada nivel. El método basado en FL aporta una solución para estos problemas: la solución es la de crear una tabla con todas las palabras clave y sus pesos correspondientes para cada objeto. Esta tabla será creada en la fase de extracción de palabras clave de las preguntas estándar. La imprecisión prácticamente no afecta el método de trabajo debido al hecho de que tanto la asignación de pesos como la extracción de la información están basadas en la lógica borrosa, lo que reduce al mínimo el efecto de las variaciones posibles de los pesos asignados.

Además, el método proporciona dos ventajas importantes:

- La asignación de pesos es automática.
- El nivel de experiencia requerida es muy inferior y no hay ninguna necesidad de que un operador conozca el funcionamiento de los motores de lógica borrosa, sino únicamente cuántas veces aparece una palabra clave aparece en subconjunto y la respuesta a dos preguntas simples:
    - ¿Cómo define una palabra clave un objeto por sí misma?
    - ¿Están ligadas varias palabras clave?

En nuestro caso, el propio desarrollador de la página web podría definir a la vez las preguntas asociadas al objeto (la página web), las palabras clave de cada objeto y la respuesta a las dos preguntas planteadas anteriormente, lo que simplifica notablemente el desarrollo del Agente Inteligente.

6.4. Implementación de ambos métodos de asignación de pesos.

En esta sección se muestra como el método TF-IDF y el método basado en FL han sido implementados en la práctica, con el fin de poder comparar ambos métodos, al aplicarlos al portal web de la Universidad de Sevilla.

6.4.1. Implementación del método TF-IDF.

Como se señaló en secciones anteriores, una medida razonable de la importancia de un término índice puede ser obtenida mediante el producto entre los factores TF y IDF (TF x IDF). Sin embargo, este fórmula ha sido modificada y mejorada por muchos autores para alcanzar mejores resultados en IR e IE. Finalmente, la fórmula escogida para las pruebas realizadas fue la propuesta por Liu et al. [LIU01] y que ya aparecía en el capítulo 2 de esta tesis. Dicha fórmula es la reseñada en la Ecuación 6.2.

$$W_{ik} = \frac{tf_{ik} \times \log(N / n_k + 0.01)}{\sqrt{\sum_{k=1}^{m} tf_{ik} \times \log(N / n_k + 0.01))^2}}$$

Ecuación 6.2. Fórmula TF-IDF modificada para el cálculo de pesos.

Donde tf_{ik} es la frecuencia de ocurrencia del término i en el subconjunto (Tema / Apartado / Objeto) k, y n_k es el número subconjuntos a los que el término T_k es asignado en una colección de N objetos, es decir, tiene en cuenta si un término está presente en otros conjuntos de la colección.

Veamos lo que ocurriría, por ejemplo, con el término "virtual" que, como vimos, era utilizado en el Objeto 12.6.2 correspondiente al ejemplo de la sección 5.2.2.

En el nivel de Tema:

- "Virtual" aparece 8 veces en el Tema 12 (tf_{ik} = 8, k = 12).
- "Virtual" aparece 2 veces en otros Temas (n_k = 3).
- Existen 12 Temas en total (N = 12).
- Para normalizar, solo hace falta conocer los otros tf_{ik} y n_k del Tema.
- Sustituyendo, el coeficiente resulta W_{ik} = 0.20. Puede parecer muy bajo, pero de hecho es el más alto del Tema. Esto es debido a la normalización, ya que existen 254 palabras. La consecuencia principal de este aspecto es que será necesaria una bajada de los umbrales en los niveles más altos de la estructura jerárquica, con el fin de no eliminar demasiado contenido.

En el nivel de Apartado:

- "Virtual" aparece 3 veces en el Apartado 12.6 (tf_{ik} = 3, k = 6).
- "Virtual" aparece 5 veces en otros Apartados del Tema 12 (n_k = 6).
- Existen 6 Apartados en el Tema 12 (= 6).
- Para normalizar, solo hace falta conocer los otros tf_{ik} y n_k del Apartado.
- Sustituyendo, el coeficiente resulta W_{ik} = 0.17.

En el nivel de Objeto:

- "Virtual" aparece 1 vez en el Apartado 12.6.2 (tf_{ik} = 1, k = 2). Lógicamente, en este nivel tf_{ik} siempre es igual a 1, debido a que una palabra solo puede aparecer una vez en cada Objeto.
- "Virtual" aparece 2 veces en otros Objetos del Apartado 12.6 (n_k = 3).

- Existen 3 Objetos en el Apartado 12.6 ($N = 3$)
- Para normalizar, solo hace falta conocer los otros tf_{ik} y n_k del Objeto.
- Sustituyendo, el coeficiente resulta $W_{ik} =$ 0.01. ("Virtual" aparece en las tres preguntas por lo que es irrelevante para distinguir este objeto de los otros tres).

Consecuentemente, el término "virtual" resulta relevante a la hora de distinguir que el Objeto pertenece al Tema 12 y al Apartado 6, pero es completamente irrelevante a la hora de definir el Objeto de la consulta. Este debe ser encontrado mediante la inclusión en la consulta de usuario de otros términos índice que estén relacionados con el Objeto.

6.4.2. Implementación del método basado en FL.

Como se explicó en la sección 6.3, el método TF-IDF tiene la desventaja de no considerar el grado de identificación del objeto si sólo se considera el término índice usado y la existencia de palabras clave ligadas. Como en el método TF-IDF, en el método de TW basado en FL, es necesario conocer los valores de los factores TF e IDF y, además, también se debe conocer la respuesta a las dos preguntas mencionadas en el punto anterior.

El método de definición de los pesos o coeficientes se define a continuación. Para definir el peso correspondiente a un término índice, se debe contestar a cuatro preguntas [ROPERO09]:

- *Pregunta 1*: ¿Con qué frecuencia aparece la palabra clave entre las de los otros subconjuntos de conocimiento? (Esta pregunta está relacionada con el factor IDF).
- *Pregunta 2*: ¿Con qué frecuencia aparece la palabra clave entre las del subconjunto de conocimiento al que pertenece? (Esta pregunta está relacionada con el factor TF).
- *Pregunta 3*: ¿En qué medida define una palabra clave a un objeto?
- *Pregunta 4*: ¿Está la palabra clave ligada a otras palabras clave?

Con la respuesta a estas preguntas, se definen una serie de valores que constituyen las entradas a un sistema de lógica borrosa, al que se denomina Asignador de Coeficientes. La salida del sistema de FL del Asignador de Coeficientes es la que finalmente define el coeficiente de peso asociado a un término índice para cada nivel jerárquico. El esquema seguido se puede observar en la Figura 6.1, siendo importante recordar que el sistema de FL (o motor borroso) utilizado no es el mismo que se usa para la extracción del conocimiento en el Agente Inteligente, descrita en capítulos anteriores, siendo las reglas borrosas distintas.

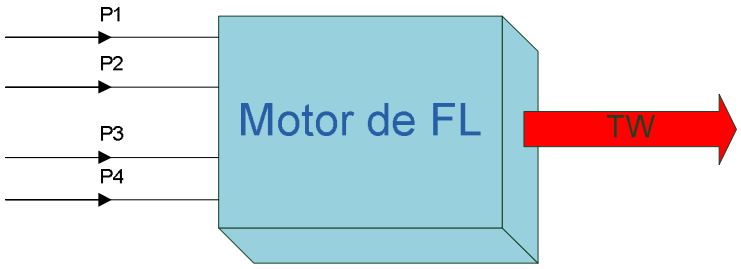


Figura 6.1. Esquema de la generación de coeficientes de peso para el método basado en FL.

En la Figura 6.1, P_i corresponde a la entrada asociada a la pregunta i-ésima de las definidas anteriormente, mientras que TW es la salida del motor de Lógica Borrosa, que no es más que el coeficiente de peso obtenido. A continuación se describe la forma de definir los valores de entrada al sistema asociados a cada una de las cuatro preguntas.

Pregunta 1 (P1):

Para definir los valores numéricos de entrada al sistema de FL, tenemos en cuenta las veces que aparecen los términos más utilizados en todo el conjunto de conocimiento. La lista de las palabras clave más usadas en el total del conjunto de conocimiento acumulado aparece en la Tabla 6.1.

Número de orden	Palabra clave	Número de apariciones en el conjunto de conocimiento acumulado
1	**Servicio**	**31**
2	**Servicios**	**18**
3	**Biblioteca**	**16**
4	**Investigación**	**15**
5	**Dirección**	**14**
	Universitaria	**14**
7	**Correo**	**13**
	Acceso	**13**
9	**Electrónico**	**12**
	Informática	**12**
	Recursos	**12**
12	**Centro**	**10**
	Educación	**10**
	Matrícula	**10**
	Programa	**10**
	Virtual	**10**

Tabla 6.1. Lista de palabras más usadas en el conjunto de conocimiento acumulado.

El valor de la entrada asociado a la pregunta 1 (P1. ¿Con qué frecuencia aparece la palabra clave entre las de los otros subconjuntos de conocimiento?) viene dado por un valor entre 0 (aparece muchas veces) y 1 (no aparece en ningún otro subconjunto). Lógicamente, mientras menos veces aparece un término en otros subconjuntos del mismo conjunto de conocimiento, mayor es la probabilidad de que la consulta esté relacionada con el subconjunto referido. Esta pregunta P1 se corresponde con el factor IDF.

Dado que para el caso del portal web de la Universidad de Sevilla se han definido un total de 1114 palabras clave, consideramos que el 1% de estas palabras debe marcar la frontera para el valor 0 (11 palabras). La palabra clave “recursos” ocupa ese undécimo lugar, apareciendo un

total de 12 veces. Por tanto, cada vez que aparezca una palabra más de 12 veces en otros subconjuntos (el valor que marca esta undécima palabra), le será asignado el valor 0.

En general, para cada Tema, el valor de la entrada al Asignador de Coeficientes asociada a la Pregunta 1 (P1) se corresponde con los valores dados en la Tabla 6.2.

Número de apariciones	0	1	2	3	4	5	6
Valor	1	0.9	0.8	0.7	0.64	0.59	0.53
Número de apariciones	7	8	9	10	11	12	≥13
Valor	0.47	0.41	0.36	0.3	0.2	0.1	0

Tabla 6.2. Valor de la entrada al Asignador de Coeficientes asociada a P1 para el nivel jerárquico de Tema.

Entre 0 y 3 (aproximadamente un tercio de los valores considerados), se considera un valor ALTO. Este valor es dominante, es decir, se trata del conjunto borroso más importante para los valores de entrada entre 0.7 y 1, de ahí los valores uniformemente repartidos. Análogamente ocurre con los valores entre 9 y ≥ 13 (valor BAJO dominante entre 0 y 0.3). El subconjunto más grande se ha tomado para el valor MEDIO (dominante entre 0.3 y 0.7). Como se puede observar, los valores están uniformemente repartidos igualmente. La Figura 6.2 muestra los tres conjuntos borrosos de entrada al sistema de FL, los cuales tienen forma triangular y, como se ha comentado, se definen por las variables lingüísticas BAJO, MEDIO y ALTO.

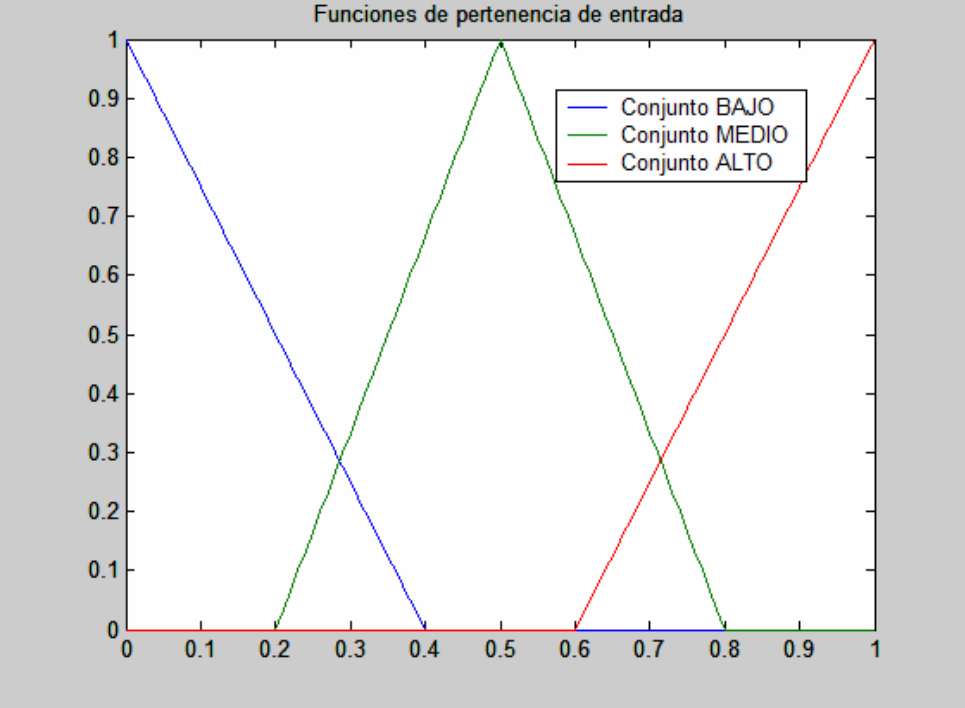


Figura 6.2: Conjuntos borrosos de entrada.

Por otra parte, y dado que en cada nivel jerárquico se define un coeficiente de peso distinto, es necesario considerar otros baremos para calcular el valor de entrada al motor difuso de asignación de coeficientes en los siguientes niveles jerárquicos. Como para el nivel de Tema se consideraba el nivel superior (conjunto de conocimiento completo), para el nivel de Apartado consideraremos las veces que aparece una palabra clave en un determinado Tema.

Hay que tener en cuenta que se consideran todos los Temas, tomándose como referencia el valor del Tema en el que la palabra clave aparece en más ocasiones. La lista de las palabras clave más usadas en un Tema y el Tema en el que estas aparecen se muestran en la Tabla 6.3.

Número de orden	Palabra clave	Número de apariciones en el conjunto de conocimiento acumulado	Tema en el que aparece en más ocasiones
1	**Servicio**	**16**	**Tema 10**
2	**Dirección**	**10**	**Tema 1**
	Biblioteca	**10**	**Tema 6**
	Matrícula	**10**	**Tema 4**
5	**Correo**	**9**	**Tema 1**
	Electrónico	**9**	**Tema 1**
7	**Virtual**	**8**	**Tema 12**
8	**Informática**	**7**	**Tema 10**
	Servicios	**7**	**Tema 1**
10	**Educación**	**5**	**Tema 1**
	Recursos	**5**	**Tema 12**

Tabla 6.3. Lista de palabras más usadas en un solo Tema.

De manera análoga al nivel jerárquico superior (el nivel de Tema), se considera que el coeficiente de entrada al Asignador de Coeficientes viene dado por un valor entre 0 (aparece muchas veces) y 1 (no aparece en ningún otro subconjunto). Consideramos de nuevo que el 1% de estas palabras debe marcar la frontera para el valor 0 (11 palabras) por lo que cada vez que aparezca una palabra más de 5 veces en otros subconjuntos, se le asigna el valor 0.

Para cada Apartado, el valor de la entrada al Asignador de Coeficientes asociada a la Pregunta 1 (P1) se corresponde con los valores dados en la Tabla 6.4.

Número de apariciones	0	1	2	3	4	5	≥ 6
Valor	1	0.7	0.6	0.5	0.4	0.3	0

Tabla 6.4. Valor de la entrada al Asignador de Coeficientes asociada a P1 para el nivel jerárquico de Apartado.

El método es análogo al anterior volviendo a tener en cuenta la definición de los conjuntos borrosos.

Para hallar el coeficiente de peso asociado al nivel de Objeto, el método es ligeramente diferente ya que, aún basándose en la definición de los conjuntos borrosos, no tiene en cuenta el número máximo de palabras para cada apartado, sino que directamente pasa a la frontera entre conjuntos borrosos cuando se aumenta el número de Objetos en los que aparece en una unidad.

Los valores de entrada al Asignador de Coeficientes asociados a P1 para el nivel jerárquico de Objeto quedan, pues, de la forma en la que se puede observar en la Tabla 6.5.

Número de apariciones	0	1	2	≥ 3
Valor	1	0.7	0.3	0

Tabla 6.5. Valor de la entrada al Asignador de Coeficientes asociada a P1 para el nivel jerárquico de Objeto.

Pregunta 2 (P2):

Para hallar el valor de entrada al Asignador de Coeficientes asociado a la pregunta 2 (P2. ¿Con qué frecuencia aparece la palabra clave entre las del subconjunto de conocimiento al que pertenece?), el razonamiento es análogo al utilizado para P1, pero teniendo en cuenta que ahora solo hay que considerar la frecuencia dentro de un único subconjunto de conocimiento y no la frecuencia de aparición en los demás. Por consiguiente, el número de apariciones se reduce considerablemente. Lógicamente, mientras más veces aparece un término en un subconjunto, mayor es la probabilidad de que la consulta esté relacionada con este. Esta pregunta P2 se corresponde con el factor TF.

Si tomamos nuevamente la lista con las palabras clave más usadas en un Tema (Tabla 6.3) y, teniendo en cuenta que mientras más veces aparezca la palabra clave en el Tema o Apartado, mayor debe ser el valor de la entrada, se obtienen las magnitudes mostradas en las Tablas 6.6 y 6.7. Estas tablas corresponden a los valores para los niveles jerárquicos de Tema y Apartado, respectivamente.

Número de apariciones	1	2	3	4	5	≥ 6
Valor	0	0.3	0.45	0.6	0.7	1

Tabla 6.6. Valor de la entrada al Asignador de Coeficientes asociada a P2 para el nivel jerárquico de Tema.

Número de apariciones	1	2	3	4	5	≥ 6
Valor	0	0.3	0.45	0.6	0.7	1

Tabla 6.7. Valor de la entrada al Asignador de Coeficientes asociada a P2 para el nivel jerárquico de Apartado.

P2 carece de sentido para determinar el valor la entrada al Asignador de Coeficientes para el último nivel jerárquico, puesto que a este nivel, una palabra clave aparece únicamente una vez en cada Objeto.

Pregunta 3 (P3):

En el caso de Pregunta 3 (P3. ¿En qué medida define una palabra clave a un Objeto?), la respuesta es completamente subjetiva. En esta tesis, proponemos las respuestas "Sí / Mucho", "Algo" y "Poco / Nada". En la Tabla 6.8, se muestran los valores de entrada al Asignador de Coeficientes asociados a P3. Este valor es independiente del nivel jerárquico al que nos refiramos.

Respuesta (¿Define la palabra al Objeto?)	Si / Mucho	Algo	Poco / Nada
Valor	1	0.5	0

Tabla 6.8. Valor de la entrada al Asignador de Coeficientes asociada a P3.

Por lo tanto, el desarrollador de una página web, solo tendría que contestar "Mucho", "Algo" o "Poco" a la pregunta de si la palabra clave define bien al objeto, sin complicadas fórmulas matemáticas que lo describan.

Pregunta 4 (P4):

Finalmente, para la Pregunta 4 (P4. ¿Está la palabra clave ligada a otras palabras clave?), se proponen los valores de entrada al Asignador de Coeficientes de la Tabla 6.9. Si la palabra clave está ligada a otras tiene un peso menor, puesto que para referirse al Objeto en cuestión debe aparecer la/s palabra/s clave ligadas a ella.

Palabras a las que está ligada la palabra clave	0	1	2	≥ 3
Valor	1	0.7	0.3	0

Tabla 6.9. Valor de la entrada al Asignador de Coeficientes asociada a P4.

Nuevamente, tomar los valores 0.7 y 0.3 es consecuencia de considerar la frontera entre los conjuntos borrosos dominantes (Figura 6.1).

Después de la consideración de todos estos factores, hay que definir las reglas borrosas. En el caso de los niveles jerárquicos de Tema y Apartado, hay que considerar los cuatro valores de entrada asociados a las preguntas planteadas anteriormente y que denominamos P1, P2, P3 y P4 en función de la pregunta con la que se asocian. Aparte de los tres conjuntos borrosos de entrada, dados en la Figura 6.2, se han definido cuatro conjuntos de borrosos de salida: ALTO, MEDIO ALTO, MEDIO BAJO Y BAJO. Estos conjuntos borrosos se muestran en la Figura 6.3.

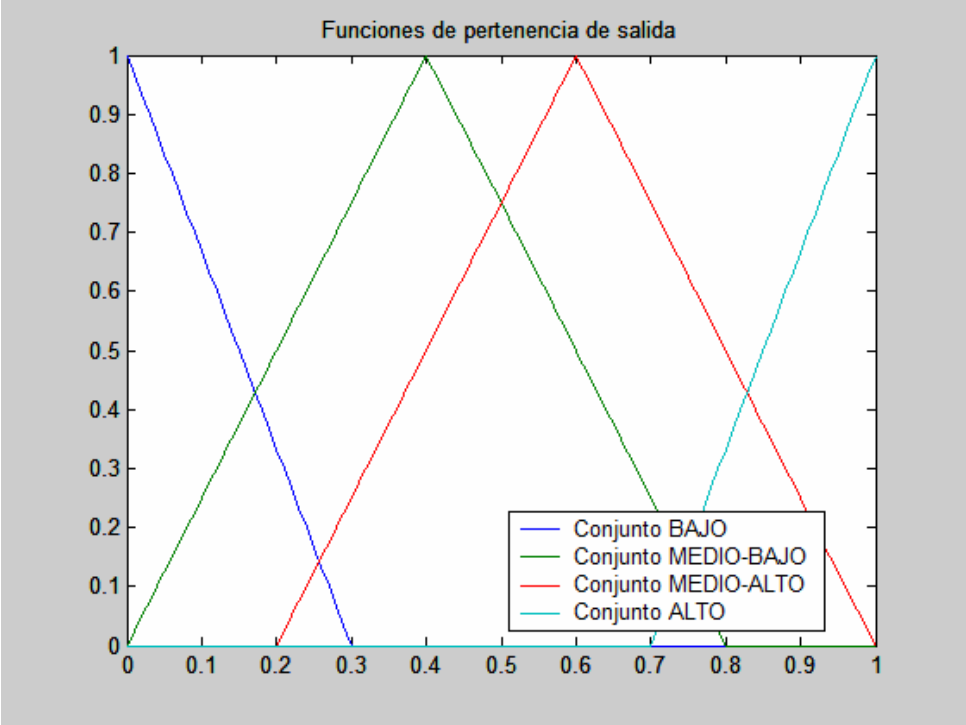


Figura 6.3: Conjuntos borrosos de salida.

Para la definición de las reglas borrosas del motor de inferencia para la generación de pesos, se han seguido básicamente los siguientes criterios:

- Un valor alto de P1 (relacionado con el factor IDF) implica que la palabra no está demasiado presente en otro conjuntos de conocimiento, por lo que la salida será alta, a no ser que la palabra tenga por sí sola muy poca importancia (P3 bajo) o que esté ligada a muchas palabras (P4 bajo).

- Un valor alto de P2 (relacionado con el factor TF), suele implicar un valor alto de la salida, puesto que la palabra clave está muy presente en un conjunto de conocimiento. Sin embargo, si P1 tiene un valor bajo, quiere decir que la palabra está presente en todo el conjunto de conocimiento, por lo que no es muy útil para la extracción de la información.

- P3 es un parámetro muy importante, puesto que si una palabra por sí sola define muy bien a un Objeto determinado, es mucho más fácil encontrar dicho Objeto.

- Un valor bajo de P4 hace que una palabra clave tenga una menor importancia, debido a que está asociada a otras palabras, por lo que provoca que el valor de la salida sea menor.

La combinación de las cuatro entradas con tres conjuntos borrosos de entrada proporciona 81 combinaciones posibles, las cuales están resumidas en la Tabla 6.10. El desglose completo de las reglas se muestra en el Apéndice B, correspondiente a las reglas borrosas.

Número de regla	*Definición de regla*	*Salida*
R1	SI P1 = ALTO y P2 ≠ BAJO	Al menos MEDIO-ALTO
R2	SI P1 = MEDIO y P2 = ALTO	Al menos MEDIO-ALTO
R3	SI P1 = ALTO , P2 = BAJO y P3 y P4 son al menos MEDIO	Al menos MEDIO-ALTO
R4	SI P1 = ALTO , P2 = BAJO y P3 es BAJO	MEDIO-BAJO
R5	SI P3 = ALTO	Al menos MEDIO-ALTO
R6	SI P4 = BAJO	Desciende un nivel
R7	SI P4 = MEDIO	SI la salida es MEDIO-BAJO, desciende a BAJO
R8	SI (R1 y R2) ó (R1 y R5) ó (R2 y R5)	ALTO
R9	En cualquier otro caso	MEDIO-BAJO

Tabla 6.10. Definición de las reglas borrosas para la asignación de coeficientes en los niveles jerárquicos de Tema y Apartado.

En el nivel de Objeto, debemos desechar la Pregunta 2 y, por tanto, se produce un cambio en las reglas, aunque los criterios son similares al caso anterior. Una entrada menos reduce el número de reglas a 27. Un resumen de estas reglas se puede observar en la Tabla 6.11. Para ver la definición completa de las reglas, nuevamente se remite al Apéndice B.

Número de regla	*Definición de regla*	*Salida*
R1	SI P3 y P4 = ALTO y P1 ≠ BAJO	ALTO
R2	SI dos entradas = ALTO y una entrada = MEDIO	Al menos MEDIO-ALTO
R3	SI dos entradas = MEDIO y una entrada ≠ BAJO	MEDIO-ALTO
R4	SI P1 = BAJO y las otras dos entradas ≠ BAJO	MEDIO-BAJO
R5	SI dos entradas son MEDIO y una entrada = BAJO	MEDIO-BAJO
R6	SI al menos dos entradas = BAJO	BAJO

Tabla 6.11. Definición de las reglas borrosas para la asignación de coeficientes en el nivel jerárquico de Objeto.

