Preguntas de Definición en un Sistema de Búsqueda de Respuestas

Autor: Marcos Álvaro Martín
Tutor: José Luis Martínez Fernández
Directora: Paloma Martínez Fernández

19 de noviembre de 2009

Agradecimientos

A mis padres y mi hermano, por compartir conmigo las ganas de terminar.

A mis primos y mis tíos, por sus ánimos. Especialmente a los molineros, por acogerme en la recta final.

A mis correctoras, por su dedicación altruista.

A mis amigos, por su apoyo.

A mi tutor, por su paciencia.

Resumen

En este proyecto se hace una introducción a los Sistemas de Búsqueda de Respuesta, sistemas cuyo propósito es dar una respuesta concreta a preguntas planteadas por un usuario en lenguaje natural. Se analizan las áreas que influyen en estos sistemas y se detallan los módulos que generalmente lo componen. Posteriormente se analiza la arquitectura del sistema *respond.es,* un SBR sobre el que se diseña e implementa una solución para responder a las preguntas de definición, creando para ello reglas sintácticas y semánticas, atendiendo a los predicados de definición extraídos de la Wikipedia.

Por último se muestra la evaluación del sistema. El sistema ha sido evaluado en el foro CLEF 2008, foro centrado en el desarrollo, evaluación y pruebas de Sistemas de Recuperación de Información. Los resultados son comparados con el resto de participantes en el foro y con los resultados obtenidos por el sistema en CLEF 2007.

La inclusión de las reglas para las preguntas de definición ha conseguido que el sistema pasase de no contestar correctamente a ninguna pregunta a obtener un 73,68 % de respuestas correctas en la evaluación realizada en el foro QA@CLEF 2008. Estos resultados evidencian la necesidad del sistema de reglas específicas para las preguntas de definición y establecen una buena base sobre la que ampliar estas reglas.

Acrónimos

CLEF Cross Lingual Evaluation Foro

EI Extracción de Información

RI Recuperación de Información

PLN Procesamiento de Lenguaje Natural

SBR Sistema de Búsqueda de Respuestas

TREC Text REtrieval Conference

QA Question Answering

XML eXtensible Markup Language

XSLT eXtensible Stylesheet Language Transformations

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción[1]

1.1. Contexto histórico. La información en las redes.

Los primeros ordenadores personales empezaron a comercializarse a finales de la década de los 70. Desde entonces, el número de ordenadores tanto para uso particular como comercial ha crecido a un ritmo vertiginoso. En la actualidad existen más de mil millones de ordenadores en todo el mundo y se estima un crecimiento anual en torno a un doce por ciento. De ser correcta esta estimación se duplicaría la cifra antes de 2014[2]. Para hacerse una idea más concreta del crecimiento, ha llevado casi treinta años llegar a la cifra de mil millones de ordenadores y se espera llegar a los dos mil millones en tan sólo siete años.

Unido al incremento de ordenadores personales, la gran explosión de Internet en la última década -como se puede ver en la Figura 1- ha hecho que la cantidad de información que está al alcance de cualquier usuario conectado a la red

[1]Este trabajo fin de carrera ha sido realizado en el marco del proyecto BRAVO -Busqueda de respuestas Avanzada Multimodal y Multilingüe- TIN2007-67404-C03, en particular en el desarrollo de un sistema de búsqueda de respuestas para español sobre una colección de noticias y de la wikipedia. En este proyecto colaboran el grupo GSI de la UPM, el grupo de Bases de Datos Avanzadas de la UC3M y el Laboratorio de Lingüística Computacional de la UAM.

[2]Estimación realizada por Garner, Inc. http://www.gartner.com/it/page.jsp?id=703807

sea abrumadora. Son cada vez más los usuarios de Internet que participan activamente en la creación de información, y también son cada vez más las facilidades que tienen a su disposición para introducir cualquier contenido en la red, generalmente de forma textual -aunque con el aumento de velocidad de conexión se ha abierto la puerta a todo tipo de contenidos-, desde los iniciales foros hasta los más recientes blogs o *proyectos wiki*[3].

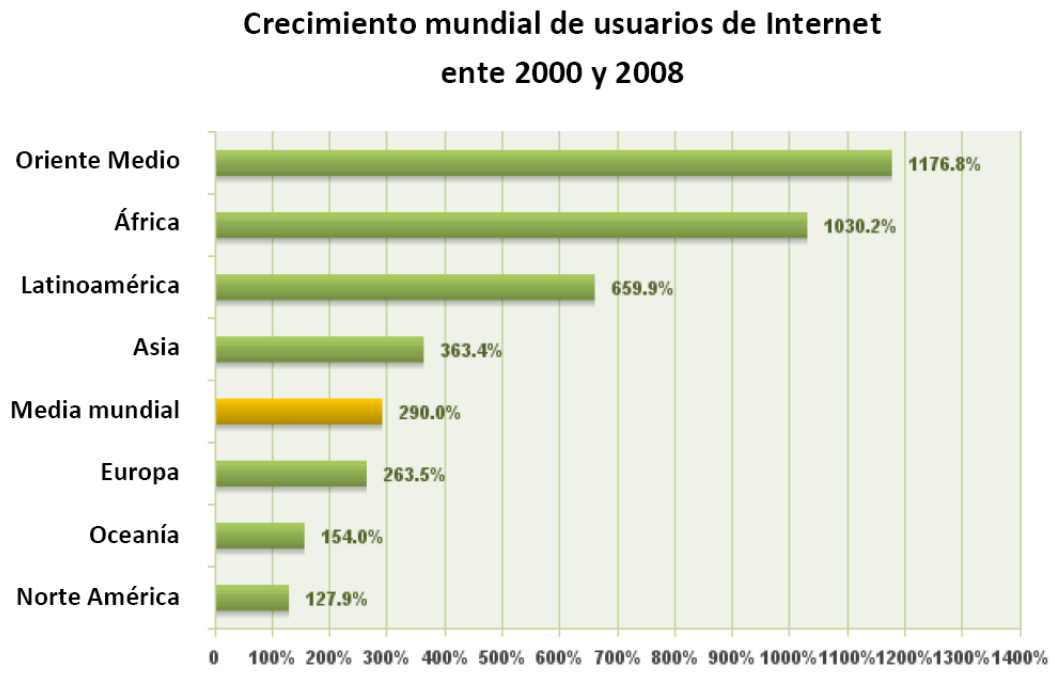

Figura 1.1: Evolución mundial de internautas

Los usuarios de Internet han necesitado de alguna herramienta para alcanzar la información de su interés entre la maraña de contenidos de la red. Desde que Internet se convirtió en la red de comunicación que es hoy en día, se han utilizado para acceder a la información buscadores -como los de Yahoo[4] o Google[5]- basados en sistemas de recuperación de información. Estos sistemas devuelven un conjunto de documentos a una consulta realizada a través de una

[3]Los *proyectos wiki* son nuevos sitios web en los cuales todos los usuarios tienen la capacidad de crear y editar los contenidos.

[4]http://www.yahoo.com

[5]http://www.google.com

lista de términos o palabras clave introducidos por el usuario. Con el gran crecimiento de la información nos encontramos con que la tarea de encontrar los datos que precisamos es cada vez más compleja. De esta necesidad y del afán por mejorar los sistemas que afronten con éxito la tarea de extraer respuestas de grandes volúmenes de información a preguntas concretas nacen los **Sistemas de Búsqueda de Respuestas** (**SBR**).

1.2. Sistemas de búsqueda de respuestas

1.2.1. Modelo de sistema de búsqueda de respuestas

Los SBR se definen como herramientas capaces de obtener respuestas concretas a necesidades de información precisa, utilizando el lenguaje natural para la realización de la consulta. Un SBR ideal sería aquel que respondiese automáticamente a cualquier pregunta dentro del dominio del conocimiento humano.

Se pueden identificar cuatro módulos principales en un SBR [Vicedo et al., 2003]:

1. Análisis de la pregunta

   Las preguntas introducidas al sistema pasan por el primer módulo de *análisis de la pregunta*, donde se realizan tres tareas principalmente: detectar de qué tipo de pregunta se trata -en el siguiente apartado se distinguen los diferentes tipos de preguntas-, detectar el tipo de información que la pregunta espera como respuesta (un lugar, una cantidad, una persona, etc.) y encontrar los términos relevantes de la pregunta que se utilizarán para la extracción de documentos con posibles respuestas.

2. Recuperación de información

   Los términos obtenidos en la fase de análisis de la pregunta sirven de entrada a la fase de *recuperación de información*. El comportamiento de este módulo es similar al de los conocidos buscadores, basado en técnicas de recuperación de información. Generalmente se suele obtener un subconjunto de los documentos fuente, donde se encuentran los términos relevantes de la pregunta. Este subconjunto es una lista ordenada según la relevancia del documento en relación a la consulta realizada. La puntuación obtenida para cada documento depende del modelo usado para recuperar la información, que suele ser generalmente el modelo de espacio de vectores, donde la similitud se mide con la fórmula del coseno [Salton, 1989].

3. Selección de pasajes relevantes

   Obtenidos los documentos candidatos a contener la respuesta se procede a la *selección de los pasajes relevantes,* entendiendo por pasaje un número determinado de frases, que puede ser fijo o variable para adaptarse a diferentes colecciones de documentos [Llopis et al. 2002]. Cada documento es dividido en pasajes, los cuales son ordenados según su relevancia. Esta relevancia se calcula atendiendo al tipo de pregunta y al tipo de información que se espera como respuesta.

4. Extracción de respuestas

   Por último, en la *extracción de respuestas* se analizan los párrafos obtenidos, extrayendo las respuestas y ordenándolas en el caso de que hubiera más de un candidato a respuesta dentro del mismo párrafo.

Las operaciones descritas dentro de los módulos citados se realizan generalmente en dos fases: en linea y fuera de linea. Existen distintos enfoques a la hora de designar las operaciones para cada fase [Lin et al., 2003]. A grandes rasgos, a la fase fuera de linea le corresponden las tareas de preprocesado de los documentos, que puede llevar desde la realización de un índice hasta analizar sintácticamente

el texto, mientras que a la fase en línea le corresponden tareas de análisis de la pregunta, recuperación de documentos y extracción de información. En la Figura 1.2 se puede ver un diagrama donde se representan los módulos descritos y su función dentro del SBR.