Lo único que queda por definir es qué ocurre si un término índice aparece varias veces en un Tema o Apartado. Por ejemplo, pudiera ser que en un caso la respuesta a la pregunta 3 fuera "Algo" y en otro "Nada". En ese caso, se realizará la media ponderada de los coeficientes correspondientes (en el caso sugerido, 0.25).

Por último, hay que tener en cuenta otro aspecto. El motor de inferencia de cinco entradas necesita de valores más altos para saturar que el motor de inferencia de tres entradas, por lo que los valores de salida son más bajos para el motor de cinco entradas. Para que los valores sean similares, los pesos de los términos índice que correspondan a preguntas tipo de las que se seleccionen más palabras clave introducen un modificador que aumenta el coeficiente hallado mediante el método basado en FL:

- Si el término índice pertenece a una pregunta tipo de la que se extraen cuatro o cinco palabras clave, el coeficiente se multiplica por 1.1.

- Si el término índice pertenece a una pregunta tipo de la que se extraen más de cinco palabras clave, el coeficiente se multiplica por 1.16.

Estos valores son experimentales y consecuencia de observar la saturación de ambos motores.

Para ilustrar el método de Asignación de Pesos basado en FL, seguimos con el ejemplo planteado en la sección 5.2.2.

El Objeto 12.6.2 está definido por la pregunta tipo

¿A qué servicios puedo acceder como Usuario Virtual de la Universidad de Sevilla?

Veamos qué ocurre en el caso de la palabra clave "Virtual"

- A nivel de Tema:

- "Virtual" aparece 2 veces en otros Temas (El valor de entrada al motor de FL asociado a P1 es 0.80 – Tabla 6.2 -).

- "Virtual" aparece 8 veces en el propio Tema 12 (El valor de entrada al motor de FL asociado a P2 es 1 – Tabla 6.6 -).

- La respuesta a P3 es "Algo" en 5 de los casos y "Nada" en 3 de ellos (El valor de entrada al motor de FL asociado a P3 es la media ponderada: (5*0.5 + 3*0) / 8 = 0.375 – Tabla 6.9 -)

- La palabra "Virtual" está ligada a una palabra en 7 ocasiones mientras que en una de ellas está ligada a dos palabras (Por tanto, está ligado a 1.1 palabras de media. Extrapolando linealmente entre los valores correspondientes a una palabra (0.7) y dos palabras (0.3) – Tabla 6.10 -, el valor de entrada al motor de FL asociado a P4 es 0.65)

- Introduciendo estos valores en el motor de lógica borrosa, el valor del peso del término índice "Virtual" para el Tema 12 resulta de $W_{ik} = 0.53$.

- A nivel de Apartado:

- "Virtual" aparece 5 veces en otros Apartados del Tema 12 (El valor de entrada al motor de FL asociado a P1 es 0.30 – Tabla 6.4 -).

- "Virtual" aparece 3 veces en el propio Apartado 12.6 (El valor de entrada al motor de FL asociado a P2 es 0.45 – Tabla 6.7 -).

- La respuesta a P3 es "Algo" en todos los casos (El valor de entrada al motor de FL asociado a P3 es 0.5 – Tabla 6.8 -).

- La palabra "Virtual" está ligada a la palabra "Usuario" (El valor de entrada al motor de FL asociado a P4 es 0.7 – Tabla 6.9 -)

- Introduciendo estos valores en el motor de lógica borrosa, el valor del peso del término índice "Virtual" para el Apartado 12.6 resulta de $W_{ik} = 0.45$.

- A nivel de Objeto:

- "Virtual" aparece 2 veces en otros Objetos del Apartado 12.6 (El valor de entrada al motor de FL asociado a P1 es 0.30 – Tabla 6.5 -).

- La respuesta a P3 es "Algo" (El valor de entrada al motor de FL asociado a P3 es 0.5 – Tabla 6.8 -).

- La palabra "Virtual" está ligada a la palabra "Usuario" (El valor de entrada al motor de FL asociado a P4 es 0.7 – Tabla 6.9 -).

- Introduciendo estos valores en el motor de FL, el valor del peso del término índice "Virtual" para el Objeto 12.6.2 resulta de $W_{ik} = 0.52$. Observamos la diferencia con el coeficiente correspondiente para el método TF-IDF (se había obtenido un valor $W_{ik} = 0.01$), pero esto es precisamente lo que buscamos: que no solo aparezcan los Objetos buscados, sino que aparezcan, aunque con un nivel de certeza menor, aquellos objetos que estén más íntimamente relacionados con este. La palabra "virtual" tiene un peso pequeño para el método TF-IDF porque no sirve para distinguir los tres objetos del Apartado 12.6. Sin embargo, en este caso, servirá para devolver todos estos Objetos, los cuales están relacionados entre sí. Los otros términos índice son los que determinan cuál es el Objeto con mayor nivel de certeza.

Asignador de coeficientes manual

Con el fin de realizar las comprobaciones necesarias sobre el correcto funcionamiento del asignador de coeficientes, se construyó un asignador de coeficientes manual separado del Agente Inteligente.

El aspecto del asignador manual de coeficientes es el mostrado en la Figura 6.4:

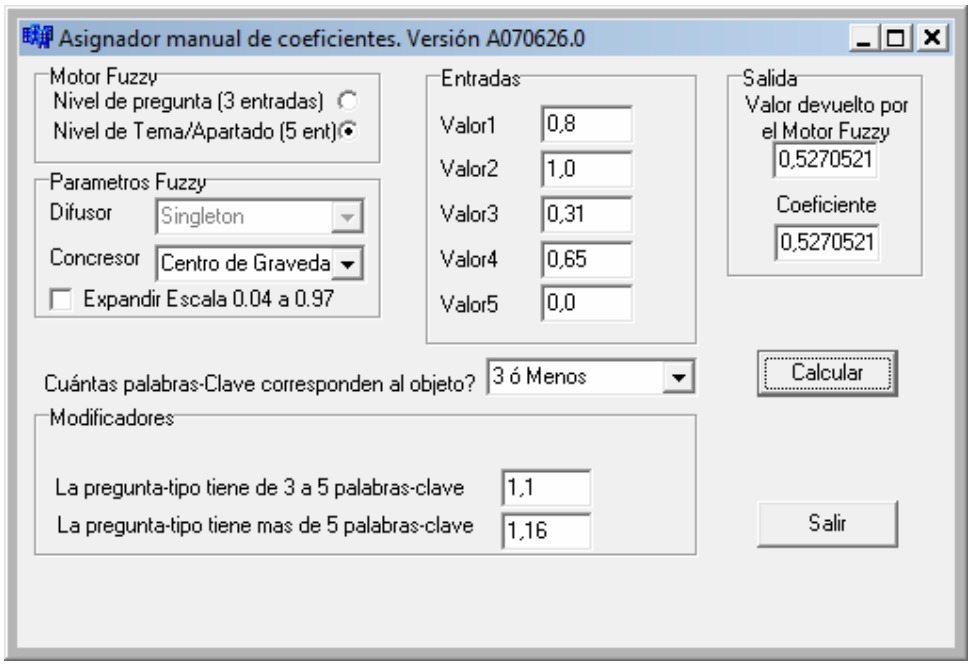


Figura 6.4. Asignador manual de coeficientes.

Aunque hasta cinco entradas sean configurables, en realidad solo se tienen en cuenta cuatro, correspondientes a las P1, P2, P3 y P4 descritas al comienzo de este apartado. En el caso del último nivel, la Pregunta P2 (¿Con qué frecuencia aparece la palabra clave entre las del subconjunto de conocimiento al que pertenece?), pierde su sentido para determinar el valor la entrada al Asignador de Coeficientes, puesto que a este nivel una palabra clave aparece únicamente una vez en cada Objeto. En este caso solo se tienen en cuenta tres valores de entrada.

En la Figura 6.4 se observan los valores introducidos para el nivel jerárquico de Tema para la palabra clave "virtual" en el ejemplo anterior y que, redondeando, proporcionaban un peso para dicho nivel jerárquico de $W_{ik} = 0.53$.

6.5. Comparación: método TF-IDF vs. método basado en FL.

6.5.1. Pruebas realizadas.

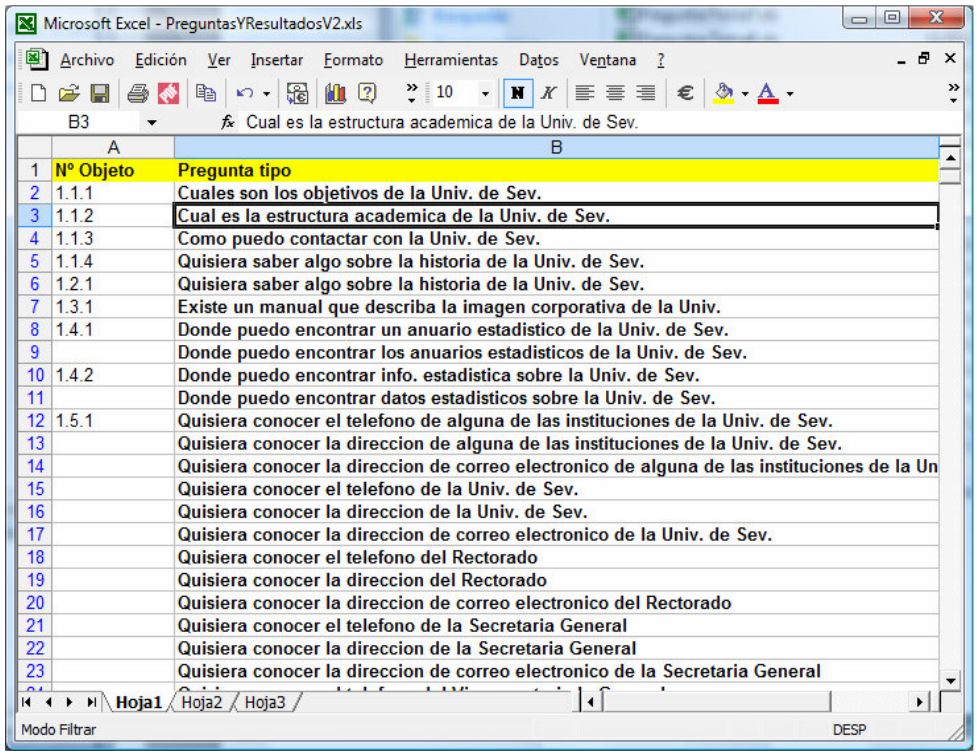
Como se comentó en la sección 5.2 de esta tesis, las pruebas fueron realizadas sobre el portal web de la Universidad de Sevilla, definiéndose 253 objetos agrupados en una estructura jerárquica con 12 Temas. Cada uno de estos Temas tiene un número variable de Apartados y de Objetos en cada uno de ellos. De estos 253 Objetos surgieron 2.107 preguntas tipo, de las cuales se eliminaron para estas pruebas algo más de la mitad, por tratarse de preguntas muy parecidas a otras y que no aportaban mucho más a dichas pruebas. Finalmente, la cifra de preguntas tipo utilizadas para las pruebas se quedó en 914 consultas. En el apartado 4.2.1 de esta tesis, se definieron distintas clases de preguntas tipo. De estas 914 consultas pertenecientes a las preguntas tipo definidas, se consideraron para las pruebas las clases mostradas en la Tabla 6.12:

Clase de pregunta tipo	Número de preguntas tipo
Preguntas tipo principales	252
Preguntas tipo que consideran sinónimos / similares	308
Preguntas tipo imprecisas	125
Preguntas tipo concretas dentro de objetos más generales	229
Preguntas tipo creadas por realimentación del sistema	0
Total de preguntas tipo	**914**

Tabla 6.12. Clases de preguntas tipo definidas para las pruebas.

Debe notarse que no se generaron preguntas tipo por realimentación del sistema para estas pruebas, puesto que estas preguntas deberían basarse en consultas, sugerencias o recomendaciones realizadas por los usuarios una vez puesta en funcionamiento la aplicación.

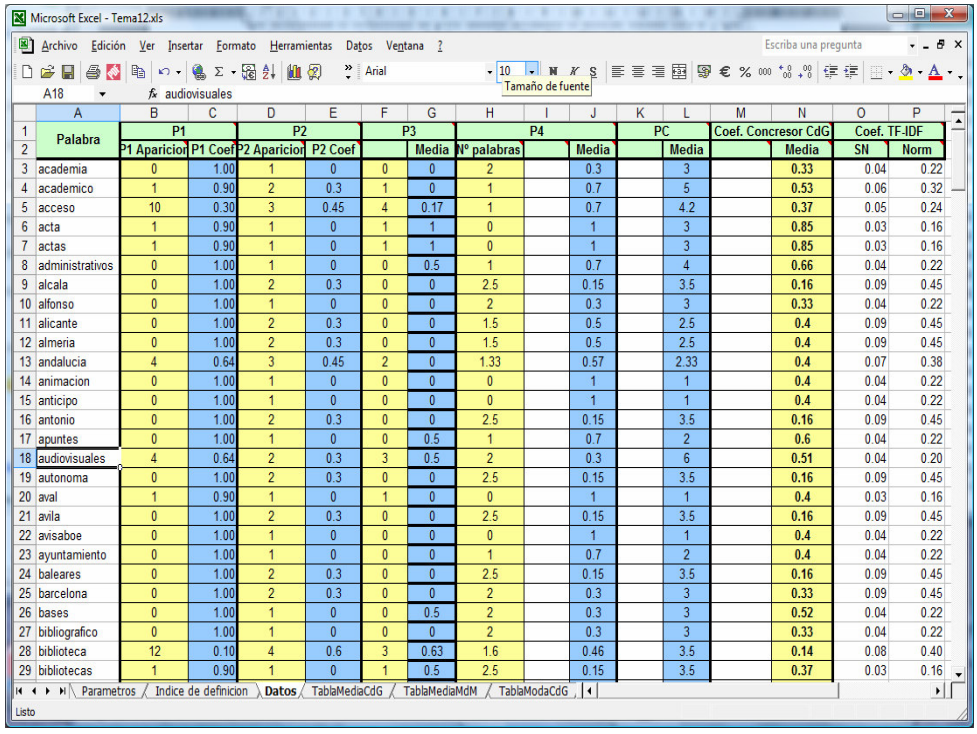
Por otro lado, en esta ocasión, a diferencia de las pruebas realizadas en el capítulo 5 de la tesis en lo referido a los parámetros del sistema de FL, y con el fin de poder llevar a cabo una mayor automatización de las pruebas, se utilizó el toolbox de lógica borrosa de MATLAB. Además, se definió una base de datos de Excel con todas las preguntas tipo (Figura 6.5) y otra con todas las respuestas asociadas a estas preguntas tipo y que se corresponderían con los Objetos. Como se puede observar en dicha Figura 6.5, existen Objetos a los que les corresponden varias preguntas tipo.



	A	B
1	Nº Objeto	Pregunta tipo
2	1.1.1	Cuales son los objetivos de la Univ. de Sev.
3	1.1.2	Cual es la estructura academica de la Univ. de Sev.
4	1.1.3	Como puedo contactar con la Univ. de Sev.
5	1.1.4	Quisiera saber algo sobre la historia de la Univ. de Sev.
6	1.2.1	Quisiera saber algo sobre la historia de la Univ. de Sev.
7	1.3.1	Existe un manual que describa la imagen corporativa de la Univ.
8	1.4.1	Donde puedo encontrar un anuario estadistico de la Univ. de Sev.
9		Donde puedo encontrar los anuarios estadisticos de la Univ. de Sev.
10	1.4.2	Donde puedo encontrar info. estadistica sobre la Univ. de Sev.
11		Donde puedo encontrar datos estadisticos sobre la Univ. de Sev.
12	1.5.1	Quisiera conocer el telefono de alguna de las instituciones de la Univ. de Sev.
13		Quisiera conocer la direccion de alguna de las instituciones de la Univ. de Sev.
14		Quisiera conocer la direccion de correo electronico de alguna de las instituciones de la Un
15		Quisiera conocer el telefono de la Univ. de Sev.
16		Quisiera conocer la direccion de la Univ. de Sev.
17		Quisiera conocer la direccion de correo electronico de la Univ. de Sev.
18		Quisiera conocer el telefono del Rectorado
19		Quisiera conocer la direccion del Rectorado
20		Quisiera conocer la direccion de correo electronico del Rectorado
21		Quisiera conocer el telefono de la Secretaria General
22		Quisiera conocer la direccion de la Secretaria General
23		Quisiera conocer la direccion de correo electronico de la Secretaria General

Figura 6.5: Base de datos de Excel con las preguntas tipo.

Así mismo, los términos índice, que en este caso se corresponden con las palabras clave seleccionadas de entre las preguntas tipo, deben quedar almacenados en archivos Excel para cada subconjunto correspondiente a cada nivel jerárquico. Además, junto a los términos índice, deben ser anotados sus correspondientes coeficientes de peso, tanto para los obtenidos con el método clásico TF-IDF, como para los obtenidos con nuestro método basado en lógica borrosa. En la Figura 6.6, vemos un ejemplo para el Tema 12. En este archivo Excel se encuentran todos los parámetros que influyen en las entradas que determinan el coeficiente de peso hallado mediante el método basado en FL y que se describieron en la sección 6.4 como P1, P2, P3 y P4. De la misma forma, también se encuentra calculado el valor de los coeficientes de peso definidos con dicho método. Por otra parte, también se almacenan los coeficientes obtenidos con el método TF-IDF.



Microsoft Excel - Tema12.xls

Archivo Edición Ver Insertar Formato Herramientas Datos Ventana ?

Escriba una pregunta

Arial 10 Tamaño de fuente

A18 audiovisuales

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Palabra	P1		P2		P3		P4			PC		Coef. Concresor CdG		Coef. TF-IDF	
2		P1 Aparicion	P1 Coef	P2 Aparicion	P2 Coef		Media	Nº palabras		Media		Media		Media	SN	Norm
3	academia	0	1.00	1	0	0	0	2		0.3		3		0.33	0.04	0.22
4	academico	1	0.90	2	0.3	1	0	1		0.7		5		0.53	0.06	0.32
5	acceso	10	0.30	3	0.45	4	0.17	1		0.7		4.2		0.37	0.05	0.24
6	acta	1	0.90	1	0	1	1	0		1		3		0.85	0.03	0.16
7	actas	1	0.90	1	0	1	1	0		1		3		0.85	0.03	0.16
8	administrativos	0	1.00	1	0	0	0.5	1		0.7		4		0.66	0.04	0.22
9	alcala	0	1.00	2	0.3	0	0	2.5		0.15		3.5		0.16	0.09	0.45
10	alfonso	0	1.00	1	0	0	0	2		0.3		3		0.33	0.04	0.22
11	alicante	0	1.00	2	0.3	0	0	1.5		0.5		2.5		0.4	0.09	0.45
12	almeria	0	1.00	2	0.3	0	0	1.5		0.5		2.5		0.4	0.09	0.45
13	andalucia	4	0.64	3	0.45	2	0	1.33		0.57		2.33		0.4	0.07	0.38
14	animacion	0	1.00	1	0	0	0	0		1		1		0.4	0.04	0.22
15	anticipo	0	1.00	1	0	0	0	0		1		1		0.4	0.04	0.22
16	antonio	0	1.00	2	0.3	0	0	2.5		0.15		3.5		0.16	0.09	0.45
17	apuntes	0	1.00	1	0	0	0.5	1		0.7		2		0.6	0.04	0.22
18	audiovisuales	4	0.64	2	0.3	3	0.5	2		0.3		6		0.51	0.04	0.20
19	autonoma	0	1.00	2	0.3	0	0	2.5		0.15		3.5		0.16	0.09	0.45
20	aval	1	0.90	1	0	1	0	0		1		1		0.4	0.03	0.16
21	avila	0	1.00	2	0.3	0	0	2.5		0.15		3.5		0.16	0.09	0.45
22	avisaboe	0	1.00	1	0	0	0	0		1		1		0.4	0.04	0.22
23	ayuntamiento	0	1.00	1	0	0	0	1		0.7		2		0.4	0.04	0.22
24	baleares	0	1.00	2	0.3	0	0	2.5		0.15		3.5		0.16	0.09	0.45
25	barcelona	0	1.00	2	0.3	0	0	2		0.3		3		0.33	0.09	0.45
26	bases	0	1.00	1	0	0	0.5	2		0.3		3		0.52	0.04	0.22
27	bibliografico	0	1.00	1	0	0	0	2		0.3		3		0.33	0.04	0.22
28	biblioteca	12	0.10	4	0.6	3	0.63	1.6		0.46		3.5		0.14	0.08	0.40
29	bibliotecas	1	0.90	1	0	1	0.5	2.5		0.15		3.5		0.37	0.03	0.16

Parametros | Indice de definicion | Datos | TablaMediaCdG | TablaMediaMdM | TablaModaCdG

Listo

Figura 6.6: Base de datos con los coeficientes de peso para ambos métodos.

Las preguntas tipo fueron introducidas como consultas de usuario en un sistema basado en FL creado en MATLAB, proporcionando dicho sistema como salidas los Objetos que sean seleccionados con una certeza mayor que un determinado umbral. Para comparar resultados, se consideró la posición en la que aparece la respuesta correcta entre el total de respuestas identificadas como probables.

Para ello, lo primero que hay que hacer es definir los umbrales de superación del motor lógico. De esta forma, se eliminan Temas y Apartados que no tengan relación con el Objeto a identificar. Esta una de las ventajas de utilizar la estructura jerárquica, ya que se gana en tiempo de procesamiento al ir descartando subconjuntos de conocimiento. En todo caso, se intenta no cercenar demasiados Objetos, con el fin de poder también obtener objetos relacionados. Lo ideal es que se presenten entre una y cinco respuestas a la consulta de usuario, dependiendo de la cantidad de Objetos relacionados con el de la consulta que se encuentre en el conjunto de conocimiento. Tal y como se explicó en el apartado anterior, los coeficientes son más bajos con el método TF-IDF debido a la normalización, por lo que, tras realizar algunas pruebas, los umbrales se fijaron a 0.2 para superar el nivel de Tema y 0.3 para superar el nivel de Apartado para el método TF-IDF, mientras que ambos umbrales tienen un valor 0.4 para el método basado en FL.

El resultado de la consulta se clasifica en 5 categorías:

- **Categoría Cat1.-** La respuesta correcta a aparece como respuesta única o es la que tiene mayor certeza de entre las ofrecidas por el sistema.

- **Categoría Cat2.-** La respuesta correcta aparece entre las 3 con mayor certeza (excluyendo el caso anterior).

- **Categoría Cat3.-** La respuesta correcta entre las 5 con mayor certeza (excluyendo los casos anteriores).

- **Categoría Cat4.-** La respuesta correcta aparece, pero no entre las 5 con mayor certeza.

- **Categoría Cat5.-** La respuesta correcta no aparece entre las ofrecidas por el sistema.

Lo ideal es que la respuesta aparezca en la Categoría Cat1, aunque será razonablemente aceptable que la respuesta se encuentre en las Categorías Cat2 y Cat3.

El sistema de FL es el que se consideró óptimo en el apartado 5.3, es decir, contempla las siguientes características:

- Las variables de entrada al sistema de FL son los coeficientes de peso W_{ik} correspondientes a los términos índice *i* en cada subconjunto *k*. Esto se repite para todos los niveles jerárquicos en los subconjuntos cuya salida supera un cierto umbral.

- Según el número de términos índice extraídos (es decir, el número de palabras clave en cada consulta), se utilizará el motor borroso de tres entradas o el motor borroso de cinco entradas.

- Existen tres conjuntos borrosos de entrada, correspondientes a los valores de las entradas BAJO, MEDIO y ALTO y cuatro conjuntos borrosos de salida, los cuales corresponden a los valores BAJO, MEDIO-BAJO, MEDIO-ALTO y ALTO.

- Las reglas borrosas utilizadas son las definidas en el apartado 5.3 de esta tesis.

- Los subconjuntos correspondientes a cada nivel jerárquico para los que la salida no supera un determinado umbral (en nuestro caso, el mencionado unas líneas arriba) son eliminados. Para aquellos subconjuntos cuya salida sí supera este umbral, el proceso se repite hasta llegar al nivel de Objeto. En el caso de que ningún subconjunto supere el umbral, se contempla la posibilidad de bajar los umbrales hasta un determinado nivel.

Una vez introducidas las 914 consultas, los resultados obtenidos se observan en la Tabla 6.13.

Método empleado	Cat1	Cat2	Cat3	Cat4	Cat5	Total
Método TF-IDF	466 (50.98%)	223 (24.40%)	53 (5.80%)	79 (8.64%)	93 (10.18%)	914
Método FL	710 (77.68%)	108 (11.82%)	27 (2.95%)	28 (3.06%)	41 (4.49%)	914

Tabla 6.13. Resultados obtenidos con los métodos TF-IDF y basado en FL.

Aunque los resultados obtenidos con el método TF-IDF son bastante razonables, detectándose el 81.18% de los Objetos entre las 5 primeras opciones (con más de la mitad de los Objetos devueltos en primer lugar), el método basado en FL resulta claramente superior, con un 92.45 % de Objetos devueltos (y más de las tres cuartas partes en primer lugar). A continuación, en las Figuras 6.7 y 6.8 se muestran los resultados en un gráfico circular, en el que se pueden observar mejor las conclusiones a las que se ha llegado.

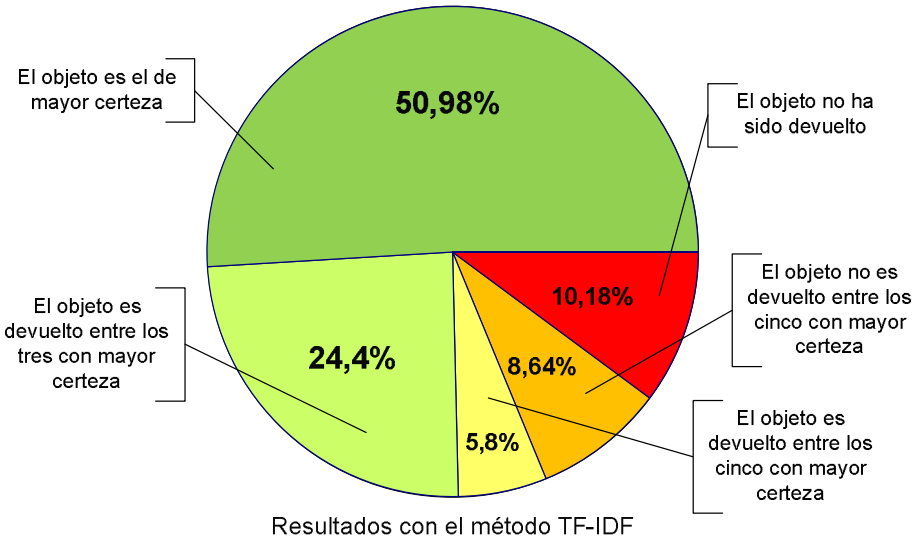


Figura 6.7: Resultados con el método TF-IDF.

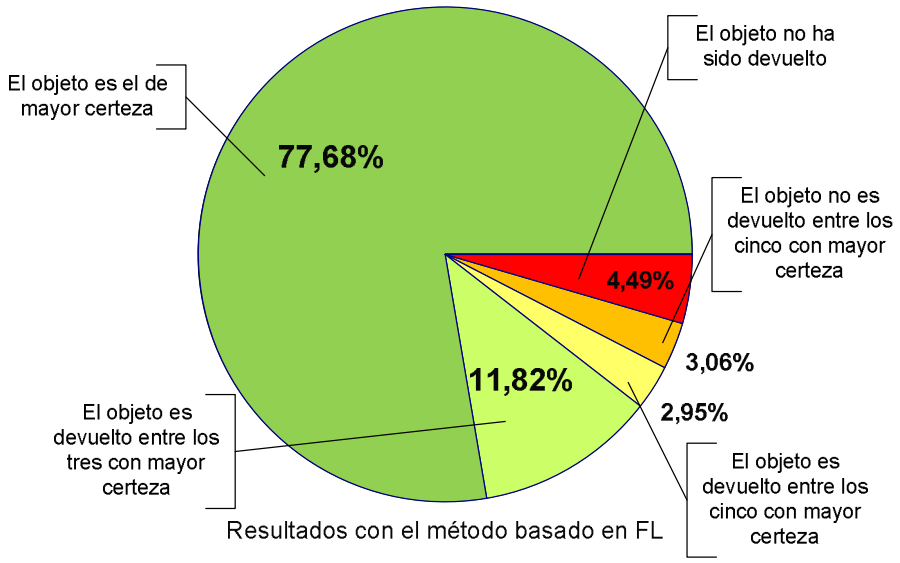


Figura 6.8: Resultados con el método basado en FL.

6.5.2 Análisis de resultados.

Análisis según las clases de preguntas tipo.

Con el fin de refinar las conclusiones acerca de los dos métodos de TW, es importante realizar un análisis más minucioso de los resultados. En la Tabla 6.12 se exponía la distinta naturaleza de las preguntas tipo introducidas en el sistema. Estas clases de preguntas tipo ya fueron definidas en la sección 4.2.1 de esta tesis. En estas pruebas se han utilizado las siguientes clases:

- Preguntas tipo principales, que son las que mejor definen al Objeto, es decir, corresponderían a una consulta de usuario perfecta.

  Un ejemplo de esta clase de preguntas tipo de entre las definidas para el portal web de la Universidad de Sevilla sería:

  Me gustaría obtener información acerca del Servicio de Asistencia Religiosa de la Universidad de Sevilla.

- Preguntas tipo que utilicen sinónimos con respecto a la pregunta tipo principal o que sean similares con respecto a su estructura.

Para la pregunta tipo principal anterior, y dado que las siglas del Servicio de Asistencia Religiosa de la Universidad de Sevilla son SARUS, una pregunta tipo similar sería la siguiente:

Me gustaría obtener información acerca del SARUS.

- Preguntas tipo imprecisas, a las que les falte información o no estén del todo bien formuladas.

  Para la pregunta tipo principal de los ejemplos anteriores, y teniendo en cuenta que las misas organizadas en la Universidad de Sevilla son gestionadas por el SARUS, una pregunta tipo imprecisa es:

  Me gustaría conocer el horario de misas de la Universidad de Sevilla.

- Preguntas tipo concretas, las cuales tienen relación con algún Objeto, pero se refieren a algo más preciso.

  Para el ejemplo anterior no existe ninguna pregunta tipo de esta clase, pero un ejemplo en el portal web de la Universidad de Sevilla sería el siguiente:

  Para la pregunta tipo principal

  Me gustaría saber donde se encuentran las aulas de informática de la Universidad de Sevilla.

  una pregunta tipo concreta relacionada sería

  Me gustaría saber donde se encuentra el aula de informática del Campus Ramón y Cajal.

Por tanto, si sometemos a ambos métodos de TW a un análisis exhaustivo teniendo en cuenta la clase de pregunta tipo, podremos llegar a mejores conclusiones acerca de dichos métodos. Para una visión más intuitiva de los resultados, estos han sido representados en diagramas circulares.

En primer lugar, sometemos a análisis los resultados obtenidos para las preguntas tipo principales, utilizando las categorías Cat1, Cat2, Cat3, Cat4 y Cat5 definidas previamente en este apartado. Se pueden observar estos resultados en la Figura 6.9.

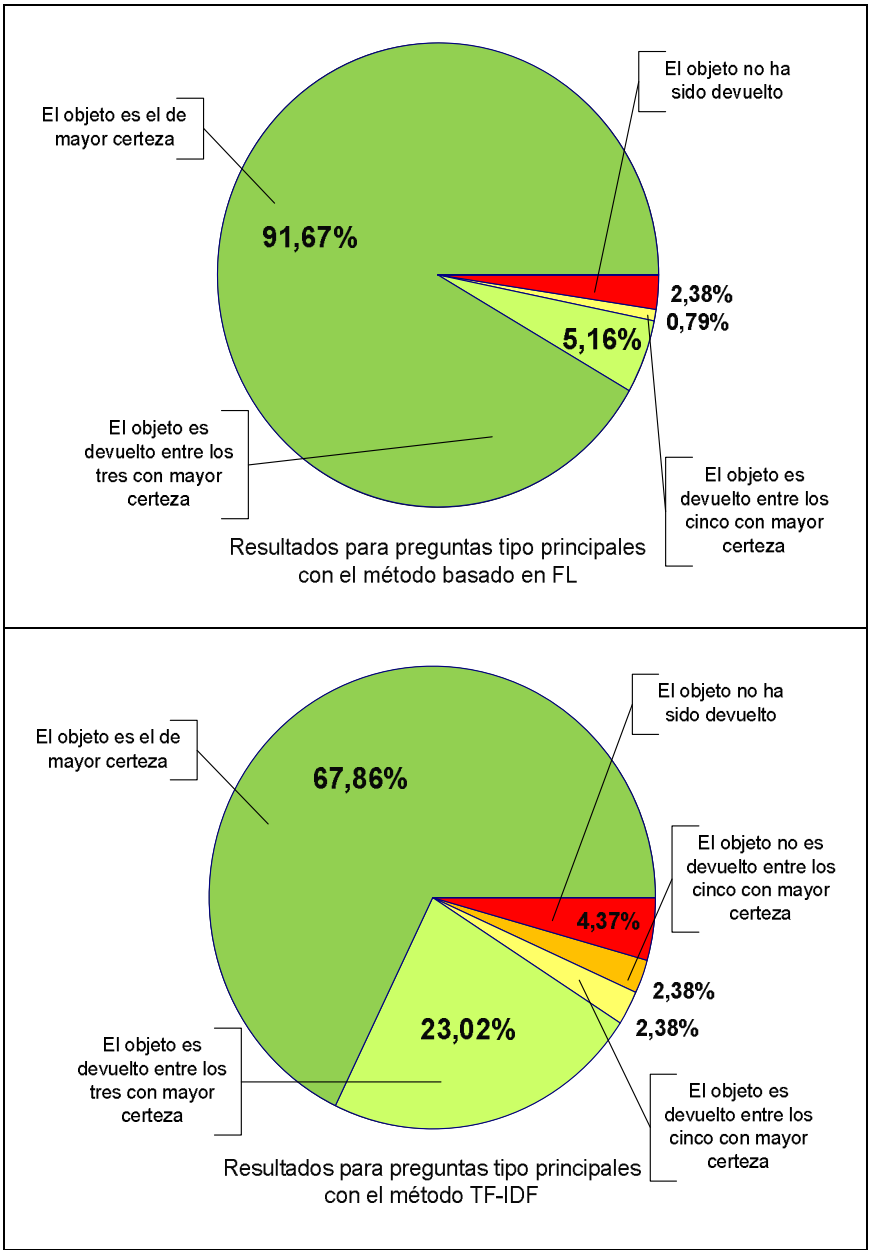


Figura 6.9: Comparación de resultados para preguntas tipo principales para ambos métodos de TW.

Observando los resultados, el método TF-IDF resiste relativamente bien la comparación teniendo en cuenta el número de Objetos devueltos, aunque el método basado en FL es mucho más preciso, devolviendo el 91.67 % de los Objetos en el primer lugar. Por otra parte, los

buenos resultados para esta clase de pregunta tipo son lógicos, habida cuenta de las preguntas tipo corresponden a consultas de usuario supuestamente bien realizadas.

Para las preguntas tipo utilizando sinónimos o una estructura similar, las conclusiones son similares: los resultados obtenidos utilizando los coeficientes de peso generados mediante el método basado en FL son mejores que los conseguidos con el método TF-IDF, sobre todo en lo que se refiere a la precisión. Así mismo, se devuelve el Objeto requerido en más ocasiones para el método basado en FL aunque, como en el caso anterior, el método TF-IDF también asegura buenos resultados en este sentido, como se puede ver en la Figura 6.10.

Sin embargo, al no ser las consultas exactas, se puede observar que los resultados empeoran más para el método TF-IDF que para el método basado en FL, lo que nos da una idea de lo acertado de utilizar la lógica borrosa para añadir una mayor flexibilidad al sistema. En todo caso, los resultados son bastante parecidos a los obtenidos para las preguntas tipo principales, empeorando los resultados sólo ligeramente (a fin de cuentas se trata de preguntas tipo similares), tanto para el método basado en FL como para el método TF-IDF.

.

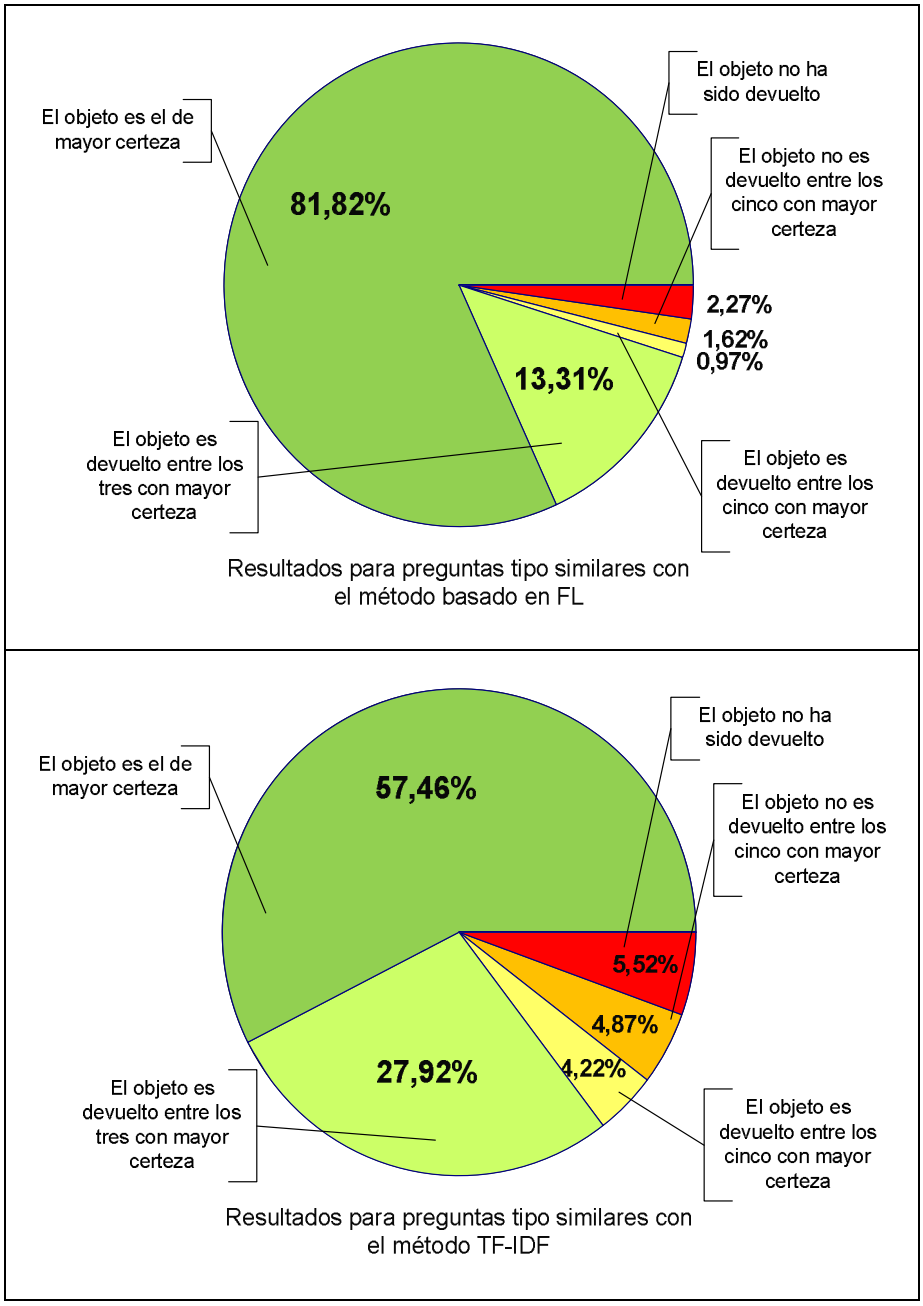


Figura 6.10: Comparación de resultados para preguntas tipo similares para ambos métodos de TW.

La diferencia es aún más notoria para las preguntas tipo imprecisas y para las que hemos denominado preguntas tipo concretas. Los resultados conseguidos para ambas clases de preguntas tipo se muestran en las Figuras 6.11 (preguntas tipo imprecisas) y 6.12 (preguntas tipo concretas).

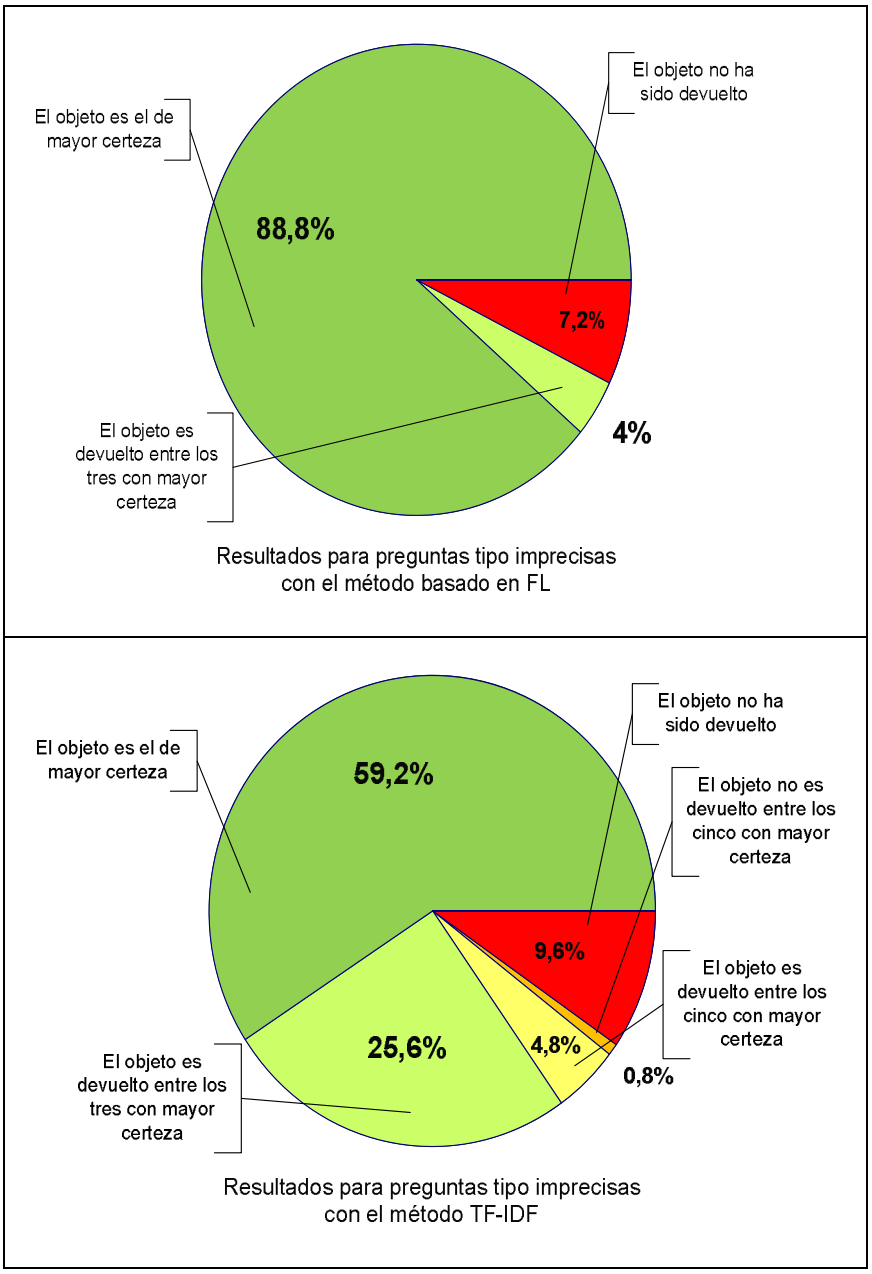


Figura 6.11: Comparación de resultados para preguntas tipo imprecisas para ambos métodos de TW.

Las preguntas tipo imprecisas son casi tan bien detectadas como las preguntas tipo principales en el caso de los coeficientes de pesos hallados mediante el método basado en FL. Esto constituye otra razón más para confirmar la idoneidad del uso de lógica borrosa.

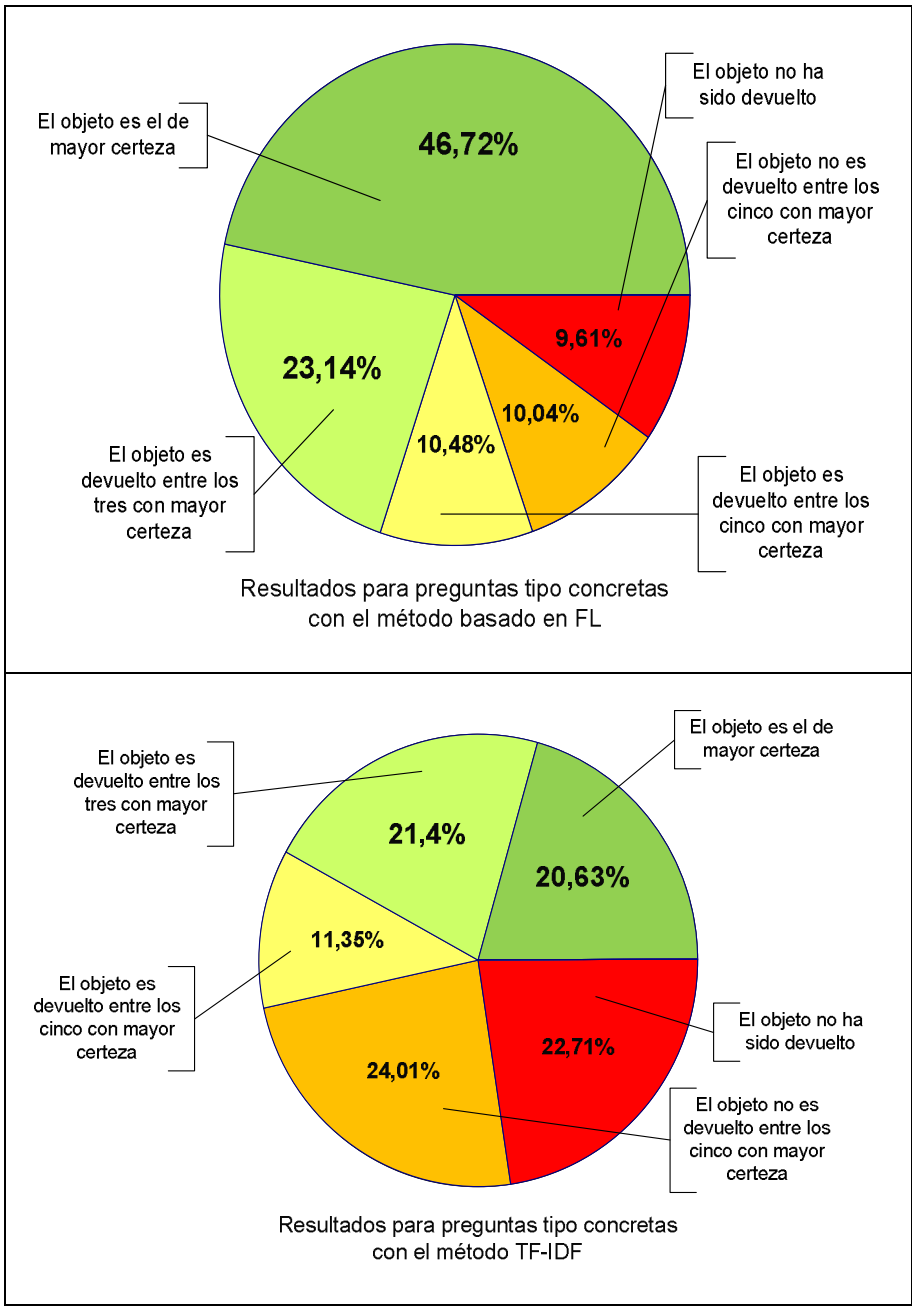


Figura 6.12: Comparación de resultados para preguntas tipo concretas para ambos métodos de TW.

Como se puede observar en la Figura 6.12, la clase de preguntas tipo a las que se ha denominado concretas arroja el peor resultado con diferencia entre todas las clases de preguntas tipo introducidas como consulta de usuario. Esto es lógico, teniendo en cuenta que se trata de preguntas asociadas a la pregunta tipo principal, pero más concretas, con lo cual la información que proporcionaría el Objeto con respecto a este tipo de consulta sería limitada. En realidad, lo habitual para las preguntas tipo concretas es que se refieran a un listado dentro de un todo, con lo que pudieran existir (y de hecho existen) Objetos más relacionados con la consulta que el propio Objeto requerido: esto es en absoluto un inconveniente, puesto que el Agente Inteligente presentaría ambos Objetos (el más concreto y el más general) al usuario, siendo este el que debe elegir cual de ellos le interesa más.

Nuevamente, y en este caso de forma más evidente, el hecho de usar los coeficientes basados en FL permite encontrar un mayor número de objetos.

A modo de resumen, se muestran los resultados para cada clase de pregunta tipo en la Tabla 6.14.

Tipo de pregunta		Cat1	Cat2	Cat3	Cat4	Cat5	Total
Preguntas tipo principales	**Método TF-IDF**	171 (67.86%)	58 (23.02%)	6 (2.38%)	6 (2.38%)	11 (4.37%)	252
	Método FL	231 (91.67%)	13 (5.16%)	2 (0.79%)	0 (0.00 %)	6 (2.38%)	252
Preguntas tipo similares	**Método TF-IDF**	177 (57.46%)	86 (27.92%)	13 (4.22%)	15 (4.87%)	17 (5.52%)	308
	Método FL	252 (81.82%)	41 (13.31%)	3 (0.97%)	5 (1.62%)	47 (2.27%)	308
Preguntas tipo imprecisas	**Método TF-IDF**	74 (59.20%)	223 (24.40%)	53 (5.80%)	79 (8.64%)	93 (10.18%)	125
	Método FL	111 (88.80%)	5 (4.00%)	0 (0.00 %)	0 (0.00 %)	9 (7.20%)	125
Preguntas tipo concretas	**Método TF-IDF**	46 (20.08%)	49 (21.40%)	26 (11.35%)	55 (24.01%)	52 (22.71%)	229
	Método FL	107 (46.72%)	53 (23.14%)	24 (10.48%)	23 (10.04%)	22 (9.61%)	229

Tabla 6.14: Resumen de resultados para las distintas clases de preguntas tipo.

Análisis según el número de preguntas tipo de cada Objeto.

Otro aspecto a tener en cuenta en el análisis de resultados es el número de preguntas tipo asignadas a cada Objeto. Evidentemente, un Objeto que quede bien definido por una sola pregunta tipo es un Objeto muy concreto y, por tanto, fácilmente extraíble del conjunto completo de conocimiento. Sin embargo, hay Objetos que pueden contener información muy difusa o imprecisa, por lo que es necesaria la definición de varias preguntas tipo, como se explicó en el apartado anterior de esta tesis.

Para realizar este estudio, se han agrupado los Objetos en los siguientes grupos:

- Grupo 1: el Objeto está definido por una única pregunta tipo.
- Grupo 2: el Objeto está definido por entre dos y cinco preguntas tipo.
- Grupo 3: el Objeto está definido por entre seis y diez preguntas tipo.
- Grupo 4: el Objeto está definido por más de diez preguntas tipo.

Lógicamente, los Grupos 1 y 2 son más numerosos, puesto que es menos habitual que muchas preguntas tengan la misma respuesta (lo que correspondería a un Objeto). Sin embargo, estos Objetos de los Grupos 3 y 4 equivalen a un amplio abanico de preguntas tipo, por lo que son igualmente importantes. En la Tabla 6.15 se define el número de Objetos para cada uno de estos grupos.

Nombre de grupo	Número de preguntas tipo por Objeto	Número de Objetos
Grupo 1	1	95
Grupo 2	2-5	108
Grupo 3	6-10	22
Grupo 4	> 10	28

Tabla 6.15: Agrupación de Objetos según el número de preguntas tipo por Objeto.