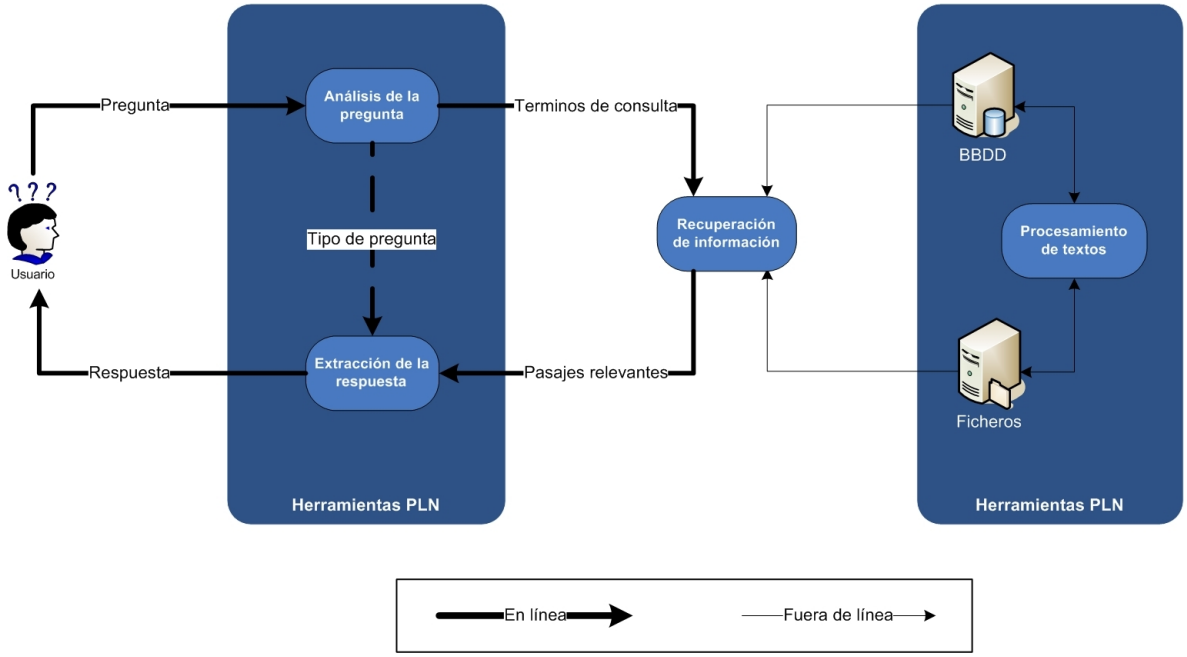

Figura 1.2: Arquitectura de un SBR

1.2.2. Tipos de preguntas

Las preguntas realizadas a un SBR pueden ser muy variadas. Para afrontar su resolución, éstas se suelen dividir en distintos tipos con la finalidad de utilizar distintas estrategias para la resolución de cada una de ellas. Existen varias clasificaciones dependiendo de cada autor [Hernández-Rubio, 2008, Denicia-Carral, 2007], atendiendo a distintos factores. Los tipos más utilizados son: **factual**, **definición**, **lista** y **temporal**.

Las preguntas **factuales** son las relacionadas con datos o hechos concretos, nombres propios, etc., que normalmente pueden ser contestadas rápidamente. Este tipo de preguntas constituyen la mayoría de las consultas.

Las preguntas de **definición**, como su nombre indica responden a la definición a un término,persona, organización, etc...

Las preguntas de tipo **lista** son aquellas que solicitan un cierto número de contestaciones de un mismo tipo.

Las preguntas con restricciones **temporales**, son las preguntas que buscan información restringida a un cierto periodo de tiempo.

En el Cuadro 1.1 se muestran ejemplos de preguntas con sus respectivos tipos. Las preguntas con restricciones temporales están marcadas con una *T*.

Tipo		Pregunta
Definición		¿Qué fue la Revolución de Terciopelo?
Factual		¿En qué colegio estudia Harry Potter?
Definición		¿Qué es la obsidiana?
Factual	T	¿Cuándo nació Amintore Fanfani?
Lista	T	¿Qué países entraron en la Unión Europea el 1 de enero de 1995?
Factual	T	¿Quién era Papa cuando se celebró el Concilio de Clermont?
Lista		¿Quiénes son los miembros fundadores de Star Alliance?

Cuadro 1.1: Ejemplos de tipos de preguntas

1.3. Motivación

Existe una gran cantidad de información en formato digital hoy en día que satisface casi todas las necesidades de información, pero sin métodos apropiados de búsqueda, esta información puede no ser todo lo útil que los usuarios necesitan. Hasta la fecha, la forma más común de buscar información en Internet es a través de motores de búsqueda, que responden de forma eficiente a las consultas cuando se busca información general, pero cuando se intenta responder a preguntas concretas su eficacia disminuye, dejando en manos del usuario la ardua tarea de seleccionar la información de entre los documentos encontrados.

Esta situación ha motivado el desarrollo de nuevos enfoques para recuperar información tales como los SBR.

Este trabajo está enfocado a responder preguntas de definición. Como se ha explicado anteriormente, una pregunta de definición es aquella que tiene como respuesta una frase o conjunto de frases que describen al concepto por el que se pregunta.

Hay que tener en cuenta que existen diferentes formas en las que una definición puede ser introducida en un texto, por lo tanto, la tarea de extraerla no es trivial. Este proyecto pretende estudiar las diferentes propuestas de resolución para las preguntas de tipo definición en español e implementar una solución, tomando como punto de partida el SBR *respond.es* desarrollado en Daedalus.

1.4. Objetivos

Los objetivos de este proyecto se resumen en tres puntos:

- Estudio y documentación del SBR respond.es desarrollado en Daedalus.
- Diseño e implementación de una estrategia para responder a las preguntas de definición.
- Evaluación y comparación de los resultados de distintos SBR, centrado en las preguntas de definición.

Para realizar el proyecto se parte del sistema respond.es realizado por Daedalus. Se estudiará su arquitectura para posteriormente implementar una estrategia para las preguntas de definición. El resultado será comparado con otros sistemas para medir su eficacia y evaluado en el foro CLEF[6] 2008.

[6] *Cross-Language Evaluation Forum* (CLEF) . Foro centrado en el desarrollo, evaluación y pruebas de Sistemas de Recuperación de Información

1.5. Estructura de la memoria

Este primer capítulo introduce el concepto de los SBRs y lo contextualiza, a continuación se describe el estado actual en el que se encuentran los SBR y las áreas de estudio que lo abarcan. En el capítulo 3 se aborda el análisis del sistema, donde se detalla la estructura de un SBR y las necesidades para la construcción de una solución para las preguntas de definición, en el capítulo siguiente se explica el diseño e implementación de dicha solución y por último, en el capítulo 5, se detalla la evaluación de los resultados obtenidos.

Capítulo 2

Estado de la cuestión

2.1. Historia de los SBR

Los SBR surgen de los estudios en el campo de la inteligencia artificial en la década de 1950. Los primeros sistemas creados se basan en el acceso en lenguaje natural a bases de datos. En esta línea se encuentran sistemas como BASEBALL -1961-, centrado en responder preguntas sobre estadísticas de la liga americana de béisbol, LUNAR capaz de responder a preguntas sobre análisis geológico de las piedras lunares recogidas por el transbordador Apolo o LIFER/LADDER, diseñado como una interfaz en lenguaje natural con una base de datos sobre barcos de la marina estadounidense. Estos sistemas tenían un gran índice de acierto, ya que reducían las preguntas a un dominio muy concreto.

En los años posteriores, siguieron implementándose sistemas de búsqueda de respuestas de dominio cerrado, en los que se empezaron a introducir algunas herramientas para la comprensión del lenguaje natural. A finales de los años 70, Wendy Lehnert presenta la primera discusión sobre las características deseables en un SBR [Lehnert, 1977]. En estas se incluían **entender la pregunta del usuario**, **buscar la respuesta en una base de conocimiento**, para después **generar la respuesta y devolverla al usuario** del sistema, con toda la información relacionada posible, es decir, intentado justificar la respuesta. Por lo tanto, dichos sistemas deberían integrar técnicas para el entendimiento del

lenguaje natural, búsqueda de conocimiento y generación de lenguaje natural. Dado que la investigación en BR comenzó como objeto de estudio de la Inteligencia Artificial, se consideraba como requisito que los SBR cumplieran con las características descritas por Lehnert. Sin embargo, estos intentos sólo han obtenido resultados parciales restringiendo en gran medida sus dominios de aplicación. En los últimos años, la investigación en BR ha sido retomada por las comunidades de investigación en recuperación de información, pretendiendo ampliar el dominio sobre el que se realizan las preguntas, con el objetivo de abarcar el mayor ámbito posible para las respuestas. Esto implica un importante aumento de la complejidad de los sistemas, incorporando incrementalmente herramientas más complejas que doten paulatinamente a los SBR con las características deseables descritas por Lehnert. Se puede pues, diferenciar entre dos grandes lineas de investigación de los SBR : SBR de dominio cerrado ó restringido, y SBR de dominio abierto o sin restricción.

En la década de los noventa, el desarrollo en el campo de la lingüística computacional, se ve reflejado en nuevos sistemas que aplican técnicas de procesado del lenguaje natural. Ejemplos de estos sistemas son el Unix Consultant, una herramienta de consulta en lenguaje natural para el sistema operativo UNIX que constaba con una base de datos de conocimiento comprensible y LILOG, un sistema reconocedor de texto que operaba en el dominio del turismo en ciudades alemanas.

Estos primeros sistemas estaban basados en una representación estructurada del conocimiento necesario para responder las preguntas, a diferencia de la investigación y los SBR actuales, cuyo objetivo es tratar textos completamente no estructurados.

Como se detallará en el siguiente apartado, en 1992 surgió la Conferencia de Recuperación de Texto (TREC por sus siglas en inglés) para apoyar la investigación de la recuperación de información. A finales de los 90 TREC incluyó tareas de búsqueda de respuestas que siguen realizándose hoy en día. Los sistemas que participan en esta competición deben responder cuestiones sobre un tema buscando un trozo de texto que varía de un año para otro. Esta competición encaminó la búsqueda y desarrollo de los SBR en dominio abierto.

Un creciente número de sistemas incluyen la web como uno de los cuerpos de texto. Actualmente ha crecido el interés de la integración de los SBR en la Web.

La necesidad de contar con SBR para lenguas diferentes al inglés, e incluso SBR multilingües, dio lugar a la creación de un foro especializado para la promoción de la investigación y la evaluación de SBR con esta finalidad. Así, en el año 2003 se incluyó por primera vez la evaluación de SBR para lenguajes europeos (diferentes al inglés) como parte del foro de evaluación CLEF [Magnini et al., 2003].