Para analizar los resultados, se considera en que posición se devuelve el Objeto buscado en la mayoría de las preguntas tipo que definen ese Objeto, es decir, si un Objeto está definido, por ejemplo, por 15 preguntas tipo y en 10 de ellas dicho Objeto se devuelve en segundo lugar, se considera que este Objeto ha sido efectivamente devuelto en segundo lugar.

En definitiva, este estudio no se centra en las respuestas dadas a las preguntas tipo, sino en los Objetos correctamente devueltos por el sistema, lo que proporciona un nuevo elemento de análisis del sistema.

En la Figura 6.13, se plasman los resultados correspondientes al Grupo 1, en el que solo se define una pregunta tipo por Objeto.

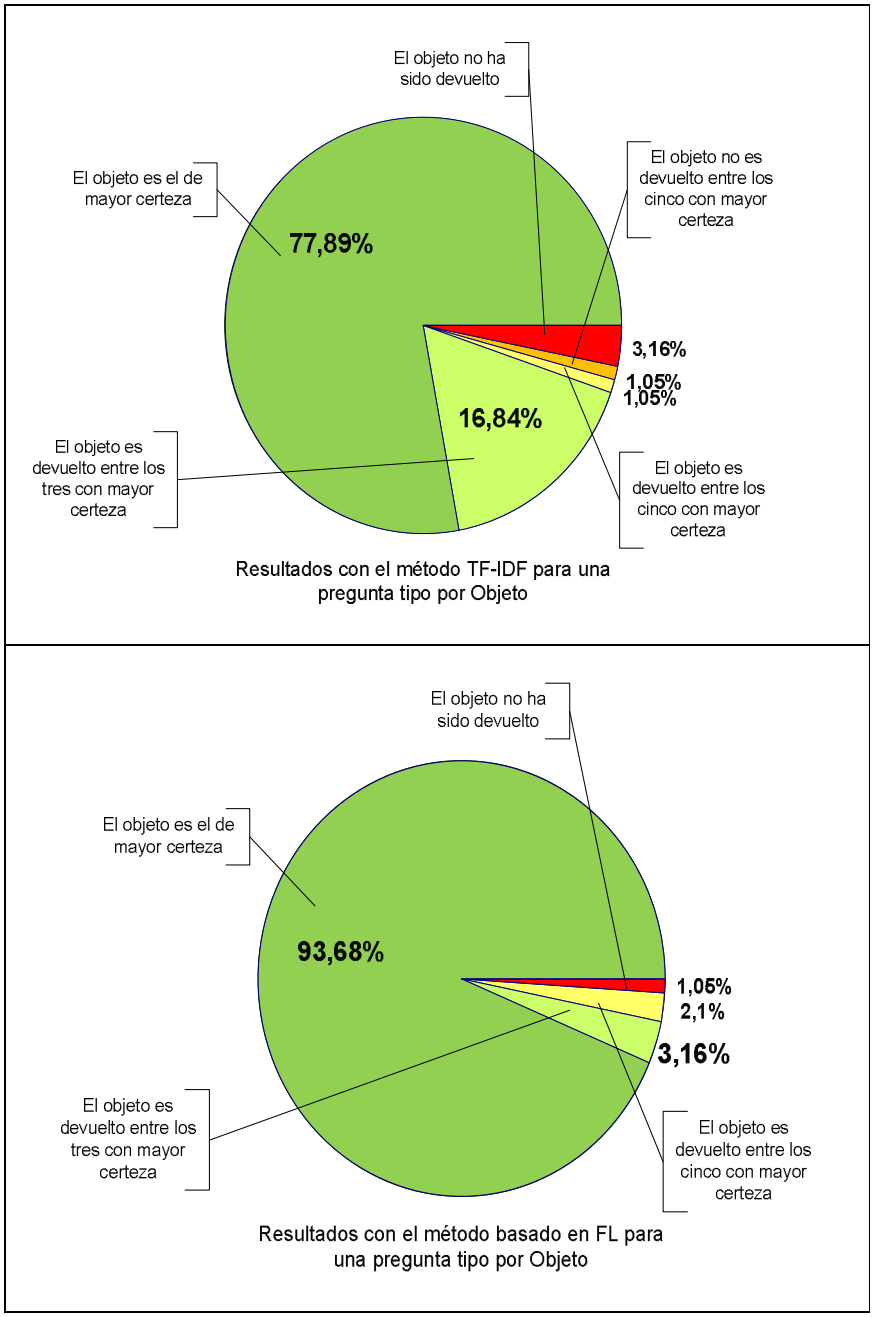


Figura 6.13: Comparación de resultados para ambos métodos de TW para el caso de una pregunta tipo por Objeto.

Los resultados son prácticamente perfectos para el método basado en FL, puesto que prácticamente devuelve todos los objetos (cerca del 94 %) en primer lugar. Sin embargo, el método TF-IDF, aun no siendo tan preciso, resiste relativamente bien la comparación, si no por

el porcentaje de Objetos devueltos en primer lugar, sí porque es capaz de devolver casi el 94 % de los Objetos entre los tres Objetos con mayor certeza.

Este comportamiento se repite prácticamente para el que se ha denominado Grupo 2, en el que se definen entre dos y cinco preguntas tipo por Objeto, como se puede observar en la Figura 6.14. Los objetos se devuelven en ambos casos entre los tres primeros Objetos con bastante frecuencia, siendo el caso del método FL nuevamente destacable por su gran precisión, al devolver más del 92 % de los Objetos en primer lugar.

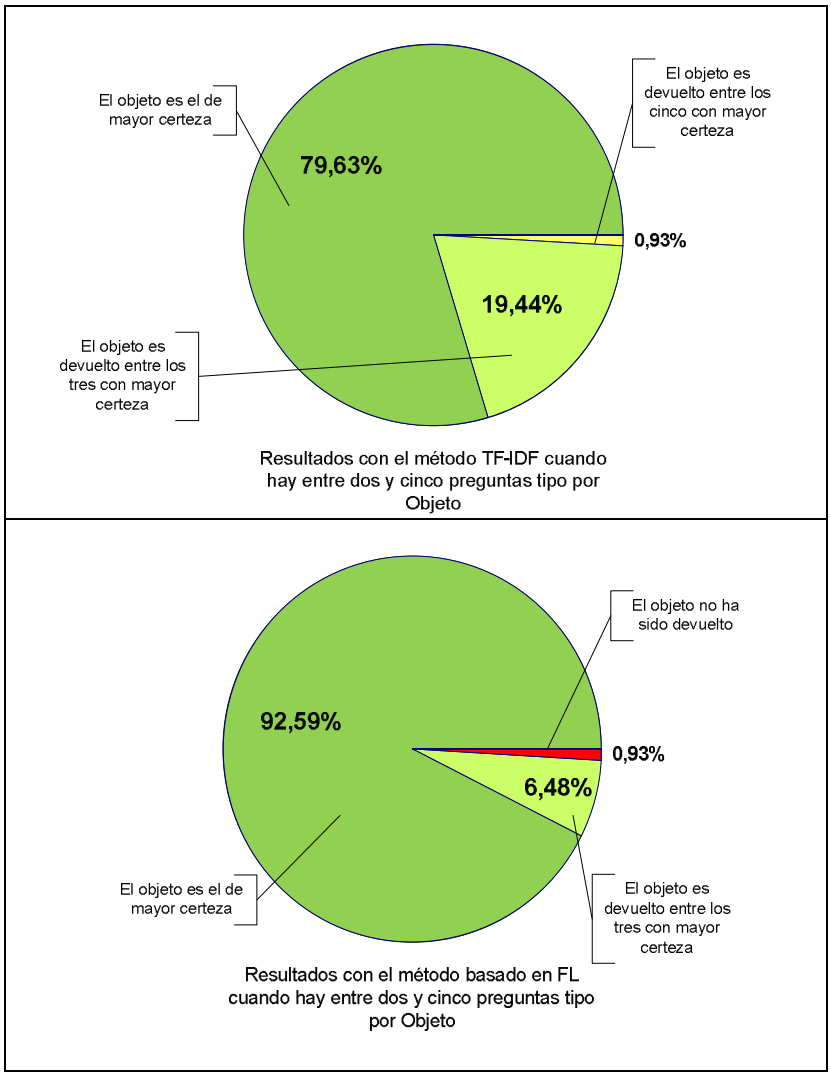


Figura 6.14: Comparación de resultados para ambos métodos de TW para el caso de una pregunta tipo por Objeto.

A la vista de lo obtenido hasta ahora en las últimas pruebas, se puede concluir que, para ambos métodos, los resultados son muy buenos cuando se definen hasta cinco preguntas tipo por Objeto, con lo que se validan también los resultados obtenidos en el capítulo anterior de esta tesis para los parámetros ya comentados del Agente Inteligente (conjuntos borrosos, número de entradas y salidas del sistema, parámetros del sistema borroso, etc.). Aunque los resultados son mejores para el nuevo método de definición de pesos propuesto, basado en FL, los resultados son también bastante razonables para el método de asignación de pesos clásico TF-IDF.

Sin embargo, la mayor ventaja de la asignación de pesos mediante FL aparece cuando se definen más preguntas tipo por Objeto, es decir, cuando la información es más confusa, desordenada o imprecisa. En la Figura 6.15 se exponen los resultados para el caso denominado como Grupo 3, en el que se definen entre seis y diez preguntas tipo por Objeto. Se puede observar que en este caso ya empieza a haber una diferencia notoria entre el método TF-IDF clásico y el método basado en FL propuesto en esta tesis. Aunque ambos métodos devuelven todos los Objetos, existe una gran diferencia en el lugar en el que son devueltos, es decir, en la precisión en la extracción de información (86 % de Objetos extraídos en primer lugar para el método basado en FL por únicamente el 45 % del método clásico TF-IDF).

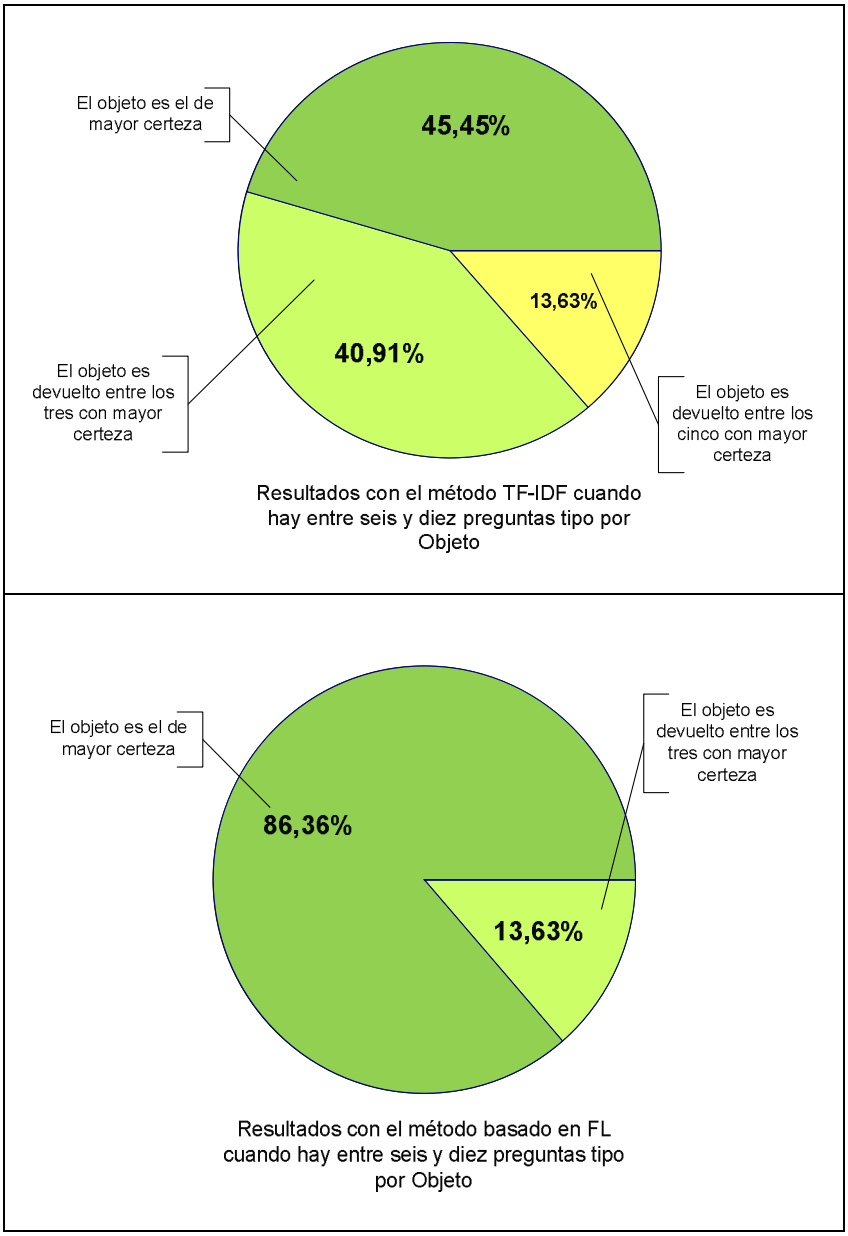


Figura 6.15: Comparación de resultados para ambos métodos de TW para el caso de una pregunta tipo por Objeto.

La diferencia es aún más notable en los casos en los que se definen más de diez preguntas tipo por Objeto. En este caso, está claro que pocas o ninguna de las preguntas tipo definen claramente al Objeto, por lo que la información es claramente muy difusa. En la Figura 6.16, se pueden ver los resultados y las evidentes y claras diferencias en estos casos. Mientras que con el método basado en FL, se devuelven más del 96 % de los Objetos (con el 75 % de ellos en

primer lugar), el método clásico TF-IDF solo devuelve correctamente el 82 % de los Objetos. Además, solo el 35.7 % de estos Objetos, son extraídos en primer lugar.

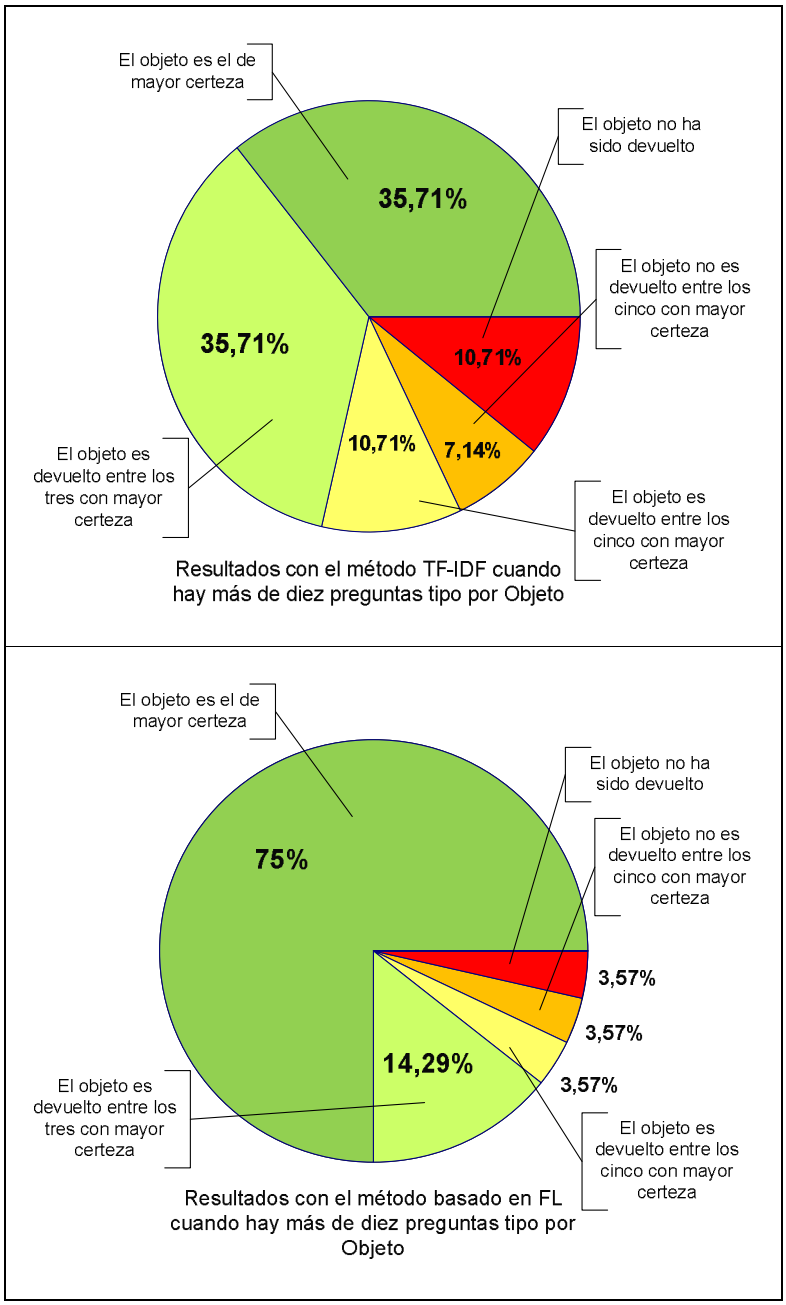


Figura 6.16: Comparación de resultados para ambos métodos de TW para el caso de más de diez preguntas tipo por Objeto.

A modo de resumen, se muestran los resultados para cada clase de pregunta tipo en la Tabla 6.16.

Tipo de pregunta		Cat1	Cat2	Cat3	Cat4	Cat5	Total
Preguntas tipo Grupo 1	**Método TF-IDF**	74 (77.89%)	16 (16.84%)	1 (1.05%)	1 (1.05%)	3 (3.16%)	95
	Método FL	89 (93.68%)	3 (3.16%)	2 (2.10%)	0 (0.00 %)	1 (1.05%)	95
Preguntas tipo Grupo 2	**Método TF-IDF**	86 (79.63%)	21 (19.44%)	1 (0.93%)	0 (0.00 %)	0 (0.00 %)	108
	Método FL	100 (92.59%)	7 (6.48%)	0 (0.00 %)	0 (0.00 %)	1 (0.93%)	108
Preguntas tipo Grupo 3	**Método TF-IDF**	10 (45.45%)	9 (40.91%)	3 (13.63%)	0 (0.00 %)	0 (0.00 %)	22
	Método FL	19 (86.36%)	3 (13.63%)	0 (0.00 %)	0 (0.00 %)	0 (0.00 %)	22
Preguntas tipo Grupo 4	**Método TF-IDF**	10 (35.71%)	10 (35.71%)	3 (10.71%)	2 (7.14%)	3 (10.71%)	28
	Método FL	21 (75.00%)	4 (14.29%)	1 (3.57%)	1 (3.57%)	1 (3.57%)	28

Tabla 6.16: Resumen de resultados para las distintas clases preguntas tipo.

A la vista de la tabla, se puede observar que, mientras más preguntas tipo hay por Objeto, mejores son los resultados del método basado en FL, comparados con los obtenidos mediante el método clásico TF-IDF. Por tanto, la evidente conclusión es que, mientras más enrevesada, desordenada y confusa es la información, mejor es el método de asignación de pesos basado en FL comparado con el método TF-IDF, lo que provoca que su aplicación sea ideal para el caso de un Agente Inteligente para un portal web, en el que la información tiene estas características y en el que los usuarios pueden realizar consultas inexactas o desorientadas.

Capítulo 7. Resumen, conclusiones y futuras líneas de trabajo.

Este capítulo constituye el punto y final de esta tesis. En él se resume todo el trabajo realizado para esta tesis, posteriormente se exponen las conclusiones obtenidas y, finalmente, se esbozan las posibles líneas de continuación de la investigación en el marco en el que se encuadra esta tesis doctoral.

7.1. Resumen.

En esta tesis se ha elaborado un nuevo método de Extracción de Información basado en el uso de la Lógica Borrosa para un conjunto de conocimiento general, aplicando este método, en particular, a portales web. Para ello, ha sido necesario realizar las siguientes tareas:

1. La elaboración de un estudio del estado del arte en cuanto a la Búsqueda de Información (*Information Retrieval*, IR) y la Extracción de Información (*Information Extraction*, IE) y las distintas técnicas existentes para llevarlas a cabo, entre las que destacan el Modelo de Espacio Vectorial (*Vector Space Model*, VSM) y las denominadas ontologías, así como las razones que nos han llevado a elegir el primero de estos modelos (VSM), para la realización del modelo basado en lógica borrosa propuesto en esta tesis.

2. El análisis del estado del arte de las aplicaciones de Lógica Borrosa a la búsqueda de conocimiento, tanto en el enfoque vectorial como en el enfoque semántico (basado en ontologías).

3. La confección de un método general de búsqueda de conocimiento mediante el uso de un Agente Inteligente basado en la Lógica Borrosa. Para ello, se presentan Agentes Inteligentes o Asistentes Virtuales, desarrollados tanto en el ámbito de la investigación como en el ámbito comercial y, habida cuenta de la falta de flexibilidad de estos Agentes Inteligentes cuando la información es abundante, confusa, imprecisa o heterogénea, se propone un método de extracción de conocimiento basado en el Modelo de Espacio Vectorial y la Lógica Borrosa. Para ello, se divide el conjunto de conocimiento en distintos niveles jerárquicos hasta llegar a un último nivel en el que se encuentran las instancias u Objetos, asignándose a cada Objeto una serie de preguntas tipo, basadas en las posibles consultas de un usuario en Lenguaje Natural. Estas preguntas tipo generadas conllevan la extracción de términos índice, cada uno con un cierto peso asignado.

El desarrollo de este método constituye una de las dos aportaciones principales de esta tesis a la comunidad científica.

4. La validación del método mediante la aplicación a la extracción de información en portales web. La información contenida en un portal web cumple todas las características antes mencionadas, que hacen que la Lógica Borrosa sea una herramienta muy útil a la hora de extraer información relevante. Para ello se realiza el estudio de los parámetros necesarios en el sistema de Lógica Borrosa para obtener los mejores resultados utilizando el portal web de la Universidad de Sevilla como marco para las pruebas.

5. Por último, y dada la necesidad de automatizar los pesos generados para los distintos términos índice derivados de la aplicación del Modelo de Espacio Vectorial, se propone un nuevo método para la Asignación de Pesos (*Term Weighting*, TW), basado así mismo en el uso de la Lógica Borrosa, sustituyendo este método al método clásico, denominado TF-IDF.

La introducción de la Lógica Borrosa a la asignación de pesos representa la segunda aportación importante de esta tesis.

7.2. Conclusiones.

Del desarrollo de esta tesis, se obtienen las siguientes conclusiones:

1. En el campo de IR e IE existe la posibilidad de trabajar con dos enfoques distintos: el modelo vectorial y el modelo basado en webs semánticas. Si en el modelo vectorial, la recuperación y extracción de información se basa en el *qué* de la información, en el caso de las webs semánticas se basa en el *cómo* está estructurada dicha información para recuperarla o extraerla. El problema que surge es que, en la actualidad, la web no provee aún de un gran número de ontologías o esquemas: hay pocas disponibles y en muy pocas materias. Es más, construir una ontología desde el principio puede resultar una tarea costosa y muy dependiente del ingeniero de conocimiento que la desarrolle. Por estas razones, en esta tesis nos inclinamos por un enfoque vectorial.

2. Para el diseño del Agente Inteligente para la búsqueda de información en entornos web, debido a la gran cantidad de información a manejar y la estructura jerárquica o posibilidad de agrupamiento de esta información, nos decidimos en esta tesis por el uso de la Lógica Borrosa.

3. El hecho de manejar información heterogénea estructurada de forma jerárquica de esta manera proporciona la ventaja de agrupar la información en *clusters* con contenidos relacionados. Esto permite que el Agente Inteligente pueda ofrecer al usuario no solo el Objeto más cercano a su petición, sino también Objetos que estén relacionados y que le pueden resultar interesantes. La lógica borrosa proporciona la flexibilidad necesaria para manejar este tipo de información. A todos los efectos, este hecho proporciona dos ventajas importantes: ahorra tiempo de computación y

elimina gran cantidad de información que ya no es necesaria y que no será presentada al usuario con toda seguridad.

4. El diseño del sistema de Lógica Borrosa está basado en las pruebas realizadas en el portal web de la Universidad de Sevilla para optimizar el método general de búsqueda de conocimiento propuesto en esta tesis. Se concluye que es conveniente utilizar un número variable de entradas al sistema, dependiente de la consulta en Lenguaje Natural realizada por el usuario, usar umbrales de certeza modificables, y un motor borroso compuesto por conjuntos borrosos triangulares, difusor singleton, y concresor por Centro de Gravedad.

5. En cuanto al novedoso sistema de asignación de pesos, aunque los resultados obtenidos con el método TF-IDF son bastante razonables, el método basado en FL resulta claramente superior. Sobre todo, cuando las consultas de usuario no son exactas a la consulta tipo o las consultas son imprecisas, se puede observar que los resultados empeoran más para el método TF-IDF que para el método basado en FL, lo que nos da una idea de lo acertado de utilizar la lógica borrosa para añadir una mayor flexibilidad al sistema.

7.3. Futuras líneas de trabajo.

Tras los estudios realizados para el desarrollo de esta tesis, se proponen una serie de futuras líneas de investigación, a saber:

1. El estudio de la gestión del interfaz de usuario de Agentes Inteligentes creados con Lógica Borrosa, mediante la mejora de las conversaciones en Lenguaje Natural con el usuario, sobre todo en lo referente a la creación de *logs* que permitan al Agente Inteligente tener una mayor información sobre el usuario con el fin de responder mejor a sus necesidades y considerar estos *logs* en sus respuestas. Así mismo, se podrían considerar otros aspectos, como el aspecto emocional del Agente Inteligente o su interfaz gráfica.

2. La posibilidad de utilizar el enfoque basado en ontologías en lugar del enfoque vectorial. El hecho de que consideremos la dificultad de utilizar el primero de ellos no quiere decir que no consideremos interesante la posibilidad de usar ontologías, lo que constituye un campo de investigación muy atractivo.

3. La integración de los Agentes Inteligentes en ámbitos distintos de los portales web y en los que la información tenga características similares en cuanto a su abundancia y heterogeneidad. En concreto, el grupo de investigación TIC150 de la Universidad de Sevilla, al cual pertenece el autor de esta tesis está trabajando en la actualidad en utilización del Agente Inteligente como profesor virtual que permita la formación online desde cualquier punto del planeta dando la sensación de que se está asistiendo a una clase ordinaria gracias al mundo virtual creado en Second Life.

4. La aplicación de otras técnicas de Inteligencia Computacional distintas de la Lógica Borrosa a la construcción de Agentes Inteligentes. Entre estas técnicas, consideramos

que las denominadas técnicas neuro-borrosas (*neuro-fuzzy*) representan una posibilidad muy interesante, puesto que combinan el razonamiento humano que aporta la FL con la estructura basada en conexiones neuronales de las ANN, aprovechando las ventajas de ambas.

Anexo A. El programa Un-fuzzy.

El programa Un-Fuzzy, actualmente en la versión 1.2, es una herramienta para el análisis, diseño, simulación e implementación de Sistemas de Lógica Difusa creado por Oscar G. Duarte para el departamento de Tecnología Eléctrica de la Universidad de Colombia, distribuyéndose gratuitamente a través de la Web de dicho departamento [LARIOS04].

Es destacable de este programa la poca cantidad de recursos que necesita para funcionar, sin necesitar ni siquiera una instalación previa, y que tanto el programa como el manual y la ayuda en línea se encuentran en español, algo no muy común en este tipo de herramientas.

Las interfaces más importantes con las que cuenta Un-fuzzy se detallan en los siguientes apartados.

A.1. Ventana principal.

En la Figura A.1, se muestra la ventana que aparece nada más arrancar la utilidad y desde la que podemos acceder a todas las opciones en el diseño, simulación e implementación de problemas en lógica difusa que proporciona el programa.

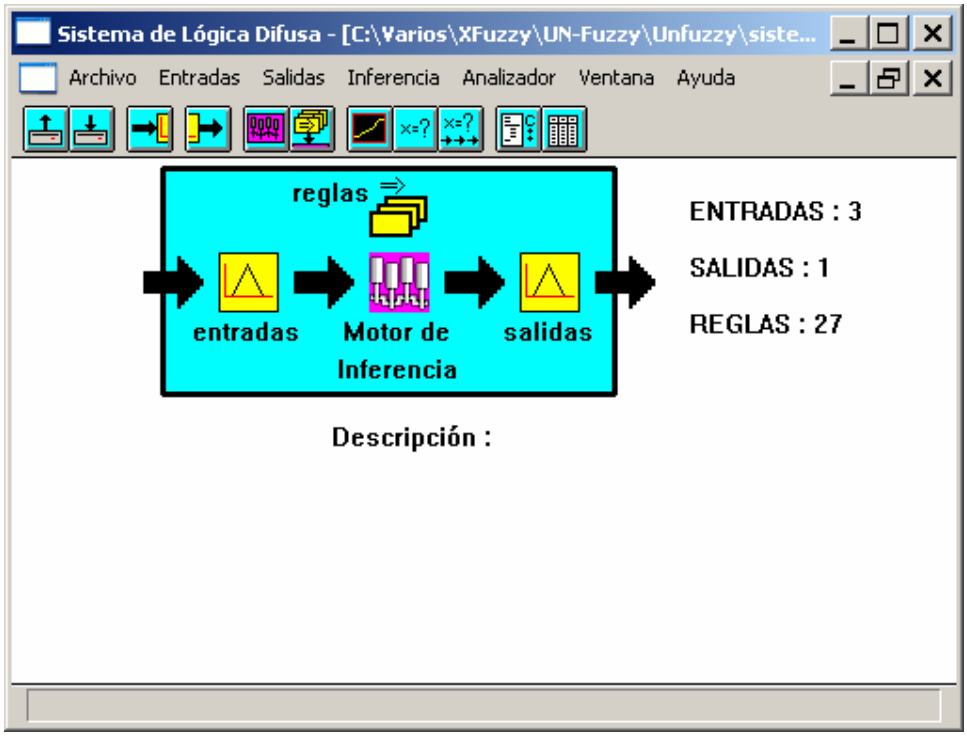


Figura A.1. Ventana principal de Un-fuzzy.

Como puede apreciarse en la Figura A.1, el programa tiene la estructura típica de cualquier programa en Windows, donde cabe destacar la barra de botones que resulta cómoda e intuitiva a la hora de utilizar el programa. El significado de estos botones aparece en la Figura A.2.

Figura A.2. Barra de botones de Un-Fuzzy.

En esta ventana se hace una representación gráfica de una máquina de inferencia en la que aparecen una serie de bloques que representan al universo de entrada, al de salida, a la base de reglas y al motor de inferencia. Si quisiéramos modificar cualquiera de estas partes, solo tendríamos que hacer doble clic sobre el bloque que lo representa.

Este programa ofrece la posibilidad de abrir un archivo, crear uno nuevo ó guardar nuestros proyectos. Todas estas opciones están accesibles desde el menú Archivo. Sólo tenemos la posibilidad de guardar a disco nuestro trabajo si tenemos definido el universo de entrada, el de salida y al menos una regla. Esto puede suponer un grave problema para aplicaciones en las que haya muchas variables de entrada y salida ya que obliga a realizar todos estos pasos en una sola sesión.

A.2. Universo de entrada.

Para la definición del Universo de entrada es escogen las variables lingüísticas del sistema, sus correspondientes conjuntos y los difusores correspondientes.

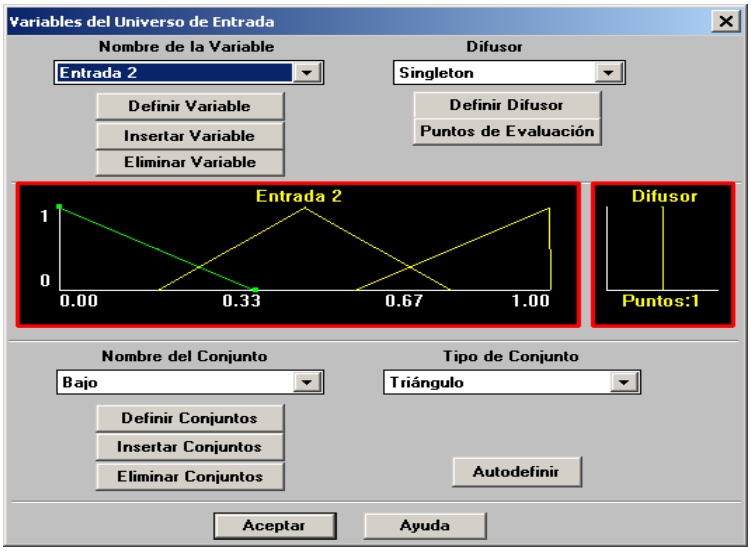


Figura. A.3. Cuadro de diálogo Definir variable de entrada.

Desde el cuadro de diálogo mostrado en la Figura A.3, se tiene acceso a las opciones para eliminar, crear ó modificar las variables lingüísticas del universo de entrada.

En el caso de definir una nueva o modificar una existente, aparece una ventana donde podemos escoger el nombre, el universo de discurso de la variable (pudiendo tomar tanto valores positivos como negativos) y el número de puntos de evaluación para la generación de las tablas.

Este cuadro de diálogo también permite crear, modificar ó eliminar los distintos valores lingüísticos de cada una de las variables, así como escoger el tipo, es decir, la forma de su función de pertenencia correspondencia.

Las funciones de pertenencia para los conjuntos difusos que podemos utilizar en este programa son:

- Tipo L.
- Triángulo.
- Tipo Pi.
- Tipo Gamma.
- Tipo Z.
- Campana.
- Tipo S.
- Pi.Campana.
- Singleton.

Por último, este cuadro de diálogo nos da la opción de definir el difusor, el número de puntos de evaluación y su intervalo de discurso. Los distintos tipos de difusores que podemos escoger son:

- Singleton.
- Triángulo.
- Campana.
- Tipo Pi.
- Pi-Campana.

A.3. Universo de salida.

En la generación del universo de salida nos encontramos con las mismas opciones que las comentadas para el universo de entrada, tal y como se puede ver en la Figura A.4

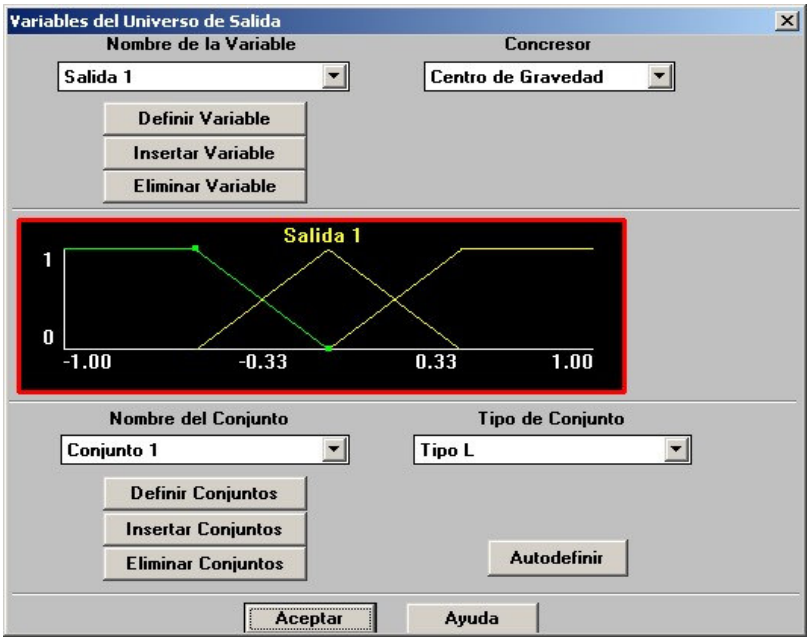


Figura A.4. Cuadro de diálogo Variables del Universo de Salida.

Los concresores que podemos utilizar con este programa son:

- Primer máximo.
- Último máximo.
- Media de Máximos.
- Centro de gravedad.

A.4. Base de reglas.

Como en todo sistema de lógica difusa, esta aplicación necesita de una serie de reglas en las que almacena el conocimiento para poder funcionar.

Hay varias opciones para definir estas reglas:

- Asistente de Definición Rápida de Reglas.

Figura A.5. Cuadro de diálogo Definición Rápida de la Base de Reglas.

Aunque existen varias formas de definir el conjunto de reglas, la más útil es la definición de un conjunto completo de reglas. Se crean tantas reglas como sea necesaria para cubrir el conjunto completo de posibilidades de combinación entre todos los conjuntos difusos de las variables lingüísticas de entrada. Este proceso se ilustra en las Figuras A.5 y A.6 .

Definir una nueva regla con este programa puede resultar algo tedioso, siendo más cómodo en muchas ocasiones crearlas todas, para ir modificando o eliminando posteriormente aquellas que sean necesarias. No obstante, la definición de reglas es el mayor inconveniente de la aplicación, por razones que se explicarán más adelante.

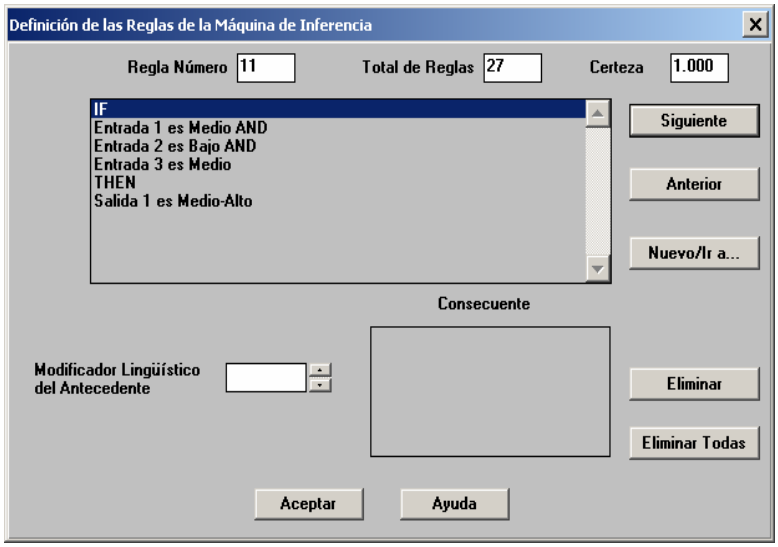


Figura A.6 Cuadro de diálogo Definición de las Reglas de la Máquina de Inferencia.

En la parte superior de la Figura A.6, se observan dos recuadros que indican el número total de reglas que tenemos definidas y cuál es la regla actual. Este cuadro de diálogo nos permite seleccionar el consecuente de una regla determinada.

La base de reglas de esta utilidad está muy limitada ya que sólo permite la definición de reglas AND y no de reglas OR. Esto motiva que haya que considerar todas las posibilidades que da una OR basándose en el operador AND con el consiguiente aumento de reglas. Este inconveniente se acentúa a medida que aumenta el número de variables y conjuntos de entrada.

A.5. Simulación.

El programa Un-Fuzzy permite dos tipos de simulaciones diferentes:

- Calcular Salidas (Figura A.7).

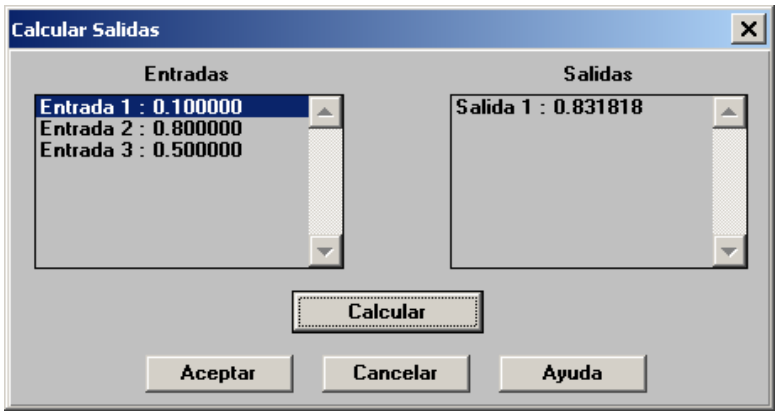


Figura A.7. Cuadro de diálogo Calcular Salidas.

Desde el cuadro de diálogo mostrado en la Figura A.7, se pueden asignar valores concretos a las entradas para observar como evolucionan las salidas. Esta opción resulta muy útil para poder comprobar si el funcionamiento de un sistema es el correcto.

- Análisis paso a paso (Figura A.8).

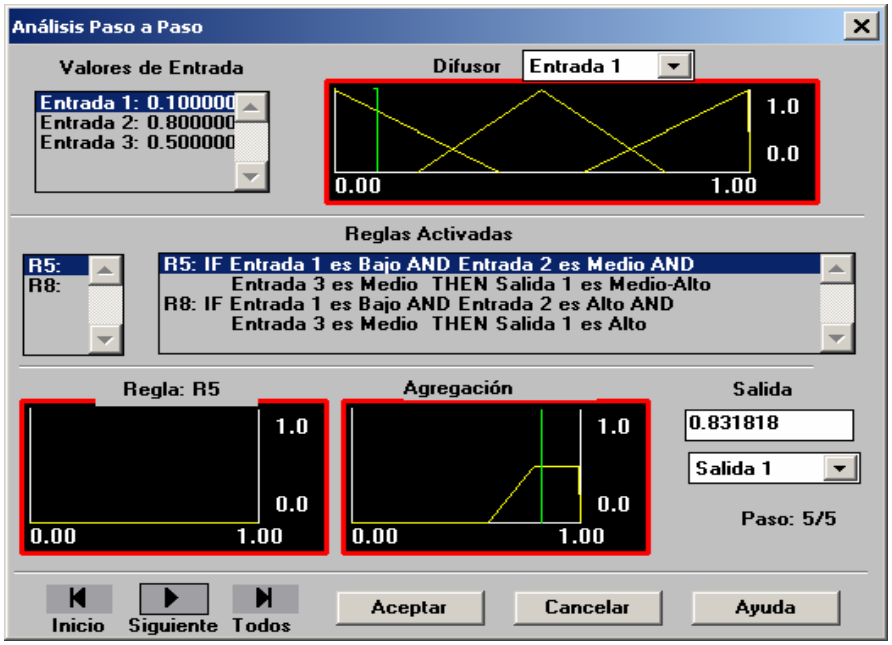


Figura A.8. Cuadro de diálogo Análisis Paso a Paso.

Esta opción permite calcular cuáles son los valores concretos de las variables de salida para ciertos valores de las variables de entrada, mostrando los pasos intermedios que llevan a obtener tal resultado. Estos pasos intermedios nos muestran el difusor sobre la variable de entrada, las reglas que se activan, la función de pertenencia de esas reglas, la intersección de todas esas funciones de pertenencia y, por último, el resultado del concresor.

A.6. Generación de código.

La principal ventaja de Un-fuzzy, y que nos hace decantarnos por la opción de usar este programa en lugar de por el potente toolbox de lógica borrosa, es la opción de generar automáticamente el código fuente del sistema de lógica borrosa en lenguaje C. El cuadro de diálogo correspondiente a esta opción se muestra en la Figura A.9.

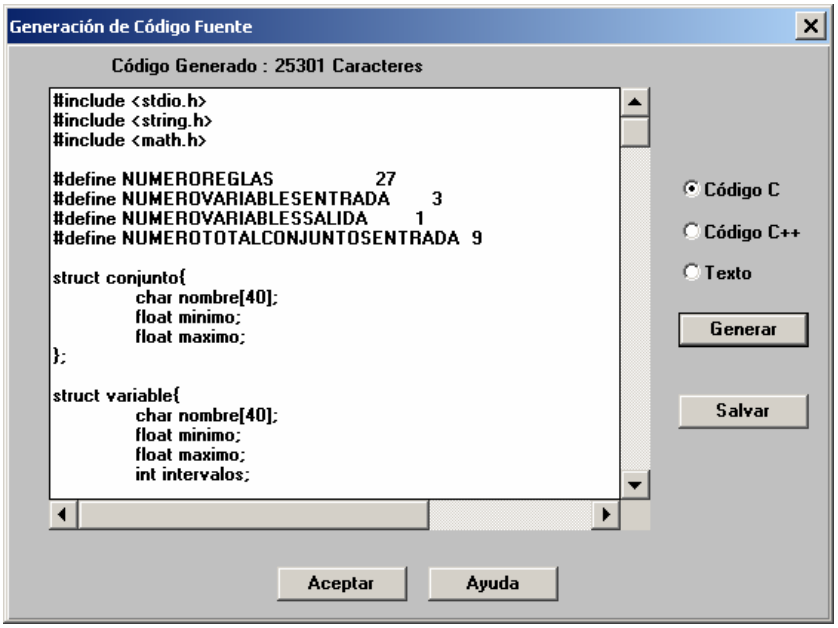


Figura A.9. Cuadro de diálogo Generación de Código Fuente.

En esta ventana podemos escoger la generación de código fuente en C o en C++, teniendo la opción de guardarlo en el disco. El código generado es, además un código relativamente sencillo y de fácil comprensión y depuración.

Anexo B. Resultados obtenidos con los dos métodos de TW.

En el Anexo B se muestran los resultados obtenidos con los métodos basados en FL y TF-IDF clásico, respectivamente, de forma exhaustiva.

B.1. Resultados obtenidos con el método basado en FL

En este apartado se recogen los resultados obtenidos mediante el método basado en FL de forma exhaustiva. Se muestran los distintos Objetos, con sus correspondientes preguntas tipo y las distintas opciones devueltas por el sistema (con el Objeto deseado en letra negrita).

Nº Pregunta	Pregunta	1ª opcion	2ª opcion	3ª opcion	4ª opcion	5ª opción	
3.1.1	**Como puedo realizar la matricula por internet en alguna titulacion impartida por la Univ. de Sev.?**	12.3.1 (45.45%)	**3.1.1 (39.12%)**	3.1.2 (39.12%)	3.1.3 (39.12%)	3.1.7 (39.12%)	(umbrales bajados a 0.3)
	Como puedo realizar la automatricula en alguna titulacion impartida por la Univ. de Sev.?	**3.1.1 (39.12%)**	3.1.2 (39.12%)	3.1.3 (39.12%)	3.1.7 (39.12%)	(umbrales bajados a 0.3)	
3.1.2	**Como puedo realizar la matricula por internet en alguna titulacion de segundo ciclo**	**3.1.2 (65.82%)**	3.3.2 (56.77%)	4.1.2 (54.53%)	4.1.1 (45.45%)		
	Como puedo realizar la matricula por internet en alguna titulacion de primer ciclo	**3.1.2 (65.82%)**	3.3.2 (56.77%)				
	Como puedo realizar la automatricula en alguna titulacion de primer ciclo	**3.1.2 (65.82%)**	3.3.2 (56.77%)				
	Como puedo realizar la automatricula en alguna titulacion de segundo ciclo	**3.1.2 (55.19%)**	3.3.2 (56.77%)	4.1.2 (54.53%)	4.1.1 (45.45%)		
3.1.3	**Cuales son las normas para la automatricula**	**3.1.3 (56.53%)**					
	Cuales son las instrucciones para la automatricula	**3.1.3 (56.53%)**					
	Cual es la documentacion necesaria para la automatricula	**3.1.3 (56.53%)**					
	Cuales son las normas para la matricula por internet	**3.1.3 (56.53%)**	3.3.4 (56.53%)	10.4.15 (41.04%)			
	Cuales son las instrucciones para la matricula por internet	**3.1.3 (56.53%)**					

	Cual es la documentacion necesaria para la matricula por internet	**3.1.3 (56.53%)**	10.6.1 (41.04%)				
3.1.4	**Necesito el modelo de solicitud para las ayudas de accion social**	**3.1.4 (86.73%)**	10.6.2 (60.86%)	3.8.3 (57.03%)	1.5.6 (56.77%)	1.5.1 (54.06%)	
3.1.5	**Me gustaria conocer el funcionamiento de la Secretaria Virtual de la Univ. de Sev.**	**3.1.5 (74.13%)**	12.1.1 (60.34%)	12.1.5 (52.76%)	12.1.6 (52.76%)		
3.1.6	**Que es un Punto de info. Universitaria**	**3.1.6 (53.10%)**	1.5.8 (45.46%)				
	Que es un PIU	**3.1.6 (61.36%)**					
3.1.7	**Como puedo realizar la matricula por internet en el Instituto de Idiomas**	**3.1.7 (66.53%)**	2.3.1 (54.53%)	1.5.6 (54.06%)			
	Como puedo realizar la automatricula en el Instituto de Idiomas	**3.1.7 (66.53%)**	2.3.1 (54.53%)	1.5.6 (54.06%)			
3.2.1	**Que y como se estudia en las Universidades de Andalucia**	**3.2.1 (66.57%)**	3.2.2 (54.53%)				
	Que estudios existen en las Universidades de Andalucia	**3.2.1 (66.57%)**	3.2.2 (54.53%)				
3.2.2	**Me gustaria encontrar info. sobre el acceso a las Universidades de Andalucia**	**3.2.2 (66.57%)**	3.2.1 (54.53%)	3.2.5 (41.04%)	3.2.6 (41.04%)		
	Me gustaria encontrar info. sobre el acceso a las Universidades andaluzas	**3.2.2 (66.57%)**	3.2.5 (41.04%)	3.2.6 (41.04%)			
3.2.3	**Deseo info. acerca del Distrito unico Andaluz**	**3.2.3 (69.02%)**	12.4.2 (50.00%)	3.2.4 (45.45%)			
3.2.4	**Deseo info. acerca del Distrito Abierto**	**3.2.4 (60.86%)**	3.2.3 (41.04%)				
3.2.5	**Me gustaria resolver mis dudas acerca del acceso a la Univ. de Sev.**	**3.2.5 (41.04%)**	3.2.2 (41.04%)	3.2.6 (41.04%)			
3.2.6	**Me gustaria contactar con el Secretariado de Acceso a la Univ. de Sev.**	**3.2.6 (55.69%)**	1.5.1 (54.06%)	1.5.6 (46.95%)	1.1.3 (45.45%)	3.2.2 (41.04%)	
3.2.7	**Como puedo realizar la matricula por internet en un Master oficial**	**3.2.7 (74.19%)**	4.1.2 (61.85%)	4.2.1 (61.36%)			
	Como puedo realizar la matricula por internet en los Masters oficiales de la Univ. de Sev.?	**3.2.7 (74.19%)**	4.1.2 (61.85%)	4.2.1 (61.36%)			
3.3.1	**Que titulaciones puedo estudiar en la Univ. de Sev.?**	**3.3.1 (44.31%)**	3.3.2 (43.75%)				
	Que carreras puedo estudiar en la Univ. de Sev.?	**3.3.1 (58.91%)**	3.3.2 (48.96%)				
	Que estudios puedo realizar en la Univ. de Sev.?	**3.3.1 (45.45%)**	3.2.1 (39.65%)	1.5.1 (38.57%)	(umbrales bajados a 0.3)		
3.3.2	**Donde puedo encontrar una lista de las titulaciones impartidas en la Univ. de Sev.?**	**3.3.2 (62.09%)**	3.3.1 (44.31%)				
	Donde puedo encontrar una lista de las titulaciones de primer ciclo impartidas en la Univ. de Sev.?	**3.3.2 (70.80%)**	3.1.2 (55.20%)	3.3.1 (44.31%)			
	Donde puedo encontrar una lista de las titulaciones de segundo ciclo impartidas en la Univ. de Sev.?	**3.3.2 (70.80%)**	3.1.2 (55.20%)	4.1.2 (54.53%)	4.1.1 (45.45%)	3.3.1 (44.31%)	
	Donde puedo encontrar una relacion de las titulaciones impartidas en la Univ. de Sev.?	**3.3.2 (62.09%)**	3.3.1 (44.31%)				
	Donde puedo encontrar una relacion de las titulaciones de primer ciclo impartidas en la Univ. de Sev.?	**3.3.2 (70.80%)**	3.1.2 (55.20%)	3.3.1 (44.31%)			
	Donde puedo encontrar una lista de las carreras de segundo ciclo impartidas en la Univ. de Sev.?	**3.3.2 (70.80%)**	3.3.1 (58.91%)	3.1.2 (55.20%)	4.1.2 (54.53%)	4.1.1 (45.45%)	