2.2. Fundamentos de los SBR

Por todo lo mencionado en el apartado anterior los SBR se pueden enmarcar en el campo de la recuperación de información, pero no solamente este campo incide en el ámbito de estudio de éstos, las áreas relacionadas sobre las que se apoyan los SBR son: **recuperación de información**, **extracción de información** y **procesamiento del lenguaje natural**.

2.2.1. Recuperación de información

La recuperación de información (RI) es la ciencia de la búsqueda de información en documentos, búsqueda de los mismos documentos, la búsqueda de metadatos que describan documentos, y búsqueda en bases de datos -ya sea en archivos locales, en una red local, o en Internet-, para textos, imágenes, sonido o cualquier otro tipo de datos.

La RI es una ciencia interdisciplinar que abarca una gran cantidad de campos: la psicología cognitiva, estadística, matemáticas, la arquitectura de la información, diseño de la información, el comportamiento humano hacia la información, la lingüística, la semiología, la informática o la biblioteconomía.

Los buscadores, tales como Google, Lycos o Yahoo, son los ejemplos más conocidos de sistemas de RI. Básicamente es un proceso donde se accede a una información previamente estructurada y almacenada, mediante la implementación de funciones de búsqueda específicas. Es necesario determinar un proceso donde se establezcan herramientas de indización y control terminológico y una

base de datos donde estén almacenados los documentos, así como definir el lenguaje de consulta y operadores que soporten la base de datos y, establecer que tipo de operaciones serán permitidas.

Los Sistemas de RI asumen que el contenido de los documentos de la base de datos y las necesidades de información de cada usuario puede expresarse mediante un conjunto de términos, que serán utilizados para comparar los documentos almacenados con la consulta introducida por el usuario.

Existen varias formas de representar los documentos almacenados, basadas en distintos modelos teóricos. Se citan tres de las más conocidas y utilizadas:

El **modelo booleano**, primer modelo utilizado en RI, representa cada documento con los términos que lo componen y a las consultas como un conjunto de términos relacionados a través de los operadores lógicos *and*, *or* y *not*. La recuperación de documentos relevantes se realiza seleccionando el conjunto de documentos asociados a cada término de la pregunta y combinándolos atendiendo a las restricciones de los operadores lógicos para devolver el conjunto resultante.

El **modelo vectorial** formulado inicialmente por G. Salton en los años 70 constituye el modelo de representación más utilizado en sistemas de RI debido a su simplicidad y a su eficiencia. Este modelo representa la consulta y cada documento a través de un vector de n dimensiones cuyos componentes son los términos que aparecen en el texto. El peso de cada término se calcula a partir del valor de discriminación general del término y su frecuencia de aparición. Los vectores de los documentos y el de la pregunta son comparados empleando una función de similitud, de la que se obtiene un valor no booleano por la que se pueden ordenar los resultados. Existen varias modificaciones de este modelo, una de las más utilizadas es el modelo BM25.

La **recuperación de pasajes.** Estos sistemas utilizan los mismos modelos de RI pero se sustituye al documento por el pasaje como unidad elemental de indexación. Un pasaje es una secuencia contigua de texto dentro de un documento y su longitud puede estar definida por diversos factores desde signos de puntuación hasta un número fijo de términos. La eficiencia de estos sistemas es generalmente mayor que los sistemas tradicionales de RI.

La idea de utilizar los ordenadores para recuperar información se popularizó en un artículo de Vannevar Bush en 1945 [Amit, 2001] y los primeros sistemas fueron implementadas en los años 1950 y 1960. En los años noventa, la recuperación de información recibió un empuje gracias a las conferencias TREC centradas en un principio en la investigación de la recuperación de información. Las motivaciones para la creación de esta conferencia fueron:

- Fomentar la investigación en el campo de la RI a través de colecciones de prueba y de grandes colecciones de documentos.
- Aumentar la relación entre industria, educación y el gobierno, de manera que permita el intercambio de investigaciones relacionadas.
- Transformar los productos pertenecientes a investigaciones en productos comerciales que consigan mejorar problemas reales de RI.
- Permitir la disposición de técnicas de evaluación que puedan ser usadas en la industria y en el ámbito académico.
- Realizar una presentación del estado de la investigación y desarrollo en este campo de forma anual.
- Incrementar la transferencia de tecnología y perfeccionamiento de las técnicas de evaluación.

La conferencia sigue realizándose hasta la fecha, y es tomada como referencia en el campo de la RI. Acuden desarrolladores e investigadores de todo el mundo, tanto de universidades como del mundo empresarial. Según datos de TREC tras los primeros años de la realización de los *workshops*, la eficacia de sistemas de RI se incrementó aproximadamente en el doble. Tecnologías desarrolladas por primera vez en TREC se incluyen ahora en muchos de los motores de búsqueda del mundo comercial.

2.2.2. Extracción de información

La Extracción de la información (EI) es un tipo de recuperación de la información cuyo objetivo es extraer automáticamente información estructurada de documentos legibles. Los sistemas EI extraen datos sobre entidades, relaciones y eventos a partir de documentos. Generalmente el tipo de información que se pretende extraer se conoce con anterioridad al diseño y desarrollo del sistema lo que es característico de estos sistemas frente a los Sistemas de Búsqueda de Respuestas. A pesar de esto, las técnicas y módulos de EI como el Reconocimiento y Clasificación de Entidades son frecuentemente usados en los SBR como un paso más que facilita la selección de respuestas.

La importancia de la EI está determinada por la creciente cantidad de información no estructurada disponible (es decir, sin metadatos). Este conocimiento puede hacerse más accesible por medio de patrones o de marcado con etiquetas.

La EI es una tarea muy compleja, que aunque no necesita de una comprensión profunda de los textos sí requiere del uso de herramientas avanzadas para el procesamiento del lenguaje natural.

A continuación se muestra un ejemplo[1] de extracción realizado sobre una noticia ficticia.

En las tablas siguientes se pueden contemplar las entidades y atributos extraídos mediante un SEI, así como los hechos y eventos encontrados.

Personas	Organizaciones	Loc.	Artefactos	Fechas
Fletcher Maddox	UCSD Business School	La Jolla	Geninfo	Junio 1999
Dr. Maddox	La Jolla Genomatics	CA		
Oliver	La Jolla Genomatics			
Oliver	L.J.G			
Ambrose				
Maddox				

Cuadro 2.2: Ejemplo de entidades extraídas por un SEI

[1]Extraído de http://extraccioninformacion.latinowebs.com/

"Fletcher Maddox, jefe de la UCSD Universidad de negocios, anunció la formación de La Jolla Genomatics en conjunto con sus dos hijos. La Jolla Genomatics lanzará su siguiente producto, Geninfo, en Junio de 1999. Geninfo es un sistema para ayudar a investigadores en biotecnología a poder mantenerse al día con la voluminosa literatura que existe en todos los campos del área".
"El Dr. Maddox será el CEO de la compañía. Su hijo, Oliver, será el Científico Jefe, además es propietario de múltiples patentes utilizadas en los algoritmos que contiene Geninfo. El hermano de Oliver, Ambrose, sigue más los pasos de su padre y será el CFO de La Jolla Genomatics, situada en cerca de la ciudad de residencia del Dr. Maddox, La Jolla, California".

Cuadro 2.1: Ejemplo EI

Nombre	Descripción	Entidad
Fletcher Maddox Maddox	jefe de la UCSD Business School su padre CEO de la compañía	Persona
Oliver	Su padre Científico Jefe	Persona
Ambrose	Hermano de Oliver el CFO de L.J.G.	Persona
UCSD Business School		Organización
La Jolla Genomatics L.J.G.		Organización
Geninfo	su producto	Artefacto
La Jolla	ciudad de residencia de la familia Maddox	Localización
California		Localización

Cuadro 2.3: Ejemplo de atributos extraídos por un SEI

Persona		Organización
Fletcher Maddox	Empleado de	UCSD Business School
Fletcher Maddox	Empleado de	La Jolla Genomatics
Oliver	Empleado de	La Jolla Genomatics
Ambrose	Empleado de	La Jolla Genomatics
Artefacto		Organización
Geninfo	Producto de	La Jolla Genomatics
Localización		Organización
La Jolla	Localización de	La Jolla Genomatics
California	Localización de	La Jolla Genomatics

Cuadro 2.4: Ejemplo de hechos extraídos por un SEI

Se puede apreciar en los cuadros, los resultados de la identificación y distinción automática entre las entidades (personas, organizaciones, localizaciones, artefactos y fechas). Este sistema también realiza extracción de descripciones de entidades y relación entre ellas. Un buen motor de Extracción de Información es herramienta básica en los SBR.

2.2.3. Procesamiento del lenguaje natural

El procesamiento del lenguaje natural (PLN) es una disciplina que suele encuadrarse en el ámbito de la inteligencia artificial y la lingüística computacional. Se centra en el estudio de los problemas relacionados con la comunicación en lenguaje natural a través de máquinas, tanto en la interpretación como en la generación del lenguaje. En la última década se ha ido incrementado la cantidad y la complejidad de las técnicas de PLN usadas en los SBR. Estas técnicas se aplican en las etapas de anales de la pregunta y extracción de la respuesta. Se realiza un análisis detallado de la pregunta con la finalidad de encontrar las palabras claves en la búsqueda y el tipo de respuesta esperada. Sobre el conjunto de documentos de los que se extrae la respuesta, suelen utilizarse patrones o clasificadores de entidades.

Dentro de las técnicas de PLN se pueden distinguir varios tipos de análisis:

- **Análisis léxico**. Encargado de transformar cada secuencia de caracteres en una secuencia de unidades significativas. Se utilizan diccionarios, reglas

morfológicas o tesauros para buscar prefijos, sufijos, sinónimos, etc. Las herramientas utilizadas para el análisis léxico son los *tokenizadores* o etiquetadores de texto, así como diccionarios o tesauros.

- **Análisis sintáctico**. Extrae los sintagmas de la oración. Analiza la estructura gramatical de la cadena de unidades léxicas y produce una representación estructurada. Una de las herramientas más comunes que utiliza el análisis sintáctico son los *parser*.

- **Análisis semántico**. Genera una estructura lógica asociada a la creada en el análisis sintáctico, representando el sentido de la sentencia, independientemente del contexto. Una herramienta frecuentemente utilizada que se enmarca en el análisis semántico es WordNet[2] -en inglés- o EuroWordNet[3] -para idiomas europeos-, que constan de una base de datos formada por relaciones semánticas entre los significados de las palabras, agrupadas, entre otras, por relaciones de sinonimia o hiper/hiponimia.

- **Análisis de discurso o contextual**. Encargada de obtener la interpretación final de la oración en función de la estructura semántica y el contexto.

2.3. Arquitectura genérica de un SBR

Una vez obtenida la idea general del comportamiento de los SBR y las bases en las que se fundamenta, en esta sección se presenta una descripción de los módulos que generalmente los componen: Análisis de la Pregunta, Recuperación de la Información, Selección de Pasajes Relevantes y Extracción de Respuestas. Se ejemplifica mostrando el comportamiento de cada módulo durante la resolución a la pregunta: *¿Quién es Bill Clinton?*.

[2] http://wordnet.princeton.edu/
[3] http://www.illc.uva.nl/EuroWordNed/

2.3.1. Análisis de la Pregunta

Las dos tareas principales del módulo de Análisis de la Pregunta son: extraer los términos clave para realizar la consulta sobre el sistema de recuperación de información y determinar el tipo de pregunta. Los términos clave serán utilizados en la etapa de recuperación de información para recuperar los documentos relacionados, mientras que el tipo de pregunta es utilizado en la selección de pasajes relevantes y/o en la extracción de respuestas para aplicar distintas estrategias de resolución.

Además de estas tareas hay otros datos que pueden ser extraídos al analizar la pregunta: el tipo de respuesta esperada, el foco y el tema.

El foco es una palabra o conjunto de palabras que perteneciendo a la pregunta, están muy relacionadas con el tipo de entidad de la respuesta. El tema se puede definir como la "materia" de la que trata la pregunta. En definitiva, el tema es *dónde* buscar y el foco *qué* buscar. Por ejemplo, para la pregunta: "¿Qué actor hace el papel de Chema en la película Tesis?" el tema sería Tesis (o película_Tesis) mientras que el foco sería actor.

Cada implementación identifica unos tipos de pregunta, dependiendo de los sistemas de análisis de que se dispongan y de los criterios de cada desarrollador. Cuantos más tipos se diferencien, más específicos serán estos, y mayor va a ser la reducción del campo de búsqueda de la respuesta.

La formación de la consulta, que se utilizará para la recuperación de información, suele constar de dos métodos: la lematización y la expansión. Con la lematización se obtiene la forma canónica de la palabra, que es sustituida por la palabra original, mientras que con la expansión se buscan sinónimos que serán añadidos a la búsqueda. Existen también palabras que son consideradas superfluas para la búsqueda, este conjunto de palabras se denomina *stopwords* o palabras de parada.

Otros aspectos que pueden tratarse de manera especial son los nombres compuestos, los acrónimos y las fechas.

Pregunta: ¿Quién es Bill Clinton?
Tipo de Pregunta Definición
Lemas Quién ▷ ser ▷ Bill_Clinton
Lista No
Restricciones Temporales No
Foco Bill_Clinton
Términos de Consulta Bill_Clinton

Cuadro 2.5: Ejemplo de Análisis de la Pregunta

En el cuadro 2.5 se muestra un posible análisis a la pregunta *"¿Quién es Bill Clinton?"*. Como se puede ver, se ha clasificado como una pregunta de definición y se ha extraído como término de consulta *Bill_Clinton,* pese a que los lemas devueltos son tres (*quién, ser* y *Bill_Clinton*) sólo se ha incluido este término debido a que el resto de lemas están incluidos dentro de la lista de palabras de parada.

2.3.2. Recuperación de la Información

Los términos obtenidos en la fase de análisis de la pregunta sirven de entrada a la fase de Recuperación de Información. El comportamiento de este módulo es similar al de los conocidos buscadores. Generalmente se suele obtener un subconjunto de los documentos fuente, donde se encuentra los términos de consulta. Este subconjunto es una lista ordenada según la relevancia del documento en relación con la consulta realizada. La puntuación obtenida para cada documento depende de la técnica de recuperación de información, que suele ser generalmente el modelo del coseno.

Una API de código abierto ampliamente utilizada en la recuperación de información es Lucene[4], con funciones encargadas del indexado y la búsqueda. La principal ventaja de Lucene es la independencia del formato del fichero, lo que permite realizar una búsqueda sobre una colección de textos sin que estén homogeneizados previamente. La fase de indexación es una de las tareas que hay que realizar previamente fuera de linea.

En el ejemplo mostrado en el cuadro 2.6 se pueden ver los documentos (extraídos de la Wikipedia) devueltos por el sistema. El orden de los documentos suele atenerse a la frecuencia de repetición de los términos de consulta con respecto a la longitud del texto.

Pregunta: ¿Quién es Bill Clinton?

Términos de Consulta Bill_Clinton

Documentos con posibles respuestas

1. Bill_Clinton
2. Hipertimia
3. Janet_Reno
4. Monica_Lewinsky
5. Hillary_Rodham_Clinton

Cuadro 2.6: Ejemplo de Recuperación de Información

2.3.3. Selección de Pasajes Relevantes

Esta fase está fuertemente vinculada a la Recuperación de Información y a menudo es considerada una subetapa dentro de ésta. La recuperación de información hace de un primer filtro quitando los documentos menos relevantes.

[4] http://lucene.apache.org/

Sobre este subconjunto de documentos sigue siendo muy costoso aplicar técnicas que extraigan directamente la respuesta, por ello, los documentos son divididos en pasajes que vuelven a ser filtrados para seleccionar los que contengan información relevante. Como se ha visto en el apartado anterior, para la recuperación de información sólo se utilizan las *palabras clave* mientras que en la selección de pasajes se introducen otros campos relacionados con el contexto de la pregunta, como pueden ser el tipo de pregunta o al tipo de respuesta esperada.

Cada documento es dividido en pasajes, los cuales son ordenados según la relevancia obtenida durante la recuperación de información, descartando aquellos que no tienen ninguna relación con la pregunta planteada. La relevancia de las frases se comprueba puntuando cada una de ellas proporcionalmente al número de palabras (o expresiones) contenidas, que coincidan con las pertenecientes a la pregunta, siguiendo reglas que tengan en cuenta el tipo de respuesta esperada.

Se puede observar en el ejemplo mostrado (2.7) que no todas las frases contienen expresiones pertenecientes a la pregunta, si no que han seguido otras reglas para la selección de frases. Estas pueden venir dadas por patrones o por la inclusión de etiquetas semánticas.

Pregunta: ¿Quién es Bill Clinton?

Términos de Consulta Bill_Clinton

Pasajes Seleccionados

1. Bill Clinton, cuadragésimo segundo presidente de los Estados Unidos ; Hillary Clinton, Senadora por el estado de Nueva York y precandidata a la Presidencia estadounidense ; Clinton, localidad del estado de Nueva York ; Clinton, localidad del estado de Oklahoma ., William Jefferson " Bill" Clinton (n. William Jefferson Blythe III el 19 de agosto de 1946) fue el cuadragésimo segundo Presidente de los Estados Unidos y gobernador del estado de Arkansas.

2. Su madre, Virginia Dell Cassidy (1923 - 1994), se volvió a casar con Roger Clinton, apellido que adquirió a los 15 años., Su padre biológico murió cuando su madre estaba embarazada de Bill.

3. Era la primera vez desde la primera victoria electoral de Richard Nixon en 1968 que un candidato ganaba la elección presidencial sin la mayoría absoluta del voto popular ; o sea, sin llegar al menos al 50 % de los sufragios .

4. Clinton fue gobernador de Arkansas entre 1978 y 1992 .

5. Pero aún así Clinton era Presidente electo con una clara ventaja sobre Bush , que había fracasado en su aspiración reeleccionista .

6. Algunos consideran que fue un presidente moderado y que la economía de los Estados Unidos experimentó una fuerte alza durante su presidencia .

7. A principios del año 1992 Clinton era uno de los 10 Pre-candidatos presidenciales que competían por la Candidatura oficial del Partido Demócrata ; las elecciones primarias internas fueron muy reñidas y Clinton no partía como favorito.

8. Los puntos a favor de Clinton también fueron su juventud y carisma, en una lucha generacional contra un Bush más viejo (veterano de la Segunda Guerra Mundial) y poco carismático.

Cuadro 2.7: Ejemplo de Selección de Pasajes

2.3.4. Extracción de Respuestas

Por último, en *extracción de respuestas* se analizan los párrafos obtenidos extrayendo las respuestas y ordenándolas en el caso de que hubiera más de un candidato a respuesta dentro del mismo párrafo.

Primero se realiza una selección de las respuestas candidatas. Los pasajes que se han obtenido en la fase anterior son analizados y se extraen los candidatos, atendiendo a las reglas, patrones y/o etiquetas semánticas. Las respuestas candidatas son nuevamente analizadas, las que son iguales o equivalentes (por ejemplo: "Cristóbal Colón" y "Cristóforo Colombo") se unifican. El número de veces que una misma candidata se repite puede ser un dato significativo, a la hora de puntuarlas. Después se puntúa la posible respuesta en función del parecido a la pregunta; en función del número de palabras que coincidan, de su orden, proximidad o cualquier otro parámetro que se le pueda ocurrir al desarrollador. Las respuestas posibles se ordenan en una lista en función del valor de respuesta obtenido, mostrándose al usuario junto con el párrafo y el documento fuente.

Las etapas de detección de respuestas posibles y de valoración de dichas respuestas utilizan diferentes métodos para su realización en función del tipo de pregunta y de la clase a la que pertenece el tipo de respuesta esperado.

Algunos sistemas incluyen un módulo final para comprobar la fiabilidad de las respuestas, validando las respuestas. Para ello se utiliza otro método de extracción y se comparan los resultados.

En el cuadro 2.8 se muestra la salida al ejemplo realizado, con la respuestas, el texto de soporte y el documento fuente.

Pregunta: ¿Quién es Bill Clinton?

Respuestas Ordenadas

1. **Respuesta**: fue el cuadragésimo segundo Presidente de los Estados Unidos y gobernador del estado de Arkansas

   Texto Soporte: Bill Clinton, cuadragésimo segundo presidente de los Estados Unidos.

   Documento: Clinton

   Texto Soporte: William Jefferson " Bill" Clinton (n. William Jefferson Blythe III el 19 de agosto de 1946) fue el cuadragésimo segundo Presidente de los Estados Unidos y gobernador del estado de Arkansas.

   Documento: Bill Clinton

2. **Respuesta**: padre biológico murió cuando su madre estaba embarazada de Bill

   Texto Soporte: Su madre, Virginia Dell Cassidy (1923 - 1994), se volvió a casar con Roger Clinton, apellido que adquirió a los 15 años.

   Documento: Bill Clinton

   Texto Soporte: Su padre biológico murió cuando su madre estaba embarazada de Bill.

   Documento: Bill Clinton

3. **Respuesta**: fue gobernador de Arkansas entre 1978 y 1992

   Texto Soporte: Clinton fue gobernador de Arkansas entre 1978 y 1992.