	Donde puedo encontrar una lista de los estudios de segundo ciclo disponibles en la Univ. de Sev.?	**3.3.2 (69.86%)**	3.1.2 (55.20%)	4.1.2 (54.53%)	1.5.1 (54.06%)	3.2.1 (45.45%)	
3.3.3	**Deseo info. acerca del Plan de Organizacion Docente**	11.4.2 (57.67%)	**3.3.3 (53.08%)**	11.1.1 (52.59%)	1.5.1 (45.47%)	11.1.4 (45.47%)	
	Deseo info. acerca del POD	**3.3.3 (61.36%)**	11.1.1 (58.91%)	11.1.4 (48.38%)			
3.3.4	**Cuales son las normas para realizar la matricula**	**3.3.4 (56.53%)**	3.1.3 (56.53%)				
	Deseo conocer la normativa para realizar la matricula	**3.3.4 (56.53%)**					
3.3.5	**Como puedo ampliar la matricula**	**3.3.5 (56.53%)**					
	Como puedo realizar una ampliacion de matricula	**3.3.5 (56.53%)**					
3.3.6	**Como puedo anular la matricula**	**3.3.6 (57.33%)**					
	Como puedo pedir la anulacion de la matricula	**3.3.6 (57.33%)**					
3.3.7	**Cual es el regimen economico de la matricula**	**3.3.7 (56.51%)**					
3.3.8	**Quisiera conocer los planes de estudio de las titulaciones impartidas por la Univ. de Sev.**	**3.3.8 (69.02%)**	3.3.1 (44.31%)	3.3.2 (43.75%)	6.5.1 (41.04%)		
	Quisiera conocer el plan de estudio de una carrera impartida por la Univ. de Sev.	**3.3.8 (66.53%)**	6.5.1 (41.04%)				
3.4.1	**Me gustaria obtener info. acerca de una asignatura**	**3.4.1 (61.36%)**					
	Me gustaria obtener info. acerca de los programas de una asignatura	**3.4.1 (63.74%)**					
3.5.1	**Me gustaria obtener info. acerca del Consejo de Alumnos de la Univ. de Sev.**	**3.5.1 (56.51%)**	1.5.1 (52.76%)	1.5.6 (52.76%)	1.5.8 (52.76%)		
	Me gustaria obtener info. acerca del CADUS	**3.5.1 (61.36%)**					
3.6.1	**Desearia info. sobre Libre Configuracion**	**3.6.1 (60.34%)**	3.6.2 (60.05%)	3.6.3 (60.05%)			
3.6.2	**Me gustaria obtener info. sobre como realizar la matricula en Libre Configuracion**	**3.6.2 (86.51%)**	3.6.1 (60.34%)	3.6.3 (60.05%)			
3.6.3	**Que asignaturas puedo escoger para Libre Configuracion**	**3.6.3 (86.97%)**	3.6.1 (60.34%)	3.6.2 (60.05%)			
3.7.1	**Existe una Guia del Estudiante de la Univ. de Sev.?**	**3.7.1 (61.85%)**					
3.8.1	**Me gustaria obtener info. acerca de las becas y ayudas a las que se puede acceder en las titulaciones de la Univ. de Sev.**	**3.8.1 (64.76%)**	3.10.1 (60.86%)	3.8.2 (54.53%)	3.8.5 (54.53%)	3.8.4 (54.07%)	
	Me gustaria obtener info. acerca de las becas a las que se puede acceder en las titulaciones de la Univ. de Sev.	**3.8.1 (54.53%)**	3.3.2 (43.75%)	3.3.1 (44.31%)	3.10.1 (41.04%)	3.10.4 (41.04%)	
	Me gustaria obtener info. acerca de las ayudas a las que se puede acceder en las titulaciones de la Univ. de Sev.	**3.8.1 (54.53%)**	3.3.2 (43.75%)	3.3.1 (44.31%)			
	Me gustaria obtener una beca para estudiar en la Univ. de Sev.	**3.8.1 (45.45%)**	3.8.2 (45.45%)	9.2.1 (45.45%)			
	Me gustaria obtener una ayuda para estudiar en la Univ. de Sev.	**3.8.1 (41.04%)**	3.8.2 (41.04%)				
	Me gustaria obtener una subvencion para estudiar en la Univ. de Sev.	**3.8.1 (45.45%)**					

	Me gustaria obtener info. acerca de las becas y ayudas a las que se puede acceder en las carreras de la Univ. de Sev.	3.10.1 (60.86%)	3.3.1 (58.91%)	**3.8.1 (54.53%)**	3.8.2 (54.53%)	3.8.5 (54.53%)	
	Me gustaria obtener info. acerca de las becas a las que se puede acceder en las carreras de la Univ. de Sev.	3.3.1 (58.91%)	3.3.2 (48.96%)	3.10.1 (41.04%)	3.10.4 (41.04%)	3.10.5 (41.04%)	
3.8.2	**Que becas y ayudas propias ofrece la Univ. de Sev.?**	**3.8.2 (66.53%)**	3.10.1 (60.86%)	3.8.1 (54.53%)	3.8.5 (54.53%)	3.8.4 (54.07%)	
	Que becas propias ofrece la Univ. de Sev.?	**3.8.2 (56.51%)**	3.10.1 (41.04%)	3.10.4 (41.04%)	3.10.5 (41.04%)		
	Que ayudas propias ofrece la Univ. de Sev.?	**3.8.2 (56.51%)**					
	Como puedo conseguir una beca para estudiar la carrera	**3.8.2 (45.45%)**	3.8.1 (45.45%)	9.2.1 (45.45%)			
	Como puedo conseguir una ayuda para estudiar la carrera	**3.8.2 (41.04%)**	3.8.1 (41.04%)				
	Como puedo conseguir una subvencion para estudiar la carrera	3.8.1 (45.45%)					
3.8.3	**Deseo obtener el impreso de solicitud para la convocatoria de ayudas de colaboracion del SACU para Escuela Infantil**	**3.8.3 (66.54%)**	3.1.4 (62.09%)	10.1.1 (61.36%)	3.8.5 (53.41%)	3.8.4 (52.76%)	
	Deseo obtener el impreso de solicitud para la convocatoria de ayudas de colaboracion para la mediacion universitaria del instituto andaluz de la Juventud y la Univ. de Sev.	**3.8.3 (66.54%)**	3.1.4 (62.09%)	9.3.1 (59.15%)	1.5.1 (55.67%)	3.8.5 (53.41%)	
	Deseo obtener el impreso de solicitud para la convocatoria de ayudas del SADUS	**3.8.3 (68.40%)**	3.1.4 (62.09%)	8.3.2 (58.91%)	3.8.5 (53.41%)	3.8.4 (52.76%)	
	Deseo obtener el impreso de solicitud para la convocatoria de ayudas al Estudio de la Junta de Andalucia	**3.8.3 (66.48%)**	3.8.4 (62.26%)	3.1.4 (62.09%)	3.8.5 (53.41%)	12.4.5 (51.96%)	
	Deseo obtener el impreso de solicitud para la convocatoria extraordinaria de ayudas del SACU, modalidad ayuda de cooperacion al desarrollo	**3.8.3 (72.02%)**	3.1.4 (62.09%)	9.4.1 (61.03%)	3.8.1 (55.69%)	3.8.2 (55.69%)	
	Deseo obtener el impreso de solicitud para la convocatoria de ayudas del CDE	**3.8.3 (68.40%)**	3.1.4 (62.09%)	10.6.1 (61.36%)	3.8.5 (53.41%)	3.8.4 (52.76%)	
	Deseo obtener el impreso de solicitud para la convocatoria de ayudas para los Bonos del Comedor	**3.8.3 (71.58%)**	3.1.4 (62.09%)	3.8.5 (53.41%)	3.8.4 (52.76%)		
	Deseo obtener el impreso de solicitud para la convocatoria de ayudas para alumnos ganadores de las olimpiadas de fisica, matematicas y quimica	**3.8.3 (66.54%)**	3.1.4 (62.09%)	1.5.5 (56.77%)	2.1.2 (54.53%)	2.4.1 (54.53%)	
3.8.4	**Deseo obtener info. acerca de la convocatoria de becas y ayudas al estudio de la Junta de Andalucia**	**3.8.4 (66.90%)**	3.8.5 (63.16%)	3.10.1 (60.86%)	3.8.1 (54.53%)	3.8.2 (54.53%)	
	Deseo obtener info. acerca de la convocatoria de becas de la Junta de Andalucia	**3.8.4 (66.23%)**	3.8.5 (53.41%)	12.4.5 (51.96%)	3.10.1 (41.04%)	3.10.4 (41.04%)	
	Deseo obtener info. acerca de la convocatoria de ayudas al estudio de la Junta de Andalucia	**3.8.4 (62.26%)**	3.8.3 (54.06%)	3.8.5 (53.41%)	12.4.5 (51.96%)	3.1.4 (43.75%)	
	Deseo obtener info. acerca de la convocatoria de becas de la Junta	**3.8.4 (66.59%)**	3.8.5 (53.41%)	3.10.1 (41.04%)	3.10.4 (41.04%)	3.10.5 (41.04%)	

3.8.5	**Deseo obtener info. acerca de la convocatoria de becas y ayudas del Ministerio de Educacion y Ciencia**	**3.8.5 (63.18%)**	3.8.4 (62.60%)	3.10.1 (60.86%)	3.8.1 (54.53%)	3.8.2 (54.53%)	
	Deseo obtener info. acerca de la convocatoria de becas y ayudas del MEC	**3.8.5 (63.16%)**	3.8.4 (62.60%)	3.10.1 (60.86%)	3.8.1 (54.53%)	3.8.2 (54.53%)	
	Deseo obtener info. acerca de la convocatoria de becas del Ministerio de Educacion y Ciencia	**3.8.5 (61.69%)**	3.8.4 (54.06%)	2.4.1 (44.41%)	3.10.1 (41.04%)	3.10.4 (41.04%)	
	Deseo obtener info. acerca de la convocatoria de ayudas del MEC	**3.8.5 (64.79%)**	3.8.3 (54.06%)	3.8.4 (52.76%)	3.1.4 (43.75%)		
3.9.1	**Desearia obtener info. sobre el carnet de estudiante**	**3.9.1 (61.85%)**					
	Desearia obtener info. sobre el carnet universitario	**3.9.1 (48.38%)**	1.5.8 (46.02%)				
	Desearia obtener info. sobre el carne de estudiante	**3.9.1 (61.85%)**					
	Desearia obtener info. sobre la Tarjeta Inteligente Universitaria	**3.9.1 (60.86%)**	8.3.3 (45.45%)	1.5.8 (46.02%)			
	Desearia obtener info. sobre la TIU	**3.9.1 (61.36%)**					
3.10.1	**Me gustaria obtener info. acerca de los programas de becas y ayudas para la movilidad**	**3.10.1 (73.30%)**	3.8.1 (54.53%)	3.8.2 (54.53%)	3.8.5 (54.53%)	3.8.4 (54.07%)	
	Me gustaria obtener info. acerca de los programas de becas y ayudas para el intercambio de estudiantes	**3.10.1 (74.13%)**	3.8.1 (54.53%)	3.8.2 (54.53%)	3.8.5 (54.53%)	3.8.3 (54.07%)	
	Me gustaria obtener info. acerca de los programas de becas para la movilidad	**3.10.1 (73.30%)**	3.10.4 (41.04%)	3.10.5 (41.04%)			
	Me gustaria obtener info. acerca de los programas de ayudas para el intercambio de estudiantes	**3.10.1 (74.13%)**	3.8.3 (54.07%)	3.10.2 (41.04%)			
3.10.2	**Desearia info. acerca del programa SICUE**	**3.10.2 (48.78%)**					
	Desearia info. acerca del programa para el Sistema de Intercambio entre Centros Universitarios Españoles	**3.10.2 (46.66%)**	2.3.1 (45.45%)	3.10.1 (45.45%)			
3.10.3	**Desearia info. acerca del programa SENECA**	**3.10.3 (48.78%)**					
3.10.4	**Desearia info. acerca de las becas internacionales**	**3.10.4 (55.72%)**	3.10.5 (55.72%)	9.2.1 (45.45%)	3.10.1 (41.04%)		
	Desearia info. acerca del programa Erasmus	**3.10.4 (48.78%)**	3.10.5 (46.66%)				
3.10.5	**Desearia info. acerca de las becas internacionales**	**3.10.5 (55.72%)**	3.10.4 (55.72%)	9.2.1 (45.45%)	3.10.1 (41.04%)		
	Desearia info. acerca del programa Erasmus Mundus	**3.10.5 (62.09%)**	3.10.4 (48.78%)				
3.11.1	**Me gustaria acceder al Espacio Europeo de Educacion Superior**	1.12.4 (60.05%)	**3.11.1 (58.04%)**				
	Me gustaria acceder al EEES	**3.11.1 (61.36%)**					
	Tengo una duda acerca de los Creditos ECTS	**3.11.1 (56.51%)**					

Tabla B.1. Resultados detallados para el Tema 3 con el método basado en FL.

B.2. Resultados obtenidos con el método TF-IDF.

Al igual que en el apartado B.1 se mostraban los resultados obtenidos con el método basado en FL, en el apartado B.2 se muestran los resultados obtenidos mediante el método TF-IDF.

Nº Pregunta	**Pregunta**	**1ª opcion**	**2ª opcion**	**3ªopcion**	**4ª opcion**	**5 ª opcion**	
3.1.1	**Como puedo realizar la matricula por internet en alguna titulacion impartida por la Univ. de Sev.?**	3.6.2 (86.97%)	**3.1.1 (51.66%)**	12.3.1 (51.66%)	12.4.6 (44.16%)	3.2.7 (39.72%)	
	Como puedo realizar la automatricula en alguna titulacion impartida por la Univ. de Sev.?	**3.1.1 (51.66%)**	3.1.2 (34.72%)	3.1.7 (32.51%)	3.1.3 (30.33%)		
3.1.2	**Como puedo realizar la matricula por internet en alguna titulacion de segundo ciclo**	3.6.2 (86.97%)	**3.1.2 (64.77%)**	3.3.2 (54.77%)	3.1.1 (51.66%)	12.3.1 (51.66%)	
	Como puedo realizar la matricula por internet en alguna titulacion de primer ciclo	3.6.2 (86.97%)	**3.1.2 (64.77%)**	3.3.2 (54.77%)	3.1.1 (51.66%)	12.3.1 (51.66%)	
	Como puedo realizar la automatricula en alguna titulacion de primer ciclo	**3.1.2 (64.77%)**	3.3.2 (54.77%)	3.1.1 (51.66%)	4.1.1 (35.63%)	4.1.2 (33.62%)	
	Como puedo realizar la automatricula en alguna titulacion de segundo ciclo	**3.1.2 (64.77%)**	3.3.2 (54.77%)	3.1.1 (51.66%)	4.1.2 (48.35%)	4.1.1 (35.63%)	
3.1.3	**Cuales son las normas para la automatricula**	3.1.1 (51.66%)	3.3.4 (50.56%)	**3.1.3 (47.16%)**	11.2.3 (36.80%)	3.1.2 (34.72%)	
	Cuales son las instrucciones para la automatricula	3.1.1 (51.66%)	**3.1.3 (47.16%)**	3.1.2 (34.72%)	3.1.7 (32.51%)		
	Cual es la documentacion necesaria para la automatricula	3.1.1 (51.66%)	**3.1.3 (47.16%)**	6.2.1 (39.25%)	3.1.2 (34.72%)	3.1.7 (32.51%)	
	Cuales son las normas para la matricula por internet	3.6.2 (86.97%)	3.3.4 (52.42%)	3.1.1 (51.66%)	12.3.1 (51.66%)	**3.1.3 (47.16%)**	
	Cuales son las instrucciones para la matricula por internet	3.6.2 (86.97%)	3.1.1 (51.66%)	12.3.1 (51.66%)	**3.1.3 (47.16%)**	12.4.6 (44.16%)	
	Cual es la documentacion necesaria para la matricula por internet	3.6.2 (86.97%)	3.1.1 (51.66%)	12.3.1 (51.66%)	**3.1.3 (47.16%)**	12.4.6 (44.16%)	
3.1.4	**Necesito el modelo de solicitud para las ayudas de accion social**	8.3.4 (85.20%)	**3.1.4 (73.03%)**	10.6.2 (61.70%)	1.9.6 (61.54%)	3.10.1 (42.11%)	
3.1.5	**Me gustaria conocer el funcionamiento de la Secretaria Virtual de la Univ. de Sev.**	**3.1.5 (86.68%)**	11.4.3 (63.45%)	11.1.5 (49.42%)	12.1.6 (48.39%)	10.4.34 (45.94%)	
3.1.6	**Que es un Punto de info. Universitaria**	**3.1.6 (59.95%)**	1.5.8 (38.93%)				
	Que es un PIU	**3.1.6 (40.00%)**	1.5.8 (34.52%)	(umbral1 bajado a 0.15)			
3.1.7	**Como puedo realizar la matricula por internet en el Instituto de Idiomas**	3.6.2 (86.97%)	**3.1.7 (63.13%)**	3.1.1 (51.66%)	12.3.1 (51.66%)	2.3.1 (50.30%)	
	Como puedo realizar la automatricula en el Instituto de Idiomas	**3.1.7 (63.13%)**	3.1.1 (51.66%)	2.3.1 (50.30%)	2.5.1 (35.30%)	3.1.3 (34.72%)	
3.2.1	**Que y como se estudia en**	**3.2.1**	3.2.2	7.3.3	3.8.4	9.2.1	

	las Universidades de Andalucia	**(64.61%)**	(57.50%)	(51.66%)	(43.52%)	(36.40%)	
	Que estudios existen en las Universidades de Andalucia	**3.2.1 (64.61%)**	3.2.2 (57.50%)	7.3.3 (51.66%)	3.3.1 (47.33%)	3.8.4 (43.52%)	
3.2.2	**Me gustaria encontrar info. sobre el acceso a las Universidades de Andalucia**	**3.2.2 (64.30%)**	3.2.1 (53.55%)	7.3.3 (51.66%)	3.8.4 (43.52%)	3.2.6 (43.27%)	
	Me gustaria encontrar info. sobre el acceso a las Universidades andaluzas	**3.2.2 (64.43%)**	3.2.6 (43.27%)	1.11.5 (39.93%)	10.4.29 (39.85%)	11.4.1 (39.25%)	
3.2.3	**Deseo info. acerca del Distrito unico Andaluz**	**3.2.3 (74.62%)**	3.2.4 (42.21%)	2.5.1 (35.20%)			
3.2.4	**Deseo info. acerca del Distrito Abierto**	**3.2.4 (69.67%)**	3.2.3 (39.35%)				
3.2.5	**Me gustaria resolver mis dudas acerca del acceso a la Univ. de Sev.**	3.2.6 (43.27%)	1.11.5 (39.93%)	10.4.29 (39.85%)	11.4.1 (39.25%)	12.1.2 (38.53%)	**3.2.5 está en el 9º puesto con 36.62%)**
3.2.6	**Me gustaria contactar con el Secretariado de Acceso a la Univ. de Sev.**	1.1.3 (87.00%)	**3.2.6 (68.87%)**	8.5.2 (51.66%)	8.2.2 (40.00%)	1.11.5 (39.93%)	
3.2.7	**Como puedo realizar la matricula por internet en un Master oficial**	3.6.2 (86.97%)	**3.2.7 (72.96%)**	4.1.2 (56.57%)	3.1.1 (51.66%)	12.3.1 (51.66%)	
	Como puedo realizar la matricula por internet en los Masters oficiales de la Univ. de Sev.?	3.6.2 (86.97%)	**3.2.7 (72.96%)**	4.1.2 (56.57%)	3.1.1 (51.66%)	12.3.1 (51.66%)	
3.3.1	**Que titulaciones puedo estudiar en la Univ. de Sev.?**	3.8.1 (50.66%)	**3.3.1 (39.73%)**	3.3.8 (34.85%)	3.3.2 (32.20%)		
	Que carreras puedo estudiar en la Univ. de Sev.?	**3.3.1 (47.33%)**	3.3.2 (35.74%)				
	Que estudios puedo realizar en la Univ. de Sev.?	**3.3.1 (47.33%)**	3.2.1 (45.12%)	3.3.2 (35.74%)	2.2.1 (34.97%)	2.1.2 (31.71%)	
3.3.2	**Donde puedo encontrar una lista de las titulaciones impartidas en la Univ. de Sev.?**	3.8.1 (50.66%)	**3.3.2 (46.86%)**	10.4.27 (45.62%)	3.3.1 (39.73%)	10.4.4 (38.46%)	
	Donde puedo encontrar una lista de las titulaciones de primer ciclo impartidas en la Univ. de Sev.?	3.1.2 (60.18%)	**3.3.2 (57.95%)**	3.8.1 (50.66%)	10.4.27 (45.62%)	3.3.1 (39.73%)	
	Donde puedo encontrar una lista de las titulaciones de segundo ciclo impartidas en la Univ. de Sev.?	3.1.2 (60.18%)	**3.3.2 (57.95%)**	3.8.1 (50.66%)	3.2.5 (49.85%)	4.1.2 (48.35%)	
	Donde puedo encontrar una relacion de las titulaciones impartidas en la Univ. de Sev.?	3.8.1 (50.66%)	**3.3.2 (46.86%)**	3.3.1 (39.73%)	3.3.8 (34.85%)		
	Donde puedo encontrar una relacion de las titulaciones de primer ciclo impartidas en la Univ. de Sev.?	3.1.2 (60.18%)	**3.3.2 (57.95%)**	3.8.1 (50.66%)	3.3.1 (39.73%)	4.1.1 (35.63%)	
	Donde puedo encontrar una lista de las carreras de segundo ciclo impartidas en la Univ. de Sev.?	**3.3.2 (61.78%)**	3.1.2 (60.18%)	4.1.2 (48.35%)	3.3.1 (47.33%)	10.4.4 (38.46%)	
	Donde puedo encontrar una lista de los estudios de segundo ciclo disponibles en la Univ. de Sev.?	**3.3.2 (61.78%)**	3.1.2 (60.18%)	4.1.2 (48.35%)	3.3.1 (47.33%)	10.4.27 (45.62%)	
3.3.3	**Deseo info. acerca del Plan de Organizacion Docente**	**3.3.3 (54.28%)**	1.10.1 (51.66%)	5.1.4 (51.21%)	11.1.4 (47.83%)	11.1.1 (44.89%)	
	Deseo info. acerca del POD	**3.3.3 (42.04%)**	11.1.4 (39.06%)	11.1.1 (37.51%)			