   Documento:Bill Clinton

4. **Respuesta**: presidente moderado y que la economía de los Estados Unidos experimentó una fuerte alza durante su presidencia.

   Texto Soporte: Algunos consideran que fue un presidente moderado y que la economía de los Estados Unidos experimentó una fuerte alza durante su presidencia.

   Documento:Bill Clinton

Cuadro 2.8: Ejemplo de Extracción de Respuestas

2.4. Preguntas de definición

2.4.1. Distinguiendo preguntas de definición

En el capítulo introductorio se hace mención a los tipos en los que se clasifican las preguntas con la finalidad de establecer distintas estrategias para su resolución, no obstante, se incide en este apartado en la descripción de las preguntas de definición, tema sobre el que se centra este proyecto.

Para aclarar qué es una pregunta de definición es necesario especificar qué se entiende por definición. Según la Real Academia Española de la lengua una definición es la *"Proposición que expone con claridad y exactitud los caracteres genéricos y diferenciales de algo material o inmaterial"*.

Sin embargo, un concepto puede ser definido de diferentes formas, las cuales dependen del contexto en el que es utilizado, la intención, la facilidad de comprensión deseada, el público al que va dirigido, etc. Las preguntas de definición en el contexto de SBR, están dirigidas a responder preguntas simples que dependen de diferentes factores, tales como la intención del usuario o la colección de documentos usada.

A continuación se presentan las formas de evaluación de las preguntas de definición en los foros TREC y CLEF.

En el TREC las preguntas de definición tienen como respuesta un conjunto de fragmentos que cubren características esenciales del concepto que debe ser definido. El problema principal de evaluar las preguntas de definición de esta forma es determinar cual característica es esencial y cual no. En el TREC la forma de determinarlo está sujeta al criterio de las personas encargadas de evaluar los SBR, lo cual hace que este proceso sea muy complicado y subjetivo si las respuestas son evaluadas solamente por una persona.

En el foro CLEF, a diferencia del TREC, la respuesta a una pregunta de definición es una frase que describe una característica importante del concepto, pero que debe ser respaldada por un fragmento de texto que incluye al concepto y dicha descripción. En el CLEF las preguntas de definición son de distintos

tipos. Existen las definiciones de tipo organización, con preguntas como *¿Qué es la OTAN?* en donde la respuesta para la primera pregunta es simplemente la equivalencia de la sigla con su significado, es decir, "Organización del Tratado del Atlántico Norte". Esta repuesta, desde el punto de vista del foro CLEF, proporciona una característica importante del concepto. Otro tipo de preguntas es el de definiciones de personas, en las que la respuesta esperada es el cargo o rol que desempeña o ha desempeñado una persona. Por ejemplo para la pregunta *¿Quién es Alan Turing?* la respuesta correcta podría ser "uno de los padres de la Ciencia de la computación" o más específicamente "matemático, informático teórico, criptógrafo y filósofo inglés.". Como puede observarse estás respuestas dan características importantes del concepto. Además de estos dos tipos de preguntas, se hacen referencia a cosas, por ejemplo *¿Qué es el hipocausto?* en donde la respuesta correcta es "sistema de calefacción del suelo utilizado sobre todo en las termas del Imperio Romano". A diferencia del TREC la respuesta es una frase y no pequeños fragmentos de información. Además los usuarios de los SBR lo que esperan como una definición es una respuesta que les proporcione información fundamental sobre el concepto por el que preguntan y que los ayude a entender su uso en el contexto de búsqueda.

2.4.2. Distintas estrategias

Dependiendo del dominio la forma de afrontar el problema es muy diferente. Cuando el SBR abarca un dominio cerrado es más sencillo estudiar los distintos casos en los que se puede presentar la definición del término, incluso es posible crear una herramienta automática para examinar como se presenta cada termino y su definición y crear patrones. En un dominio abierto la solución es más compleja y más dependiente del idioma usado, es necesario adentrarse más profundamente en el campo de la lingüística para dar con las respuestas, dado que existen muchas formas en las cuales un concepto puede ser descrito en lenguaje natural, obtener un conjunto completo de patrones lingüísticos para resolver el problema es una labor casi imposible.

Algunas de las estrategias a la hora de afrontar la resolución de las preguntas

de definición:

- **Patrones manuales**: Consiste en la realización de patrones por parte de un experto. Éste es el encargado de crear, depurar y mantener las reglas que se van a usar en las diferentes etapas de extracción de información. En general el experto debe estar familiarizado con el dominio, tener conocimientos de lingüística y estar familiarizado con los formalismos que se vayan a usar para la especificación de las reglas.

- **Aprendizaje automático**: Existen técnicas para la extracción de patrones de forma automática a partir de una colección de textos mediante algoritmos. El algoritmo produce una función que establece una correspondencia entre las entradas y las salidas deseadas del sistema. La base de conocimiento del sistema está formada por ejemplos de etiquetados anteriores. Para su utilización es necesario anotar un conjunto de documentos con la información que se desea extraer y generar un corpus anotado. A partir de este corpus se obtienen los conjuntos de entrenamiento. El gran número de características y los requisitos de los algoritmos generalmente exigen que el tamaño del corpus anotado sea grande. Una alternativa para reducir este coste consiste en usar texto sin anotar para complementar y reducir el tamaño del corpus anotado. A este tipo de aprendizaje se le conoce como semisupervisado o parcialmente supervisado.

- **Enfoque predictivo**: Este método pretende anotar en toda colección las posibles piezas de información susceptibles a ser respuesta de una pregunta cualquiera y almacenarlo en una base de datos. Es compatible con las dos estrategias anteriores.

Capítulo 3

Análisis del Sistema

3.1. Determinación del Alcance del Sistema

Esta fase del proyecto tiene como objetivo obtener una especificación del sistema respond.es, un SBR desarrollado en colaboración entre Daedalus y LaBDA[1], sobre el que se va a implementar una solución para las preguntas de definición. También se analizan los distintos casos de uso, así como la estrategia utilizada para las preguntas de definición y el dominio de búsqueda del sistema.

Se especifican los siguientes puntos respecto al alcance del sistema:

- El objetivo global del sistema es responder a preguntas en lenguaje natural, realizadas en lenguaje español.
- Las preguntas serán clasificadas según su tipo y resueltas en función de éste.
- Se mostrará al usuario la respuesta junto al párrafo y al documento de los que ha sido extraída.
- El sistema está dividido en subsistemas.

[1]Grupo de Bases de Datos Avanzadas de la Universidad Carlos III de Madrid

- El sistema debe ofrecer un bajo acoplamiento, alta modularidad y capacidad de reutilización.
- Las preguntas realizadas serán almacenadas.
- El tiempo de respuestas deberá ser mínimo.

3.2. Arquitectura del sistema respond.es

3.2.1. Identificación de Subsistemas

En este apartado, se descompone el sistema de información en diferentes subsistemas de menor tamaño para facilitar la tarea de análisis. Se puede ver en la siguiente figura la descomposición del sistema respond.es. Cada subsistema se corresponde con un paquete.

Figura 3.1: Especificación de Subsistemas

En los siguientes apartados se analiza cada uno de los subsistemas. Se muestra primero, la relación del subsistema con el resto de subsistemas y a continuación el diagrama de clases de las clases integrantes de cada uno de los subsistemas. Los diagramas mostrados contienen únicamente el nombre de las clases y sus relaciones, en el Anexo A se muestran los diagramas completos.

Quedan fuera del alcance de este proyecto, los subsitemas de Tratamiento Temporal y Preguntas Correferentes, por lo que no son analizados.

3.2.2. Análisis de la Pregunta

El paquete *questionanalysis* es el encargado de realizar las tareas relacionadas con el análisis de la pregunta.

A continuación se muestra los paquetes relacionados con el análisis de la pregunta. Los subsitemas de analisis lingüístico y *tokenizador* son utilizados para analizar sintáctica y semánticamente la pregunta. Se utiliza también el subsistema de extracción de la respuesta, ya que las respuestas son almacenadas en la propia estructura de la pregunta.

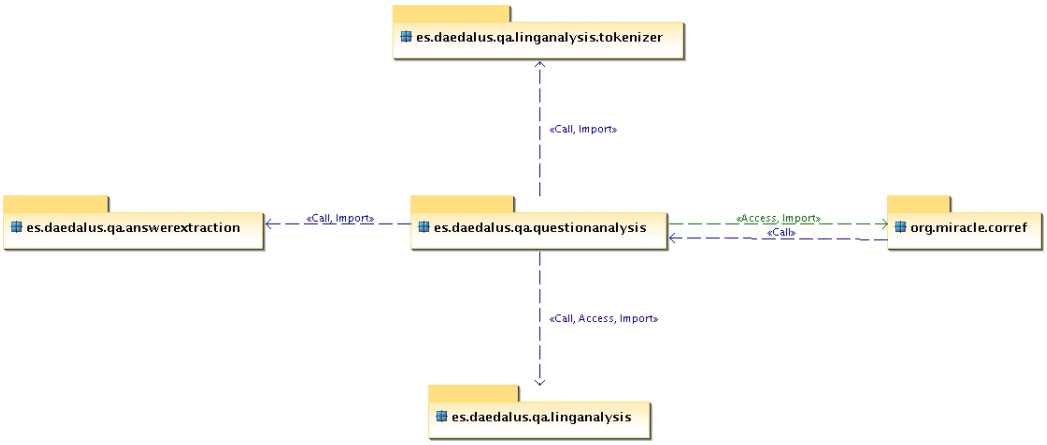

Figura 3.2: Dependencia de paquetes *questionanalysis*

El diagrama se muestra dividido en dos partes para una mejor apreciación de las clases y sus relaciones.