3.3.4	**Cuales son las normas para realizar la matricula**	3.6.2 (86.97%)	**3.3.4 (52.42%)**	3.1.1 (51.66%)	3.1.3 (47.16%)	10.4.15 (39.72%)	
	Deseo conocer la normativa para realizar la matricula	3.6.2 (86.97%)	4.1.3 (80.60%)	**3.3.4 (52.42%)**	3.1.1 (51.66%)	3.2.7 (39.72%)	
3.3.5	**Como puedo ampliar la matricula**	3.6.2 (86.97%)	**3.3.5 (52.42%)**	3.1.1 (51.66%)	3.2.7 (39.72%)	3.1.2 (34.72%)	
	Como puedo realizar una ampliacion de matricula	3.6.2 (86.97%)	**3.3.5 (52.42%)**	3.1.1 (51.66%)	3.2.7 (39.72%)	3.1.2 (34.72%)	
3.3.6	**Como puedo anular la matricula**	3.6.2 (86.97%)	**3.3.6 (52.42%)**	3.1.1 (51.66%)	3.2.7 (39.72%)	3.1.2 (34.72%)	
	Como puedo pedir la anulacion de la matricula	3.6.2 (86.97%)	**3.3.6 (52.42%)**	3.1.1 (51.66%)	3.2.7 (39.72%)	3.1.2 (34.72%)	
3.3.7	**Cual es el regimen economico de la matricula**	3.6.2 (86.97%)	**3.3.7 (64.27%)**	3.1.1 (51.66%)	3.2.7 (39.72%)	3.1.2 (34.72%)	
3.3.8	**Quisiera conocer los planes de estudio de las titulaciones impartidas por la Univ. de Sev.**	**3.3.8 (64.89%)**	3.8.1 (50.66%)	3.8.4 (43.52%)	3.3.1 (39.73%)	6.5.1 (39.25%)	
	Quisiera conocer el plan de estudio de una carrera impartida por la Univ. de Sev.	**3.3.8 (70.16%)**	1.10.1 (51.66%)	5.1.4 (51.21%)	3.8.4 (43.52%)	11.4.2 (40.00%)	
3.4.1	**Me gustaria obtener info. acerca de una asignatura**	**3.4.1 (51.66%)**					
	Me gustaria obtener info. acerca de los programas de una asignatura	**3.4.1 (63.45%)**	4.1.4 (51.21%)	3.10.1 (42.11%)	10.4.16 (36.63%)	10.4.43 (35.00%)	
3.5.1	**Me gustaria obtener info. acerca del Consejo de Alumnos de la Univ. de Sev.**	**3.5.1 (60.54%)**	1.9.4 (50.38%)	1.11.7 (44.16%)	1.9.6 (41.23%)	1.5.8 (34.99%)	
	Me gustaria obtener info. acerca del CADUS	**3.5.1 (44.16%)**					
3.6.1	**Desearia info. sobre Libre Configuracion**	**3.6.1 (63.45%)**	10.4.5 (39.43%)				
3.6.2	**Me gustaria obtener info. sobre como realizar la matricula en Libre Configuracion**	**3.6.2 (86.97%)**	3.6.1 (63.45%)	3.1.1 (51.66%)	3.2.7 (39.72%)	10.4.5 (39.43%)	
3.6.3	**Que asignaturas puedo escoger para Libre Configuracion**	**3.6.3 (86.97%)**	3.6.1 (63.45%)	1.5.2 (40.00%)	10.4.5 (39.43%)		
3.7.1	**Existe una Guia del Estudiante de la Univ. de Sev.?**	**3.7.1 (63.54%)**	1.5.4 (44.14%)	3.9.1 (39.25%)			
3.8.1	**Me gustaria obtener info. acerca de las becas y ayudas a las que se puede acceder en las titulaciones de la Univ. de Sev.**	8.3.4 (85.20%)	**3.8.1 (50.66%)**	3.10.1 (47.29%)			
	Me gustaria obtener info. acerca de las becas a las que se puede acceder en las titulaciones de la Univ. de Sev.	**3.8.1 (50.66%)**	3.3.1 (39.73%)	3.10.4 (38.45%)	9.2.1 (36.40%)	3.3.8 (34.85%)	
	Me gustaria obtener info. acerca de las ayudas a las que se puede acceder en las titulaciones de la Univ. de Sev.	8.3.4 (85.20%)	**3.8.1 (50.66%)**	3.10.1 (42.11%)	3.3.1 (39.73%)	3.1.4 (39.72%)	
	Me gustaria obtener una beca para estudiar en la Univ. de Sev.	**3.8.1 (39.02%)**	3.8.2 (39.02%)	9.2.1 (36.40%)			
	Me gustaria obtener una ayuda para estudiar en la Univ. de Sev.	9.2.1 (36.40%)	**3.8.1 (33.71%)**	3.8.2 (33.71%)			
	Me gustaria obtener una subvencion para estudiar en la Univ. de Sev.	**3.8.1 (39.02%)**	3.8.2 (39.02%)				
	Me gustaria obtener info. acerca de las becas y ayudas a las que se puede acceder en las carreras de la Univ. de Sev.	8.3.4 (85.20%)	3.3.1 (47.33%)	3.10.1 (47.29%)			
	Me gustaria obtener info. acerca de las becas a las que se puede acceder en	3.2.5 (62.50%)	9.2.1 (47.61%)	3.3.1 (47.33%)	3.1.6 (40.00%)	3.10.4 (38.45%)	

	las carreras de la Univ. de Sev.						
3.8.2	**Que becas y ayudas propias ofrece la Univ. de Sev.?**	8.3.4 (85.20%)	**3.8.2 (50.66%)**	3.2.5 (49.85%)	9.2.1 (47.61%)	3.10.1 (47.29%)	
	Que becas propias ofrece la Univ. de Sev.?	**3.8.2 (50.66%)**	3.2.5 (49.85%)	9.2.1 (47.61%)	3.10.4 (38.45%)	4.2.1 (38.22%)	
	Que ayudas propias ofrece la Univ. de Sev.?	8.3.4 (85.20%)	**3.8.2 (50.66%)**	3.2.5 (49.85%)	3.10.1 (42.11%)	3.1.4 (39.72%)	
	Como puedo conseguir una beca para estudiar la carrera	**3.8.2 (39.02%)**	3.8.1 (39.02%)	9.2.1 (36.40%)			
	Como puedo conseguir una ayuda para estudiar la carrera	9.2.1 (36.40%)	**3.8.2 (33.71%)**	3.8.1 (33.71%)			
	Como puedo conseguir una subvencion para estudiar la carrera	**3.8.2 (39.02%)**	3.8.1 (39.02%)				
3.8.3	**Deseo obtener el impreso de solicitud para la convocatoria de ayudas de colaboracion del SACU para Escuela Infantil**	8.3.4 (85.20%)	3.1.4 (57.16%)	3.10.1 (42.11%)	**3.8.3 (41.53%)**	10.1.1 (39.72%)	
	Deseo obtener el impreso de solicitud para la convocatoria de ayudas de colaboracion para la mediacion universitaria del instituto andaluz de la Juventud y la Univ. de Sev.	8.3.4 (85.20%)	8.1.1 (57.16%)	12.4.4 (50.21%)	9.2.1 (50.05%)	3.2.3 (48.00%)	**3.8.3 aparece en el puesto 8º con 41.53%**
	Deseo obtener el impreso de solicitud para la convocatoria de ayudas del SADUS	8.3.4 (85.20%)	3.1.4 (57.16%)	8.3.3 (44.16%)	3.10.1 (42.11%)	8.3.2 (39.84%)	**3.8.3 aparece en el puesto 6º con 38.99%**
	Deseo obtener el impreso de solicitud para la convocatoria de ayudas al Estudio de la Junta de Andalucia	8.3.4 (85.20%)	3.8.4 (62.66%)	3.1.4 (57.16%)	7.3.3 (51.66%)	3.10.1 (42.11%)	**3.8.3 aparece en el puesto 11º con 33.15%**
	Deseo obtener el impreso de solicitud para la convocatoria extraordinaria de ayudas del SACU, modalidad ayuda de cooperacion al desarrollo	8.3.4 (85.20%)	3.1.4 (57.16%)	9.4.1 (54.99%)	3.10.1 (42.11%)	**3.8.3 (41.53%)**	
	Deseo obtener el impreso de solicitud para la convocatoria de ayudas del CDE	8.3.4 (85.20%)	3.1.4 (57.16%)	3.10.1 (42.11%)	10.6.1 (40.00%)	**3.8.3 (38.99%)**	
	Deseo obtener el impreso de solicitud para la convocatoria de ayudas para los Bonos del Comedor	8.3.4 (85.20%)	3.1.4 (57.16%)	3.10.1 (42.11%)	**3.8.3 (41.53%)**	3.8.4 (31.90%)	
	Deseo obtener el impreso de solicitud para la convocatoria de ayudas para alumnos ganadores de las olimpiadas de fisica, matematicas y quimica	8.3.4 (85.20%)	3.1.4 (57.16%)	3.5.1 (44.16%)	1.5.5 (42.67%)	3.10.1 (42.11%)	**3.8.3 aparece en el puesto 6º con 41.53%**
3.8.4	**Deseo obtener info. acerca de la convocatoria de becas y ayudas al estudio de la Junta de Andalucia**	8.3.4 (85.20%)	**3.8.4 (62.67%)**	7.3.3 (51.66%)	3.10.1 (47.29%)	3.3.8 (41.15%)	
	Deseo obtener info. acerca de la convocatoria de becas de la Junta de Andalucia	7.3.3 (51.66%)	**3.8.4 (44.85%)**	3.2.2 (39.97%)	3.2.1 (38.87%)	3.10.4 (38.45%)	
	Deseo obtener info. acerca de la convocatoria de ayudas al estudio de la Junta de Andalucia	8.3.4 (85.20%)	**3.8.4 (62.66%)**	7.3.3 (51.66%)	3.10.1 (42.11%)	3.3.8 (41.15%)	

	Deseo obtener info. acerca de la convocatoria de becas de la Junta	**3.8.4 (48.11%)**	3.1.6 (40.00%)	3.10.4 (38.45%)	9.2.1 (36.40%)	3.8.5 (34.97%)	
3.8.5	**Deseo obtener info. acerca de la convocatoria de becas y ayudas del Ministerio de Educacion y Ciencia**	8.3.4 (85.20%)	**3.8.5 (61.56%)**	1.12.4 (47.77%)	3.10.1 (47.29%)	3.1.4 (39.72%)	
	Deseo obtener info. acerca de la convocatoria de becas y ayudas del MEC	8.3.4 (85.20%)	3.10.1 (47.29%)	3.1.4 (39.72%)	3.10.4 (38.45%)	3.8.4 (36.71%)	**3.8.5 aparece en el puesto 7º con 34.98%**
	Deseo obtener info. acerca de la convocatoria de becas del Ministerio de Educacion y Ciencia	**3.8.5 (61.61%)**	1.12.4 (47.77%)	3.11.1 (38.73%)	3.10.4 (38.45%)	2.3.1 (37.64%)	
	Deseo obtener info. acerca de la convocatoria de ayudas del MEC	8.3.4 (85.20%)	**3.8.5 (46.51%)**	3.10.1 (42.11%)	3.1.4 (39.72%)	3.8.4 (31.90%)	
3.9.1	**Desearia obtener info. sobre el carnet de estudiante**	**3.9.1 (54.99%)**	3.7.1 (51.82%)	1.5.4 (44.14%)	1.5.8 (34.52%)		
	Desearia obtener info. sobre el carnet universitario	1.9.2 (51.66%)	**3.9.1 (39.25%)**	1.5.8 (38.93%)			
	Desearia obtener info. sobre el carne de estudiante	**3.9.1 (54.99%)**	3.7.1 (51.82%)	1.5.4 (44.14%)	1.5.8 (34.52%)		
	Desearia obtener info. sobre la Tarjeta Inteligente Universitaria	**3.9.1 (54.99%)**	8.3.3 (44.16%)	3.1.6 (40.00%)	8.1.1 (40.00%)	10.1.1 (39.72%)	
	Desearia obtener info. sobre la TIU	**3.9.1 (39.25%)**					
3.10.1	**Me gustaria obtener info. acerca de los programas de becas y ayudas para la movilidad**	8.3.4 (85.20%)	**3.10.1 (83.48%)**	3.4.1 (51.66%)	4.1.4 (51.21%)	3.1.4 (39.72%)	
	Me gustaria obtener info. acerca de los programas de becas y ayudas para el intercambio de estudiantes	8.3.4 (85.20%)	**3.10.1 (52.89%)**	3.4.1 (51.66%)	4.1.4 (51.21%)	3.1.4 (39.72%)	
	Me gustaria obtener info. acerca de los programas de becas para la movilidad	**3.10.1 (62.43%)**	3.4.1 (51.66%)	4.1.4 (51.21%)	3.10.4 (38.45%)	10.4.16 (36.63%)	
	Me gustaria obtener info. acerca de los programas de ayudas para el intercambio de estudiantes	8.3.4 (85.20%)	**3.10.1 (67.78%)**	3.4.1 (51.66%)	4.1.4 (51.21%)	3.10.2 (39.98%)	
3.10.2	**Desearia info. acerca del programa SICUE**	**3.10.2 (62.43%)**	10.4.18 (37.87%)	10.4.28 (36.50%)	10.4.16 (34.49%)	10.4.7 (32.45%)	
	Desearia info. acerca del programa para el Sistema de Intercambio entre Centros Universitarios Españoles	**3.10.2 (45.22%)**	10.4.18 (37.87%)	2.3.1 (37.64%)	10.4.28 (36.50%)	10.4.16 (37.64%)	
3.10.3	**Desearia info. acerca del programa SENECA**	**3.10.3 (83.83%)**	10.4.18 (37.87%)	10.4.28 (36.50%)	10.4.16 (34.49%)	10.4.33 (32.78%)	
3.10.4	**Desearia info. acerca de las becas internacionales**	**3.10.4 (56.04%)**	3.10.5 (49.06%)	9.1.1 (44.16%)	9.4.1 (39.25%)	9.2.1 (36.40%)	
	Desearia info. acerca del programa Erasmus	**3.10.4 (48.84%)**	3.10.5 (44.15%)	10.4.18 (37.87%)	10.4.28 (36.50%)	10.4.16 (34.49%)	
3.10.5	**Desearia info. acerca de las becas internacionales**	3.10.4 (56.04%)	**3.10.5 (49.06%)**	9.1.1 (44.16%)	9.4.1 (39.25%)	9.2.1 (36.40%)	
	Desearia info. acerca del programa Erasmus Mundus	**3.10.5 (61.12%)**	3.10.4 (48.84%)	10.4.18 (37.87%)	10.4.28 (36.50%)	10.4.16 (34.49%)	
3.11.1	**Me gustaria acceder al Espacio Europeo de Educacion Superior**	**3.11.1 (61.75%)**	1.12.4 (61.61%)	3.8.5 (39.98%)	12.3.2 (38.21%)	2.3.1 (37.64%)	
	Me gustaria acceder al EEES	**3.11.1 (38.73%)**					
	Tengo una duda acerca de los Creditos ECTS	**3.11.1 (53.24%)**					

Tabla B.2. Resultados detallados para el Tema 3 con el método TF-IDF.

Anexo C. Reglas borrosas.

En el anexo C, se definen exhaustivamente las reglas borrosas utilizadas para los distintos motores de FL utilizados para el desarrollo del Agente Inteligente, tanto los usados para la Extracción de Información como los utilizados para la definición de los coeficientes de peso.

C.1. Reglas borrosas para la extracción de información

Reglas borrosas para el motor borroso de tres entradas.

E1	E2	E3	Salida
A	A	A	A
A	A	M	A
A	A	B	A
A	M	A	A
A	M	M	A
A	M	B	A
A	B	A	A
A	B	M	A
A	B	B	A
M	A	A	A
M	A	M	A
M	A	B	A
M	M	A	A
M	M	M	A
M	M	B	MA
M	B	A	A
M	B	M	MA
M	B	B	MB
B	A	A	A
B	A	M	A
B	A	B	A
B	M	A	A
B	M	M	MA

B	M	B	MB
B	B	A	A
B	B	M	MB
B	B	B	B

Tabla C.1. Reglas borrosas para el motor borroso de tres entradas para la Extracción de Información.

Reglas borrosas para el motor de cinco entradas.

E1	E2	E3	E4	E5	Salida
A	A	A	A	A	A
A	A	A	A	M	A
A	A	A	A	B	A
A	A	A	M	A	A
A	A	A	M	M	A
A	A	A	M	B	A
A	A	A	B	A	A
A	A	A	B	M	A
A	A	A	B	B	A
A	A	M	A	A	A
A	A	M	A	M	A
A	A	M	A	B	A
A	A	M	M	A	A
A	A	M	M	M	A
A	A	M	M	B	A
A	A	M	B	A	A
A	A	M	B	M	A
A	A	M	B	B	A
A	A	B	A	A	A
A	A	B	A	M	A
A	A	B	A	B	A
A	A	B	M	A	A
A	A	B	M	M	A
A	A	B	M	B	A
A	A	B	B	A	A
A	A	B	B	M	A
A	A	B	B	B	A
A	M	A	A	A	A
A	M	A	A	M	A
A	M	A	A	B	A
A	M	A	M	A	A
A	M	A	M	M	A
A	M	A	M	B	A
A	M	A	B	A	A
A	M	A	B	M	A
A	M	A	B	B	A

A	M	M	A	A	A
A	M	M	A	M	A
A	M	M	A	B	A
A	M	M	M	A	A
A	M	M	M	M	A
A	M	M	M	B	A
A	M	M	B	A	A
A	M	M	B	M	A
A	M	M	B	B	A
A	M	B	A	A	A
A	M	B	A	M	A
A	M	B	A	B	A
A	M	B	M	A	A
A	M	B	M	M	A
A	M	B	M	B	A
A	M	B	B	A	A
A	M	B	B	M	A
A	M	B	B	B	MA
A	B	A	A	A	A
A	B	A	A	M	A
A	B	A	A	B	A
A	B	A	M	A	A
A	B	A	M	M	A
A	B	A	M	B	A
A	B	A	B	A	A
A	B	A	B	M	A
A	B	A	B	B	A
A	B	M	A	A	A
A	B	M	A	M	A
A	B	M	A	B	A
A	B	M	M	A	A
A	B	M	M	M	A
A	B	M	M	B	A
A	B	M	B	A	A
A	B	M	B	M	A
A	B	M	B	B	MA
A	B	B	A	A	A
A	B	B	A	M	A
A	B	B	A	B	A
A	B	B	M	A	A
A	B	B	M	M	A
A	B	B	M	B	MA
A	B	B	B	A	A
A	B	B	B	M	A
A	B	B	B	B	MB
M	A	A	A	A	A
M	A	A	A	M	A
M	A	A	A	B	A
M	A	A	M	A	A

M	A	A	M	M	A
M	A	A	M	B	A
M	A	A	B	A	A
M	A	A	B	M	A
M	A	A	B	B	A
M	A	M	A	A	A
M	A	M	A	M	A
M	A	M	A	B	A
M	A	M	M	A	A
M	A	M	M	M	A
M	A	M	M	B	A
M	A	M	B	A	A
M	A	M	B	M	A
M	A	M	B	B	A
M	A	B	A	A	A
M	A	B	A	M	A
M	A	B	A	B	A
M	A	B	M	A	A
M	A	B	M	M	A
M	A	B	M	B	A
M	A	B	B	A	A
M	A	B	B	M	A
M	A	B	B	B	MA
M	M	A	A	A	A
M	M	A	A	M	A
M	M	A	A	B	A
M	M	A	M	A	A
M	M	A	M	M	A
M	M	A	M	B	A
M	M	A	B	A	A
M	M	A	B	M	A
M	M	A	B	B	A
M	M	M	A	A	A
M	M	M	A	M	A
M	M	M	A	B	A
M	M	M	M	A	A
M	M	M	M	M	A
M	M	M	M	B	A
M	M	M	B	A	A
M	M	M	B	M	A
M	M	M	B	B	MA
M	M	B	A	A	A
M	M	B	A	M	A
M	M	B	A	B	A
M	M	B	M	A	A
M	M	B	M	M	A
M	M	B	M	B	MA
M	M	B	B	A	A
M	M	B	B	M	MA
M	M	B	B	B	MB

M	B	A	A	A	A
M	B	A	A	M	A
M	B	A	A	B	A
M	B	A	M	A	A
M	B	A	M	M	A
M	B	A	M	B	A
M	B	A	B	A	A
M	B	A	B	M	A
M	B	A	B	B	A
M	B	M	A	A	A
M	B	M	A	M	A
M	B	M	A	B	A
M	B	M	M	A	A
M	B	M	M	M	A
M	B	M	M	B	MA
M	B	M	B	A	A
M	B	M	B	M	MA
M	B	M	B	B	MB
M	B	B	A	A	A
M	B	B	A	M	A
M	B	B	A	B	A
M	B	B	M	A	A
M	B	B	M	M	MA
M	B	B	M	B	MB
M	B	B	B	A	MA
M	B	B	B	M	MB
M	B	B	B	B	B
B	A	A	A	A	A
B	A	A	A	M	A
B	A	A	A	B	A
B	A	A	M	A	A
B	A	A	M	M	A
B	A	A	M	B	A
B	A	A	B	A	A
B	A	A	B	M	A
B	A	A	B	B	A
B	A	M	A	A	A
B	A	M	A	M	A
B	A	M	A	B	A
B	A	M	M	A	A
B	A	M	M	M	A
B	A	M	M	B	A
B	A	M	B	A	A
B	A	M	B	M	A
B	A	M	B	B	MA
B	A	B	A	A	A
B	A	B	A	M	A
B	A	B	A	B	A
B	A	B	M	A	A

B	A	B	M	M	A
B	A	B	M	B	MA
B	A	B	B	A	A
B	A	B	B	M	MA
B	A	B	B	B	MB
B	M	A	A	A	MB
B	M	A	A	M	A
B	M	A	A	B	A
B	M	A	M	A	A
B	M	A	M	M	A
B	M	A	M	B	A
B	M	A	B	A	A
B	M	A	B	M	A
B	M	A	B	B	MA
B	M	M	A	A	A
B	M	M	A	M	A
B	M	M	A	B	A
B	M	M	M	A	A
B	M	M	M	M	A
B	M	M	M	B	MA
B	M	M	B	A	A
B	M	M	B	M	MA
B	M	M	B	B	MB
B	M	B	A	A	A
B	M	B	A	M	A
B	M	B	A	B	MA
B	M	B	M	A	A
B	M	B	M	M	MA
B	M	B	M	B	MB
B	M	B	B	A	MA
B	M	B	B	M	MB
B	M	B	B	B	B
B	B	A	A	A	A
B	B	A	A	M	A
B	B	A	A	B	A
B	B	A	M	A	A
B	B	A	M	M	A
B	B	A	M	B	MA
B	B	A	B	A	A
B	B	A	B	M	MA
B	B	A	B	B	MB
B	B	M	A	A	A
B	B	M	A	M	A
B	B	M	A	B	MA
B	B	M	M	A	A
B	B	M	M	M	MA
B	B	M	M	B	MB
B	B	M	B	A	MA
B	B	M	B	M	MB
B	B	M	B	B	B

B	B	B	A	A	A
B	B	B	A	M	MA
B	B	B	A	B	MB
B	B	B	M	A	MA
B	B	B	M	M	MB
B	B	B	M	B	B
B	B	B	B	A	MB
B	B	B	B	M	B
B	B	B	B	B	B

Tabla C.2. Reglas borrosas para el motor borroso de cinco entradas para la Extracción de Información.

C.2. Reglas borrosas para la definición de pesos.

A diferencia del caso del motor borroso utilizado para la extracción de información, las entradas no son simétricas, es decir, influye el orden de dichas entradas. Nótese que igualmente se han definido cuatro salidas (ALTO, MEDIO-ALTO, MEDIO-BAJO y BAJO). En el caso de los coeficientes de peso de nivel 1 y 2 (Tema y Apartado), las reglas borrosas quedan como se muestra en la Tabla C.3.

P1	P2	P3	P4	Salida
A	A	A	A	A
A	A	A	M	MA
A	A	A	B	MA
A	A	M	A	MA
A	A	M	M	MA
A	A	M	B	MB
A	A	B	A	MB
A	A	B	M	MB
A	A	B	B	MB
A	M	A	A	A
A	M	A	M	MA
A	M	A	B	MA
A	M	M	A	MA
A	M	M	M	MA
A	M	M	B	MB
A	M	B	A	MA
A	M	B	M	MB
A	M	B	B	B
A	B	A	A	A
A	B	A	M	MA
A	B	A	B	MA
A	B	M	A	MA
A	B	M	M	MA
A	B	M	B	MB
A	B	B	A	MB

A	B	B	M	MB
A	B	B	B	B
M	A	A	A	A
M	A	A	M	MA
M	A	A	B	MA
M	A	M	A	MA
M	A	M	M	MA
M	A	M	B	MB
M	A	B	A	MB
M	A	B	M	MB
M	A	B	B	B
M	M	A	A	A
M	M	A	M	MA
M	M	A	B	MB
M	M	M	A	MA
M	M	M	M	MA
M	M	M	B	MB
M	M	B	A	MB
M	M	B	M	MB
M	M	B	B	B
M	B	A	A	MB
M	B	A	M	MB
M	B	A	B	MB
M	B	M	A	B
M	B	M	M	B
M	B	M	B	B
M	B	B	A	B
M	B	B	M	B
M	B	B	B	B
B	A	A	A	MA
B	A	A	M	MA
B	A	A	B	MB
B	A	M	A	MA
B	A	M	M	MB
B	A	M	B	B
B	A	B	A	B
B	A	B	M	B
B	A	B	B	B
B	M	A	A	MB
B	M	A	M	B
B	M	A	B	B
B	M	M	A	B
B	M	M	M	B
B	M	M	B	B
B	M	B	A	B
B	M	B	M	B
B	M	B	B	B
B	B	A	A	MB
B	B	A	M	B
B	B	A	B	B

B	B	M	A	B
B	B	M	M	B
B	B	M	B	B
B	B	B	A	B
B	B	B	M	B
B	B	B	B	B

Tabla C.3. Reglas borrosas para la asignación de coeficientes de peso (Niveles de Tema y Apartado).

Para el último nivel jerárquico (nivel de Objeto), prescindimos de P2, puesto que no tiene sentido. Por tanto, las reglas quedan de la forma mostrada en la Figura C.4.

P1	P3	P4	Salida
A	A	A	A
A	A	M	MA
A	A	B	MA
A	M	A	MA
A	M	M	MA
A	M	B	MB
A	B	A	MB
A	B	M	MB
A	B	B	B
M	A	A	A
M	A	M	MA
M	A	B	MB
M	M	A	MA
M	M	M	MA
M	M	B	MB
M	B	A	MB
M	B	M	MB
M	B	B	B
B	A	A	MB
B	A	M	MB
B	A	B	MB
B	M	A	MB
B	M	M	MB
B	M	B	B
B	B	A	B
B	B	M	B
B	B	B	B

Tabla C.4. Reglas borrosas para la asignación de coeficientes de peso (Nivel de Objeto).

Bibliografía.

[ABULAISH05]: M. Abulaish, y L. Dey. "Biological Ontology Enhancement with Fuzzy Relations: A Text-Mining Framework". Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence, France, pp. 379-385. 2005.

[AJAYI09]: A.O. Ajayi. G.A. Aderounmu y H.A. Soriyan. "An adaptive fuzzy information retrieval model to improve response time perceived by e-commerce clients". Expert Systems with Applications (ESWA) Vol. 37(1), pp. 82-91, 2010.

[ALAG08]: S. Alag. "Collective Intelligence in Action". Manning Pubn, online ed., September 2008.

[AN07]: A. An, J. Stefanowski, S. Ramanna y C. Butz. "Rough sets, fuzzy sets, data mining and granular computing". 11th International Conference, RSFDGrC 2007, Toronto, Canada, May 14-16, 2007.

[ARANO03]: S. Arano. "La ontología: una zona de interacción entre la Lingüística y la Documentación". Hipertext.net, núm. 2, 2003

[ARONSON94]: A.R. Aronson, T.C. Rindflesch y A.C. Browne, "Exploiting a large thesaurus for information retrieval". Proceedings of RIAO, 197-216. 1994.

[ARONSON97]: A.R. Aronson y T.C. Rindflesch, "Query Expansion Using the UMLS Metathesaurus". Journal of the American Medical Informatics Association, 1997: Supplement. Proceedings of the 1997 AMIA Annual Fall Symposium, p. 485-489. 1997.

[ASISTENTES09]: http://www.asistentesvirtuales.com/index.htm#

[BAEZA-YATES99]: R. Baeza-Yates y B. Ribeiro-Neto, "Modern Information Retrieval". ACM Press / Addison-Wesley. 1999.

[BARBANCHO09]: J. Barbancho. "Prescripciones técnicas para el diseño y construcción de apoyo al SOS de la Universidad de Sevilla". Departamento de Tecnología Electrónica. Universidad de Sevilla. Technical Report 0403-29. 2009.

[BELEW89]: R.K. Belew**, "**Adaptive Information Retrieval: Using a Connectionist Representation to retrieve and learn about documents". Proceedings of 12 th ACM-SIGIR Conference, pp. 11-20, Cambridge, Mass., USA. 1989

[BERNERS-LEE02]: T. Berners-Lee y E. Miller. “The Semantic Web lifts off”. ERCIM News No. 51, October 2002. Special Semantic Web.