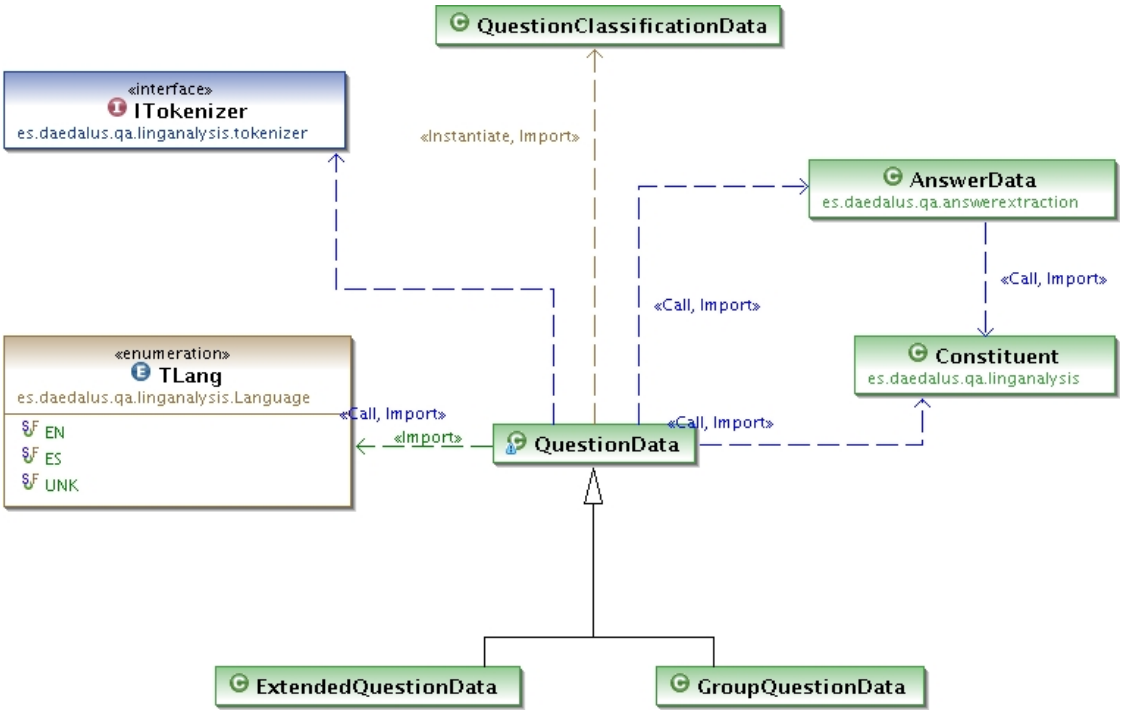

Figura 3.3: Diagrama de clases *questionanalysis* 1/2

QuestionData define la información de que se dispone para una pregunta dada. Se incluye tanto el texto original de la pregunta como su análisis lingüístico, así como las características de la pregunta ofrecidas por el clasificador. En esta misma estructura se almacenará las respuestas que se obtengan a la pregunta.

La clase Constituent es utilizada para almacenar la pregunta analizada. Esta clase identifica cualquier componente de una frase, ya sea un token, sintagma o la propia frase.

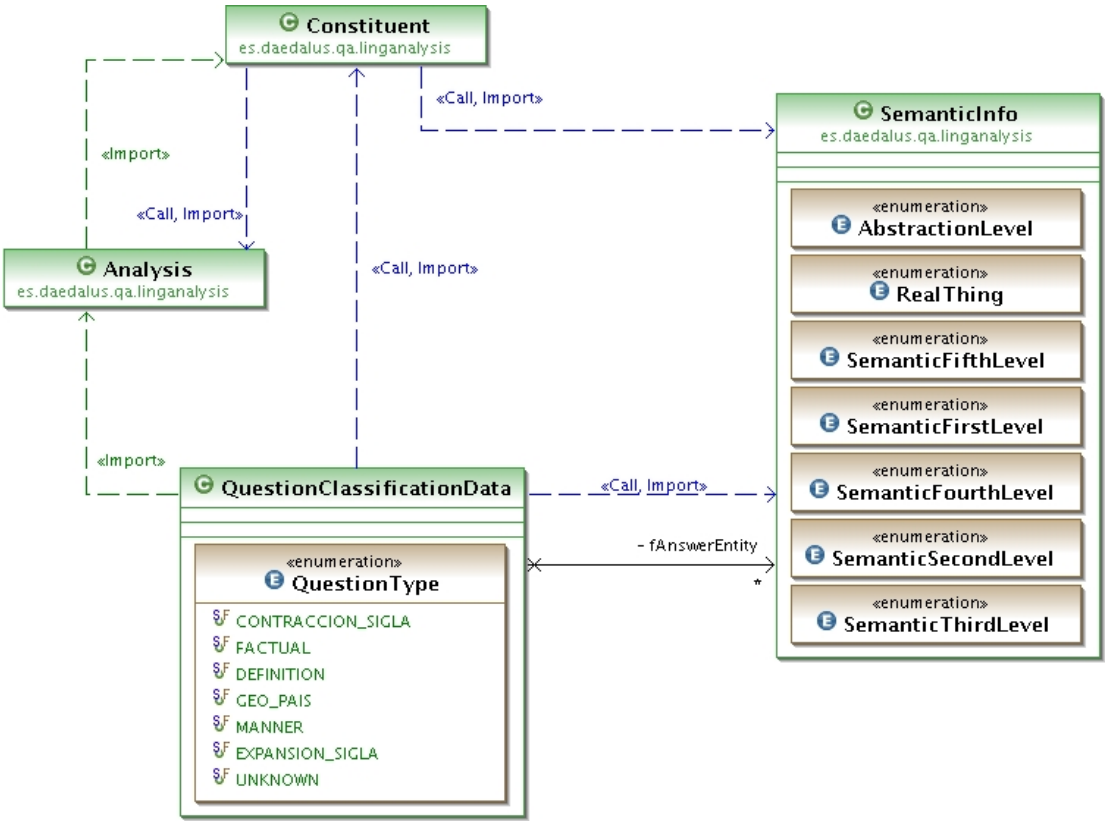

Figura 3.4: Diagrama de clases *questionanalysis* 2/2

QuestionClasificationData define los componentes necesarios para la clasificación de la pregunta, contiene la clase QuestionType, un enumerado con los distintos tipos en los que puede ser clasificada una pregunta. Además contiene datos relativos a la pregunta, como puede ser si es una lista, el foco, el tema, las restricciones temporales, la entidad de la respuesta esperada y otros factores relativos a la cuestión. Utiliza la clase Analysis para indicar las restricciones temporales y SemanticInfo para indicar la entidad a la que debe pertenecer la respuesta que dará el sistema a la pregunta.

3.2.3. Documentos Fuente

El paquete *answersoruce* es el encargado de tratar la fuente que contiene la documentación utilizada para extraer las preguntas.

En el se define la información de que se va a disponer para un documento concreto, así como la forma de crear un índice y realizar la búsqueda sobre los documentos. El objetivo es crear el índice de forma independiente a la estructura de documentos, para poder realizar búsquedas sobre distintas colecciones de documentos de manera uniforme.

Para realizar las funciones descritas anteriormente se utilizan clases de otros paquetes. Las clases pertenecientes al Análisis Lingüístico son utilizadas para realizar el análisis del documento. El paquete de Extracción de la respuesta, es utilizado en el caso de que la respuesta sea conocida de antemano, como puede ser el caso de los acrónimos, para los cuales el sistema tiene la respuesta. En ese caso sólo será necesario encontrar documentos para justificar la respuesta. El paquete de Análisis de la pregunta es importado, y utilizado un objeto QuestionData que se usa para crear la consulta. También es almacenado en dicho objeto el número de documentos con posibles respuestas a la pregunta.

Esta relación entre distintos paquetes se puede ver en el siguiente diagrama.

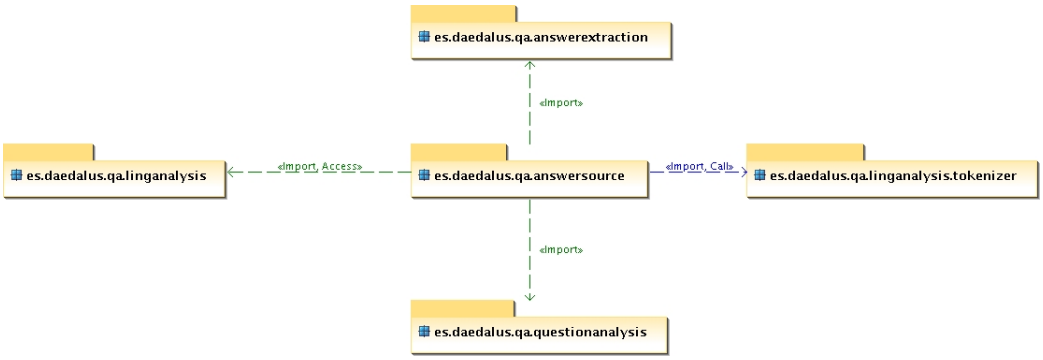

Figura 3.5: Dependencia de paquetes *answersource*

En el diagrama de clases mostrado en la Figura A.3 puede contemplarse las asociaciones y dependencias de las clases pertenecientes al paquete *answersource*. La clase QADocument define la información de que se va a disponer para un documento concreto. Esta será la información que deberá proporcionar cualquier sistema que acceda a una fuente de información a buscar frases de texto en las que puedan aparecer una respuesta a una pregunta. Es decir, un objeto de tipo Document puede ser cualquier porción de texto de una fuente

que pueda contener una respuesta y que pueda ser identificado y accedido de forma independiente. Los objetos Document se crearán sin contenido textual, será el método *analyze* el encargado de incluir la información con el texto del documento y su análisis lingüístico. Dicho método analiza el contenido textual del documento, almacenando el resultado en el propio objeto Document. Si el documento ya había sido analizado y el resultado de dicho análisis almacenado, se cargará de nuevo ese análisis, sin necesidad de volver a efectuarlo.

IAnswerSource define el interfaz necesario para cada fuente de respuestas a integrar en el sistema.

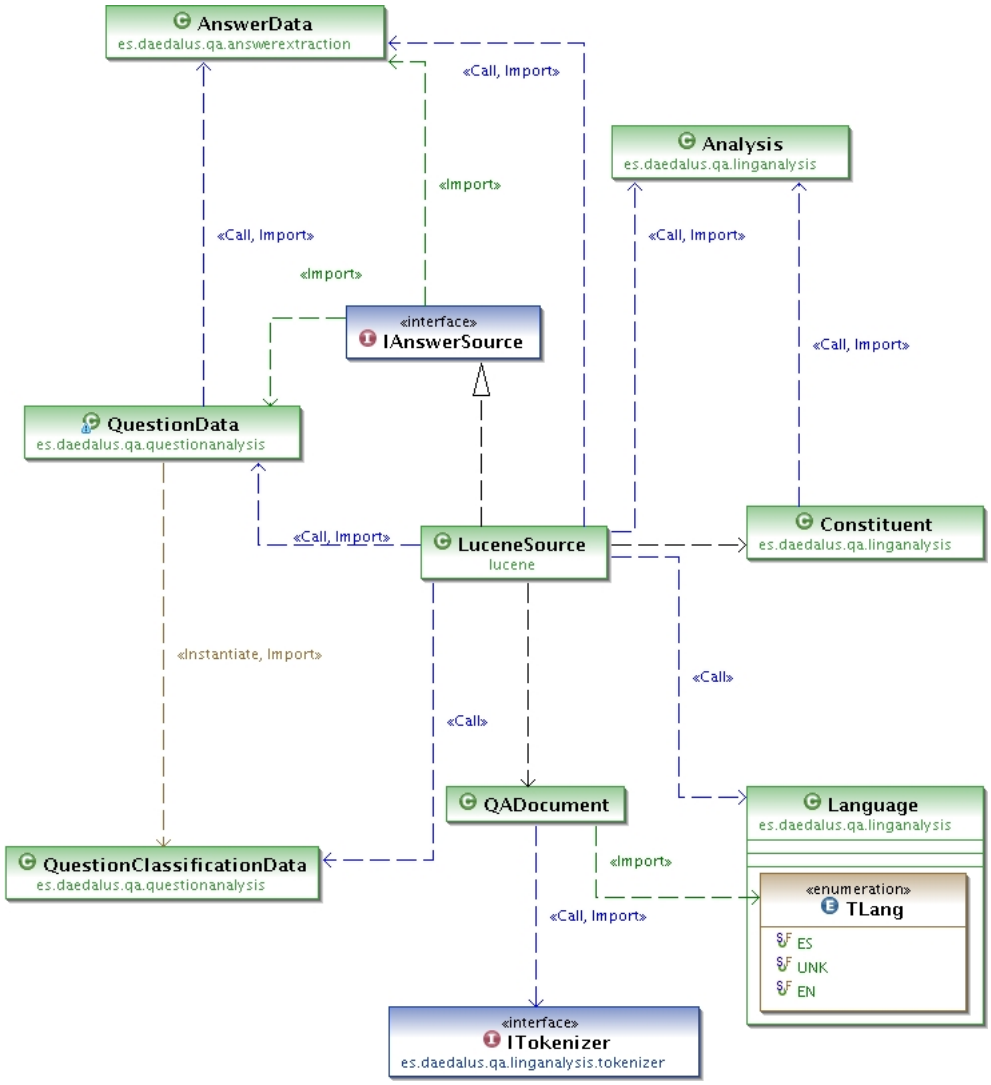

Figura 3.6: Diagrama de clases *answersource*

Dentro del paquete *answersource* se encuentra el subpaquete *lucene,* donde se especifica la forma de crear índices y realizar búsquedas utilizando la API Lucene de Apache. Esta API permite realizar una búsqueda sobre una colección de textos sin que estén homogeneizados previamente. Para ello se definen campos -de la clase *Field* de lucene- para cada documento. El sistema respond.es cuenta con dos colecciones: los artículos de la Wikipedia y artículos de prensa de la agencia EFE. Para cada documento, independientemente de la colección a la que pertenezca se definen los siguientes campos:

- Título: Título del artículo tanto para la colección de Wikipedia como para EFE.
- Subtitulo: Sólo para los documentos de EFE, algunas noticias tienen una subcabecera. Para la colección de Wikipedia se deja en blanco.
- Nombre de la colección: Nombre identificativo de cada colección. Actualmente son ”wiki” para la Wikipedia y ”efe” para la colección de artículos EFE.
- Ruta del documento: Path completo indicando donde está almacenado el documento.
- Idioma: Español o inglés. Para este proyecto sólo se abarcará el sistema en español.
- Texto: Cuerpo de la noticia o artículo.

Los paquetes importados por el subpaquete lucene, se pueden ver en la siguiente figura.

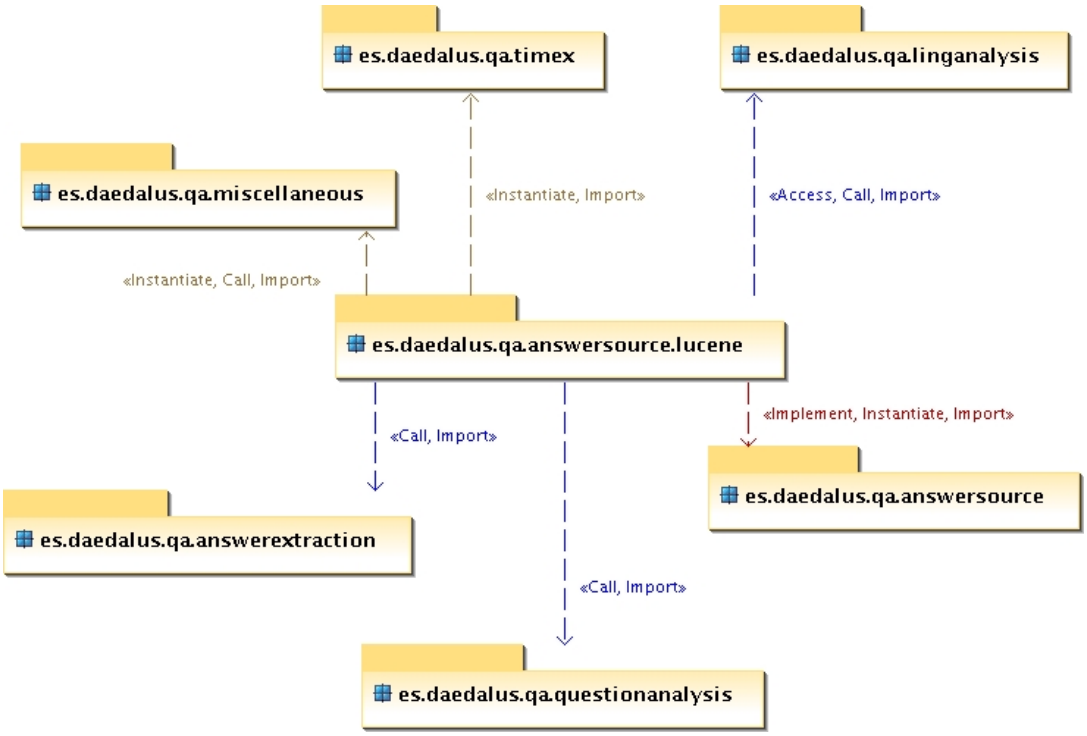

Figura 3.7: Dependencia de paquetes *lucene*

En el diagrama de clases de la Figura A.4, se muestra las clases que forman el paquete *lucene*. En éste se define un interfaz para crear manejadores para los documentos para Lucene, y dos clases que implementan esta interfaz, una para los documentos de Wikipedia y otra para los de EFE.

LuceneSource implementa las funciones de búsqueda necesarias para localizar entradas en un índice de Lucene. Implementa la interfaz IAnswerSource, que especifica las funciones de búsqueda que deben proporcionarse sobre una fuente de información. Contiene los métodos para generar la consulta -a partir de un objeto QuestionData- y para realizar la búsqueda sobre el índice, utilizando dicha consulta. La clase LuceneIndex, indiza colecciones de documentos usando Lucene. También se define la herramienta LuceneSearchExe para realizar consultas a través de línea de comandos en índices de Lucene.

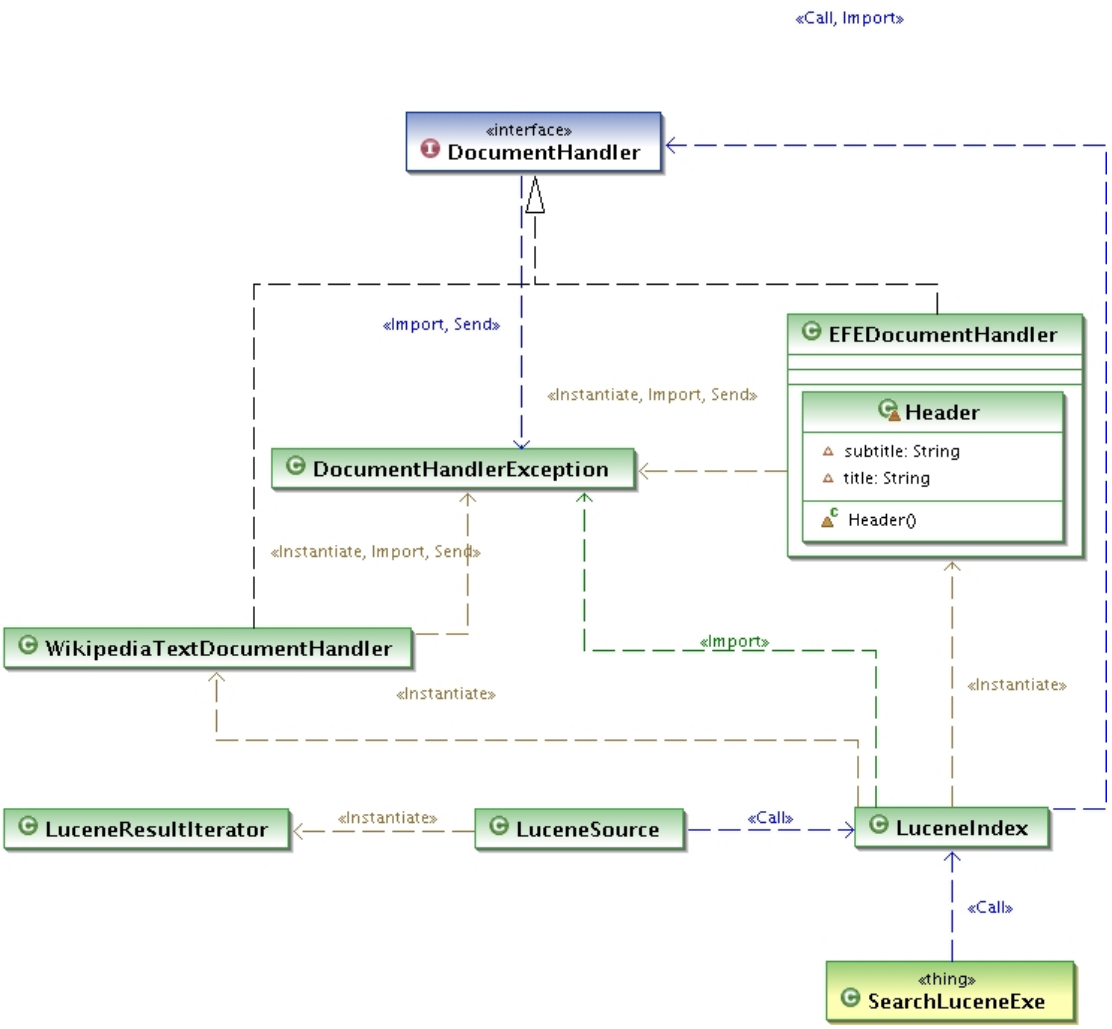

Figura 3.8: Diagrama de clases *lucene*

3.2.4. Análisis Lingüístico

El paquete *linganalysis* es el paquete encargado del análisis lingüístico. Está compuesto por dos subpaquetes: tokenizer y stilus. El subsistema de análisis lingüístico es utilizado por el resto de subsistemas, como puede verse en el diagrama de dependencias de paquetes.