[BICKMORE09]: T. W. Bickmore, L. M. Pfeifer, y M. K. Paasche-Orlow, “Using computer agents to explain medical documents to patients with low health literacy”. Patient Education and Counseling, vol. 75, no. 3, pp. 315-320, June 2009.

[BOOKSTEIN95]: A. Bookstein, S.T. Klein y T. Raita, “Detecting content bearing words by serial clustering”. SIGIR Forum (ACM Special Interest Group on Information Retrieval), p. 319-327, 1995.

[BOUTELL08]: http://www.boutell.com/newfaq/misc/sizeofweb.html

[BURGET04]: R. Burget, “Information Extraction from HTML Document Based on Logical Document Structure”. Tesis Doctoral. Universidad de Brno, 2004.

[CAJAMADRID09]: http://www.cajamadrid.es/CajaMadrid/Home/puente?pagina=3447

[CANTOR06]: G. Cantor. “Fundamentos para una teoría general de conjuntos: escritos y correspondencia selecta”. Barcelona: Crítica. 2006.

[CHAKRABARTI98]: S. Chakrabarti, B. Dom, R. Agrawal y P. Raghavan, “Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies”. VLDB Journal Vol.7 3, p. 163-178, 1998.

[CHAKRABARTI00]: S. Chakrabarti, “Data Mining for hypertext: a tutorial survey”. ACM SIGKDD Explorations, Newsletter of the Special Interest Group on Knowledge Discovery and Data Mining, 2000.

[COOLEY97] R. Cooley, B. Mobasher y J. Srivastava, “Web Mining: Information and Pattern Discovery on the World Wide Web”. Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence. ICTAI’97, 1997.

[CORDÓN04]: O. Cordon, F. de Moya y C. Zarco, “Fuzzy logic and multiobjective evolutionary algorithms as soft computing tools for persistent query learning in text retrieval environments”. Proceedings of the IEEE International Conference on Fuzzy Systems, vol.1, pp. 571-576, 25-29 July 2004.

[CREATIVE09]: http://www.creativevirtual.com/

[DE_KOOL08]: D. de Kool y J. van Wamelen, “Web 2.0: A New Basis for E-Government” 3rd International Conference on Information and Communication Technologies: From Theory to Applications, ICTTA 2008., vol. 1 pp.1-7, 7-11 April 2008.

[DING04]: C. Ding y X. He. “K-means Clustering via Principal Component Analysis”. Proceedings of the International Conference in Machine Learning, ICML04, pp 225-232. July 2004.

[DUARTE98]: http://www.ing.unal.edu.co/~ogduarte/softwareDetallado.htm

[EISMAN09]: E. M. Eisman, V.Lopez y J.L. Castro, "Controlling the emotional state of an embodied conversationalagent with a dynamic probabilistic fuzzy rules based system". Expert Systems with Applications, Vol. 36, Issue 6, pp. 9698-9708, August 2009.

[ETZIONI96] O. Etzioni, "The World Wide Web: Quagmire or gold mine". Commum. ACM, vol. 39, no. 11, pp. 65-68, 1996.

[FRAWLEY91] W. Frawley, G. Piatetsky-Shapiro y C. Matheus, "Knowledge Discovery in Databases: an Overview". Eds. Menlo Park, Ca: AAAI Press, pp 1-7. 1991.

[FREITAG98] D. Freitag, "Information Extraction from HTML: Application of a General Learning Approach". Proceedings of the Fifteenth National Conference on Artificial Intelligence, pp 517-524. 1998.

[FRIEDMAN04]: M. Friedman, M. Last, O. Zaafrany, M. Schneider y A. Kandel, "A new approach for fuzzy clustering of Web documents". Proceedings of the IEEE International Conference on Fuzzy Systems, vol.1, pp. 377-381, 25-29 July 2004.

[GÁLVEZ08]: C. Gálvez. "Minería de textos: la nueva generación de análisis de literatura científica en biología molecular y genómica". Encontros Bibli: Revista Eletrônica de Biblioteconomia e Ciência da Informação, vol. 13, n. 25. 2008.

[GARCÍA-SERRANO04]: A. Garcia-Serrano, P. Martínez y J. Hernández, "Using AI techniques to support advanced interaction capabilities in a virtual assistant for e-commerce". Expert Systems with Applications, Vol. 26, Issue 3, pp. 413–426. 2004

[GEHLERT07] A. Gehlert y W. Esswein, "Toward a formal research framework for ontological analyses". Advanced Engineering Informatics 21(2): 119-131, 2007.

[GÓMEZ06]: A. Gómez, J. Ropero y C. León. "A fuzzy logic system for classifying the contents of a database and searching consultations in natural language". Proceedings of the Mediterranean Electrotechnical Conference - MELECON 2006, pp. 721-724, 2006.

[GREENES06]: R. A. Greenes. "Clinical Decision Making - The Road Ahead". Elsevier, November 2006.

[GRUBER93]: T.R. Gruber. "A translation approach to portable ontologies". Knowledge Acquisition, 5(2), 1993.

[GUARINO95]: N. Guarino y P. Giaretta. "Ontologies and Knowledge Bases: Towards a Terminological Clarification". Towards Very Large Knowledge Bases: Knowledge Building and Knowledge sharing, N. Mars (ed.) IOS Press, Amsterdam, 1995, pp. 25-32.

[HAASE02]: V. H. Haase, C. Steinmann y S. Vejda. "Access to Knowledge: Better Use of the Internet". IS2002 Proceedings of the Informing Science + IT Education Conference. Cork, Ireland, pp. 618-627 . June 19-21, 2002.

[HARDEBERG01]: J. Y. Hardeberg. “Acquisition and Reproduction of Color Images: Colorimetric and Multispectral Approaches”. Universal-Publishers.com. 2001.

[HARMAN91] D. Harman, “How effective is suffixing?” Journal of the American Society for Information Science, 42(1): pp 7-15, 1991.

[HERNÁNDEZ04] J. Hernández Orallo, M. Ramírez Quintana y C. Ferri Ramírez, “Introducción a la Minería de Datos. Prentice-Hall, 2004.

[HIROTA 99] K. Hirota y W. Pedrycz, “Fuzzy Computing for Data Mining”. Proceedings of the IEEE, Vol.87, no.9, pp.1575-1600. 1999.

[HORNG05]: Y. J. Horng, S. M. Chen, Y. C. Chang, C. H. Lee. “A new method for fuzzy information retrieval based on fuzzy hierarchical clustering and fuzzy inference techniques”. IEEE T. Fuzzy Systems 13, vol. 2, pp. 216-228. 2005.

[IANNONE07] L. Iannone, I. Palmisano y N. Fanizzi, “An algorithm based on counterfactuals for concept learning in the Semantic Web”. Applied Intelligence, 26(2), ISSN 0924-669X 2007 pp. 139-159, Springer 2007.

[INDISYS09]: http://www.indisys.es/

[JANG92]: J.S.R. Jang. “ANFIS : Adaptative Network based Fuzzy Inference System”. IEEE Trans. Syst, Man and Cybern., 23 665-685. 1992.

[JANTZEN98]: J. Jantzen, “Tutorial on fuzzy logic”. Technical University of Denmark, Department of Automation, Tech. report no 98-E 868. 1998.

[JOHN95]: R.I. John, “Fuzzy Inferencing Systems – Problems and Some Solutions”. Computing Science Research. School of Computing Sciences. De Montfort University. The Gateway, Leicester, Working Paper N. 62, Diciembre, 1995.

[JOHNSON08]: E. Johnson y J. Jones. “A developer's guide to data modeling for SQL server: covering SQL server 2005 and 2008”. Upper Saddle River, N.J. Addison-Wesley. 2008.

[KERLY07]: A. Kerly, R. Ellis y S. Bull, “CALMsystem: A Conversational Agent for Learner Modelling”. Knowledge-Based Systems, Vol. 21, Issue 3, pp. 238-246, Abril 2008.

[KIM06]: K. Kim, J. Hong, and S. Cho, “A semantic Bayesian network approach to retrieving information with intelligent conversational agents”. Information Processing Management, Vol. 43, Issue 1, pp.225-236, January 2007.

[KLOGSEN02] W. Klogsen y J. Zytkow, “Handbook of data mining and knowledge discovery””. New York: Oxford University Press. 2002

[KOSALA00] R. Kosala y H. Blockeel, “Web Mining Research: A Survey”. SIGKDD: SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery & Data Mining, ACM, Vol. 2 (2000).

[KROVETZ93] R. Krovetz, "Viewing morphology as an inference process". Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1993.

[KUSHMERICK02] N. Kushmerick, "Gleaning answers from the Web". Proceedings of the AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases, Palo Alto, 43-45. 2002.

[KWOK89] K.L. Kwok, "A neural network for probabilistic information retrieval". Proceedings of the 12th annual international ACM SIGIR conference on Research and development in information retrieval. Cambridge, Massachusetts, United States. 1989.

[LARIOS04] D. Larios. "Un-Fuzzy y la implementación de sistemas de lógica difusa: Comparación con otras alternativas". 2004.

[LARSEN93] H. Larsen, R. Yager, "The use of fuzzy relational thesauri for classificatory problem solving in information retrieval and expert systems". IEEE Transactions On Systems, Man, and Cybernetics. 1993; Vol 23(1):31-41. 1993.

[LARSEN99]: H. L. Larsen. "An Approach to Flexible Information Access Systems using Soft Computing". Proceedings of the 32nd Annual Hawaii International Conference on System Sciences, HICCS99. 5-8 January, 1999.

[LEE97]: D.L. Lee, H. Chuang, K. Seamons, "Document ranking and the vector-space model". IEEE Software, Vol. 14. Issue 2, pp. 67 – 75. 1997.

[LERTNATTEE03]: V. Lertnattee, T. Theeramunkong, "Combining homogenous classifiers for centroid-based text classification". Proceedings of the 7th International Symposium on Computers and Communications, pp. 1034-1039. 2003

[LEU07]: Y.G. Leu, C.M. Hong, H.J. Zhon: "GA-Based Adaptive Fuzzy-Neural Control for a Class of MIMO Systems". Lecture Notes In Computer Science; Vol. 4491. Proceedings of the 4th international symposium on Neural Networks: Advances in Neural Networks. Nanjing, China, p. 45 – 53. 2007

[LIAO05]: S.H. Liao. "Expert system methodologies and applications—a decade review from 1995 to 2004". Expert Systems with Applications, Vol. 28, Issue 1, pp. 93-103. 2005.

[LIU01] S. Liu, M. Dong, H. Zhang, R. Li, Z. Shi, "An approach of multi-hierarchy text classification Proceedings of the International Conferences on Info-tech and Info-net, 2001". Beijing. Vol 3, pp. 95 – 100. 2001.

[LIU05] L. Liu, D. Buttler, J. Caverlee, C. Pu y J. Zhang, "A methodical approach to extracting interesting objects from dynamic web pages". Int. Journal of Web and Grid Services 2005 - Vol. 1, No.2 pp. 165 - 195. 2005.

[LIU07] D.R. Liu y C.K. Ke, “Knowledge support for problem-solving in a production process: A hybrid of knowledge discovery and case-based reasoning”. Expert Systems with Applications. 33(1): 147-161. 2007.

[LOH03] S. Loh, J. Palazzo, M. De Oliveira y M. Gameiro, “Knowledge Discovery in Texts for Constructing Decision Support Systems”. Applied Intelligence, New York, NY, USA, v. 18, n. 3, p. 357-366, 2003.

[LU02] M. Lu, K. Hu, Y. Wu, Y. Lu y L. Zhou, “SECTCS: towards improving VSM and Naive Bayesian classifier”. IEEE International Conference on Systems, Man and Cybernetics, Vol. 5, p. 5. 2002.

[MARTHI03]: B. Marthi, B. Milch y S. Russell, “First-order probabilistic models for information extraction”. Proceedings of the Workshop on Learning Statistical Models from Relational Data, IJCAI 2003.

[MARTÍN-DEL-BRÍO01]: B. Martín del Brío y A. Sanz Molina**, “**Redes neuronales y sistemas borrosos”. Ra-Ma, 2001.

[MATEO06]: R. M. A. Mateo, M. Lee, S.C. Joo y J. Lee. “Location-Aware Data Mining for Mobile Users Based on Neuro-fuzzy System”, Lecture Notes in Computer Science, Vol.4223, pp. 1269 – 1278, 8-11 May 2006.

[MATHWORKS02]: “Fuzzy Logic Toolbox. User’s guide”. The Mathworks Inc., 2002.

[MENGUAL01]: L. Mengual, N. Barcia, J. Bobadilla, E. Jiménez, J. Setién, J. Yágüez. “Arquitectura multi-agente segura basada en un sistema de implementación automática de protocolos de seguridad”. I Simposio Español de Negocio Electrónico. Málaga 2001.

[MERCIER05]: A. Mercier, M. Beigbeder. “Fuzzy Proximity Ranking with Boolean Queries”. Proceedings of the 14th Text REtrieval Conference (TREC 2005). Gaithersburg, Maryland, USA, 15-18 Novembre 2005.

[MILLER05]: P. Miller, “Web 2.0: building the new library”. Ariadne, vol. 45, 2005.

[MORADI08]: P. Moradi, M. Ebrahim, M.M. Ebadzadeh. “Personalizing Results of Information Retrieval Systems Using Extended Fuzzy Concept Networks”. 3rd International Conference on Information and Communication Technologies: From Theory to Applications, ICTTA 2008, pp.1-7, 7-11 April 2008.

[OLSON07]: D. Olson e Y. Shi, “Introduction to business data mining”. McGraw-Hill. 2007.

[OSIMO07]: D. Osimo y J.C. Burgelman, “Web 2.0 for e-Government: Why and How”, 4th Ministerial e-Government Conference, Lisboa 2007.

[PAIJMANS99]: J. J. Paijmans: "Explorations in the Document Vector Model of Information Retrieval", pp. 16--19, 1999.

[PAL02] S. K. Pal, V. Talwar and P. Mitra, "Web Mining in Soft Computing Framework: Relevance, State of the Art and Future Directions". IEEE Transactions on Neural Networks Vol. 13, No. 5 pp 1163-1177, September 2002.

[PANT04]: S.N. Pant, K.E. Holbert. "Fuzzy Logic in Decision Making and Signal Processing". Powerzone, Arizona State University, Abril 2004.

[PAPADAKIS05] N. K. Papadakis, D. Skoutas, K. Raftopoulos y T. A. Varvarigou, "STAVIES: A System for Information Extraction from Unknown Web Data Sources through Automatic Web Wrapper Generation Using Clustering Techniques". IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 12, pp. 1638-1652, Dec., 2005.

[PELACHAUD08]: C. Pelachaud, "Studies on gesture expressivity for a virtual agent". Speech Commun. Vol. 51, Issue 7 pp. 630-639, July 2009.

[PIERRE02]: S. Pierre, "Intelligent and Heuristic Approaches and Tools for the Topological Design of Data Communication Networks". Data Communication Network Techniques and Applications. New York Academic Press, pp. 289-326, 2002.

[POPOVIC92]: M. Popovic, P. Willett. "The effectiveness of stemming for natural language access to Slovene textual data". Journal of the American Society for Information Science, 43(5): pp. 384-390, 1992.

[PORTER80] M. Porter, "An Algorithm for Suffix Stripping". Program, 14(3): 130-137.

[PRINCE06] V. Prince y M. Lafourcade, "Mixing Semantic Networks and Conceptual Vectors: the Case of Hyperonymy". IEEE Proceedings of the 2nd International Conf. on Cognitive Informatics (ICCI'03), pp. 121-128. Marzo 2006.

[QAVATAR09]: http://www.qavatar.com/new/e_home.html.

[QUAN06]: T.T. Quan, S.C. Hui, A.C.M. Fong. "Automatic fuzzy ontology generation for semantic help-desk support". Industrial Informatics, IEEE Transactions on, Vol. 2, No. 3. (2006), pp. 155-164.

[RAGHAVAN86] V.V. Raghavan y S. K. Wong, "A critical analysis of vector space model for information retrieval". Journal of the American Society for Information Science, Vol.37 (5), p. 279-87, 1986

[RFC1738]: T. Berners-Lee, et. al., "Uniform Resource Locators (URL)", RFC 1738, CERN, December 1994, http://www.internic.net/rfc/rfc1738.txt.

[RÍOS06]: S.A. Ríos, J.D. Velásquez, H. Yasuda, T. Aoki. "Improving the web site text content by extracting concept-based knowledge". Lecture Notes in Artificial Intelligence, 4252 Vol. 1, pp. 371-378. 2006.

[ROMERO07] C. Romero y S. Ventura, "Educational data mining: A survey from 1995 to 2005". Expert Systems with Applications. 33(1): 135-146. 2007.

[ROPERO07a]: J. Ropero, A. Gómez, C. León y A. Carrasco, "A method for the access to the contents in a set of knowledge using a fuzzy logic based intelligent agent". Proceedings - Fourth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2007, Vol. 4, pp. 103-108, 2007.

[ROPERO07b]: J. Ropero, A. Gómez, C. León y A. Carrasco, "Information extraction in a set of knowledge using a fuzzy logic based intelligent agent". Lecture Notes in Computer Science, Vol. 4707 LNCS, part 3, pp. 811-820. 2007.

[ROPERO09]: J. Ropero, A. Gómez, Carlos León y A. Carrasco, "Term Weighting: Novel Fuzzy Logic Based Method Vs. Classical TF-IDF Method for Web Information Extraction". Proceedings of the 11th International Conference on Enterprise Information Systems. Iceis 2009, Milán, pp. 130-137. 2009.

[RUIZ98] M.E. Ruiz y P. Srinivasan, "Automatic Text Categorization Using Neural Networks". Advances in Classification Research vol. 8: Proceedings of the 8th ASIS SIG/CR Classification Research Workshop. Ed. Efthimis Efthimiadis. Information Today, Medford:New Jersey. 1998. pp 59-72.

[RUIZ02]: M.E. Ruiz y P. Srinivasan. "Hierarchical text categorization using neural networks". Information Retrieval, Vol. 5, Nº 1, pp. 87-118. 2002.

[RUSSELL99]: J.A. Russell, L.F. Barrett, "Core affect, prototypical emotional episodes, and other things called emotion: Dissecting the elephant". Journal of Personality and Social Psychology, Vol. 76, Issue 5, pp. 805–819. 1999.

[SALTON83]: G. Salton, McGill, "Introduction to modern information retrieval". McGraw-Hill.1983.

[SALTON88]: G. Salton. Automatic Text Processing. Addison-Wesley Publishing Company, 1988.

[SALTON96]: G. Salton, C. Buckley. Term Weighting Approaches in Automatic Text Retrieval. Technical Report TR87-881, Department of Computer Science, Cornell University, 1987. Information Processing and Management Vol.32 (4), p. 431-443, 1996.

[SPARCK-JONES99]: K. Sparck-Jones, "What Is The Role for NLP in Text Retrieval". T. Strzalkowski (ed.). Natural Language Information Retrieval. Kluwer. pp. 1-25. 1999.

[SUBASIC01]: P. Subasic, A. Huettner. **"**Affect Analysis of Text Using Fuzzy Semantic Typing"**.** IEEE Transactions on Fuzzy Systems, Special Issue, 2001.

[SYCARA96]: K. Sycara, K. Decker, A. Pannu, M. Williamson, D. Zeng. "Distributed Intelligent Agents". IEEE Expert, Diciembre 1996.

[TAO08] Y.H. Tao, T.P. Hong y Y.M. Su, "Web usage mining with intentional browsing data". Expert Systems Applications. 34(3): 1893-1904. 2008.

[TURBAN01]: E. Turban, J.E. Aronson. "Decisión support systems and intelligent systems". 6th ed. Hong Kong: Prentice Internacional Hall. 2001.

[US08]: Universidad de Sevilla, "Memoria del Curso Académico 2007-2008". http://servicio.us.es/secgral/normativa/memoria07_08.pdf. 2008.

[US09]: Universidad de Sevilla, "Anuario estadístico del Curso Académico 2008-2009". http://servicio.us.es/splanestu/WS/Present.pdf.

[VANRIJSBERGEN79]: C.J. van Rijsbergen, "Information retrieval". Butterworths, 1979.

[VERCELLIS09]: C. Vercellis, "Business Intelligence: Data Mining and Optimization for Decision Making". Wiley Publishing. 2009.

[VOICETEXT09]: http://www.voicetext.co.uk/

[WEBOMETRICS09]: http://www.webometrics.info/index.html .2009.

[WIK09]: P. Wik y A. Hjalmarsson, "Embodied conversational agents in computer assisted language learning". *Speech Commun.* Vol. 51, Issue 10, 1024-1037. Octubre 2009.

[XIE05]: D. Xie, "Fuzzy Association Rules Discovered on Effective Reduced Database Algorithm,". Proceedings of the 14th IEEE International Conference on Fuzzy Systems, FUZZ '05, pp.779-784, 25 May 2005.

[XU03] J. Xu y Z. Wang, Z. "TCBLSA: A new method of text clustering". International Conference on Machine Learning and Cybernetics. Vol. 1, pp. 63-66. 2003.

[YAHOO05] http://www.ysearchblog.com/archives/000172.html

[YANARU97]: T. Yanaru, N. Shirahama, K. Yoshida y M. Nagamatsu, "An emotion processing system based on fuzzy inference and subjective observations". Information Sciences, Vol. 101, Issues 3–4, pp. 217–247. 1997.

[ZADEH65]: L.A. Zadeh. "Fuzzy Sets". Information & Control, 8, 338-353, 1965.

[ZADEH94]: L.A. Zadeh. "Fuzzy logic, neural networks and soft computing". Communications of the ACM, 3, 3, 77-84. 1984.

[ZADEH01]: L.A. Zadeh. Prefacio del libro. [MARTÍN-DEL-BRÍO01].

[ZHAI08]: J. Zhai, Q. Wang, M. Lv, "Application of Fuzzy Ontology Framework to Information Retrieval for SCM". Proceedings of ISIP08, International Symposiums on Information Processing, pp.173-177. 2008.

[ZHANG03]: R. Zhang, Z. Zhang . "Addressing CBIR efficiency, effectiveness, and retrieval subjectivity simultaneously". Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval, MIR03, pp 71-78, New York, NY, USA. ACM Press. 2003.

[ZHAO02] Y. Zhao y G. Karypis, “Improving precategorized collection retrieval by using supervised term weighting schemes”. Proceedings of the International Conference on Information Technology: Coding and Computing, 2002. pp 16 – 21.

Glosario de abreviaturas.

AI: Artificial Intelligence (Inteligencia Artificial).

ALU : Aritmethic Logic Unit (Unidad Aritmético-lógica).

ANN: Artificial Neural Networks (Redes Neuronales Arificiales).

API: Application Programming Interface (Interfaz de Programación de Aplicaciones).

BP : Backpropagation (Retropropagación).

CBIR: Content Based Image Retrieval (Recuperación de Imágenes Basada en Contenidos)

CI: Computational Intelligence (Inteligencia Computacional).

CPU: Central Processing Unit (Unidad Central de Procesos).

DM: Data Mining (Minería de Datos).

E/S: Entrada / Salida.

FCA: Formal Concept Análisis (Análisis de Conceptos Formales).

FCM : Fuzzy K-means (Media C Borroso – Clasificador -)

FIS : Fuzzy Inference Sistem (Sistema de Inferencia Borrosa).

FL: Fuzzy Logic (Lógica Borrosa).

GA: Genetic Algorithms (Algoritmos Genéticos).

IDF: Inverse Document Frequency (Frecuencia Inversa de Documento).

IE: Information Extraction (Extracción de Información).

IR: Information Retrieval (Recuperación de Información).

HTML: HyperText Markup Language (Lenguaje de Marcado de Hipertexto).

KD: Knowlegde Discovery (Descubrimiento del Conocimiento).

KDD: Knowlegde Discovery in Databases (Descubrimiento del Conocimiento en Bases de Datos).

KNN: K Nearest Neighbour method (método del Vecino K más Próximo).

LSI: Latent Semantic Indexing (Indexado Semántico Latente).

LMI: Linguistically Motivated Indexing (Indexado Motivado Lingüísticamente).

MeSH: Medical Subject Headings (Encabezados de Temas Médicos).

MLP: Multilayer Perceptron (Perceptrón Multicapa).

MT: Machine Translation (Traducción Automática).

NLI: Non-Linguistic Indexing (Indexado No Lingüístico).

NL: Natural Language (Lenguaje Natural).

NLP: Natural Language Processing (Procesado del Lenguaje Natural).

NN: Neural Networks (Redes Neuronales).

ODBC: Open Database Connectivity (Conectividad de Bases de Datos Abiertas).

OEM: Object Exchange Model (Modelo de Intercambio de Datos).

OLAP: Online Analitic Processing (Procesado Analítico En Línea).

POST: Part-of-Speech Tagger (Analizador Gramatical).

QoS: Quality of Service (Calidad de Servicio).

RDBMS: Relational Database Management Systems (Sistemas de Gestión de Bases de Datos Relacionales)

RS: Rough Sets (Conjuntos Aproximados).

SCM: Supply Chain Management (Administración de la Cadena de Suministro).

SQL: Structured Query Language (Lenguaje de Búsqueda Estructurado).

SVM: Support Vector Machine (Máquinas de Soporte Vectorial).

TF: Term Frequency (Frecuencia de Término).

TW: Term Weighting (Asignación de Pesos).

US: Universidad de Sevilla.

VSM: Vector Space Model (Modelo de Espacio Vectorial).

WCM: Web Content Mining (Minería Web de Contenidos).

WM: Minería Web (Web Mining).

WSM: Web Structure Mining (Minería Web de Estructuras).

WSN: Wireless Sensor Networks (Red de Sensores Inalámbrica).

WUM: Web Usage Mining (Minería Web de Uso).

WWW: World Wide Web.

XML: Extensible Markup Language (Lenguaje de Marcas Extensible).