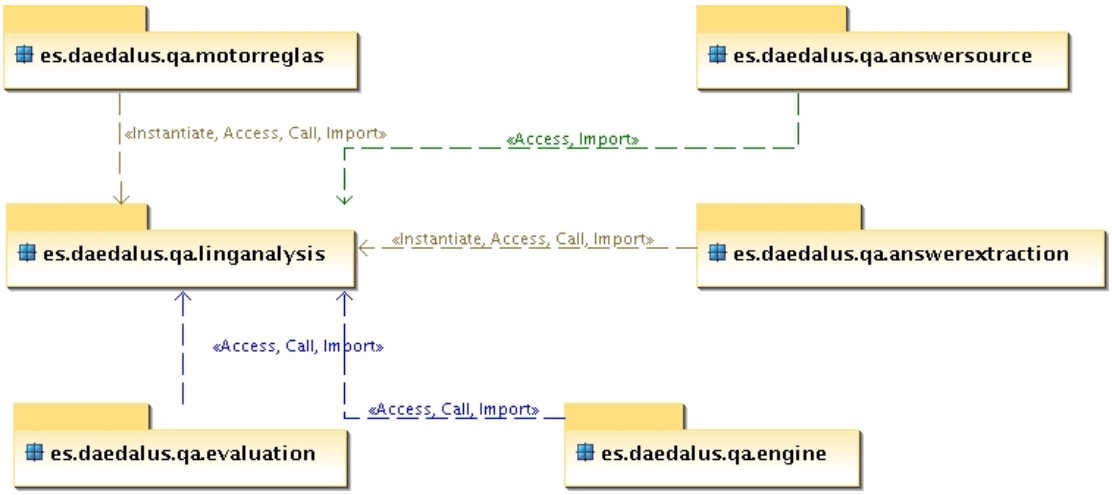

Figura 3.9: Dependencia de paquetes *linganalysis*

En este subsistema se encuentran las clases utilizadas para el análisis sintáctico y semántico. Estos análisis son realziados tanto a las preguntas como a la colección de documentos donde se realiza la búsqueda de la respuesta. En la clase Syntax se definen las etiquetas morfosintácticas que pueden asociarse a un token.

La clase SemanticInfo contiene etiquetas semánticas interpretadas por el sistema, como puede verse en el diagrama estas etiquetas son utilizadas tanto en la pregunta (por la clase QuestionClassificationData), para definir el tipo de respuesta esperada, como en la propia respuesta (AnswerData). Las etiquetas semánticas están organizadas en niveles siguiendo la clasificación de [Sekine et al., 2002]. La clases Analysis, contiene información del análisis síntactico. Almacena la información a través de etiquetas, utilizando la estructura descrita en la clase Tag. La clase Constituent define la forma de almacenar las frases. Un objeto Constituent puede representar cualquier componente de la frase, ya sea un token, sintagma o la propia frase.

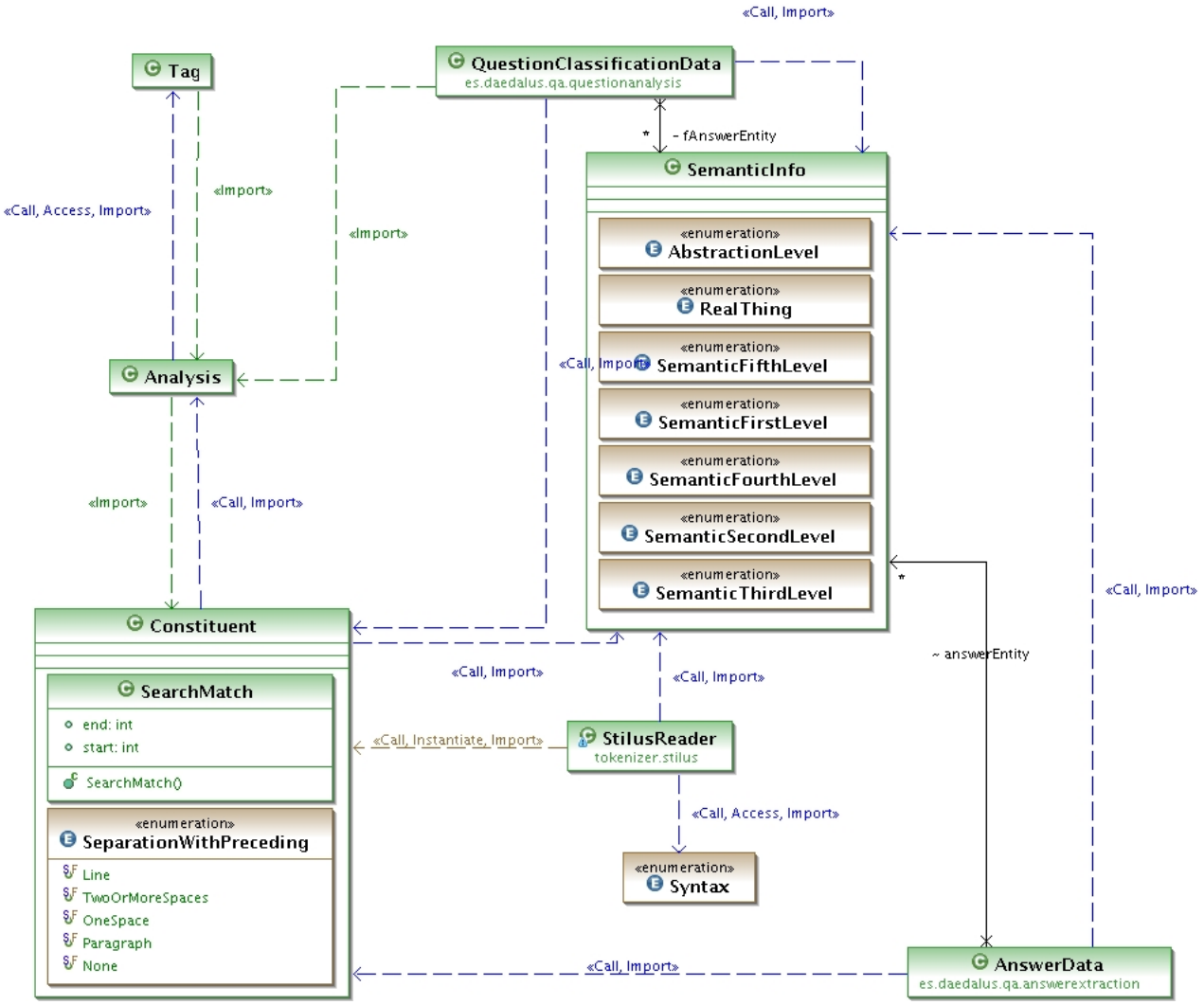

Figura 3.10: Diagrama de clases *linganalysis*

Se muestran a continuación los paquetes utilizados por *tokenizer*. Se utilizan instancias del paquete *linganalysis* para realizar la tokenización. Los elementos devueltos al tokenizar un texto son del tipo Constituent. Se puede apreciar también que el paquete *stilus* implementa un tokenizador.

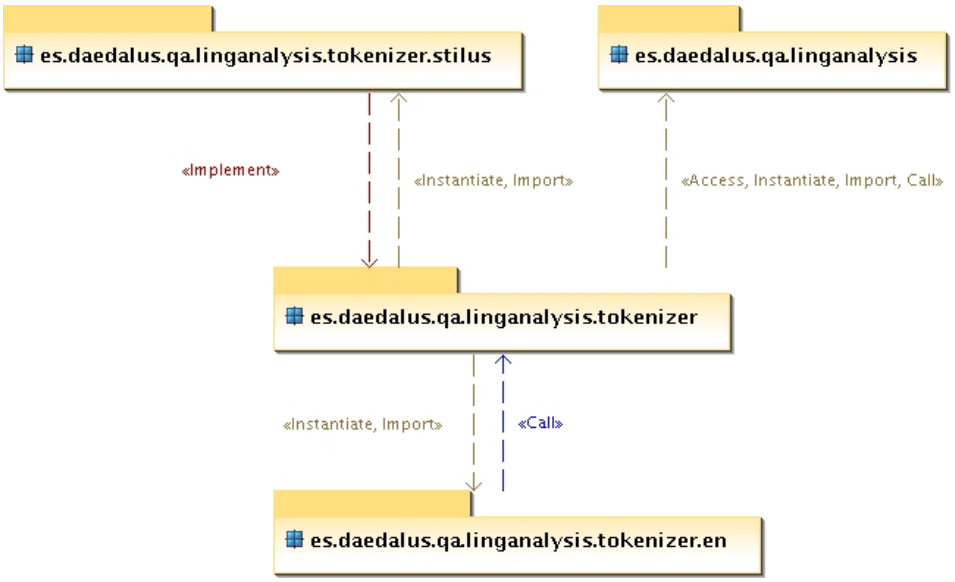

Figura 3.11: Dependencia de paquetes *tokenizer*

El paquete *tokenizer* contiene la interfaz ITokenizer que define las operaciones que debe proporcionar cualquier módulo de tokenización que se desee integrar en el SBR. Estas operaciones se reducen a tokenizar un pasaje o conjunto de sentencias y tokenizar el contenido de un fichero. El resultado de dicha tokenización es un objeto Constituent. La clase TokenizerFactory implementa el patrón de creación AbstractFactory. Contiene el método factory que genera el objeto ITokenizer adecuado dependiendo del idioma.

Este paquete contiene la clase BasicTokenizer que define un tokenizador básico. Implementa una función que tokeniza un texto a partir de una cadena de texto. LuceneTokenizerFilter, que implementa la interfaz ITokenizer, es un filtro que da a una frase ya tokenizada anotaciones alternativas obtenidas de un diccionario almacenado en un índice de Lucene. Las alternativas son añadidas a la lista de análisis de la frase.

La herramienta TokenizeDirExe es utilizada para tokenizar documentos dejando los análisis almacenados para ser usados por el SBR. Utiliza la clase TokenizeFactory para construir el tokenizador dependiendo del idioma. Se le pude pasar directamente un fichero con un documento o una carpeta que se recorrerá recursivamente. Se utilizará el tokenizador y la carpeta de almacén de

análisis especificadas para el idioma en el fichero de configuración. Si los análisis ya existen en la carpeta, no se reanalizarán, por lo que si se desea esto hay que borrarlos antes a mano. Esta tarea es realizada fuera de línea, con el objetivo de optimizar tiempos durante la realización de la consulta.

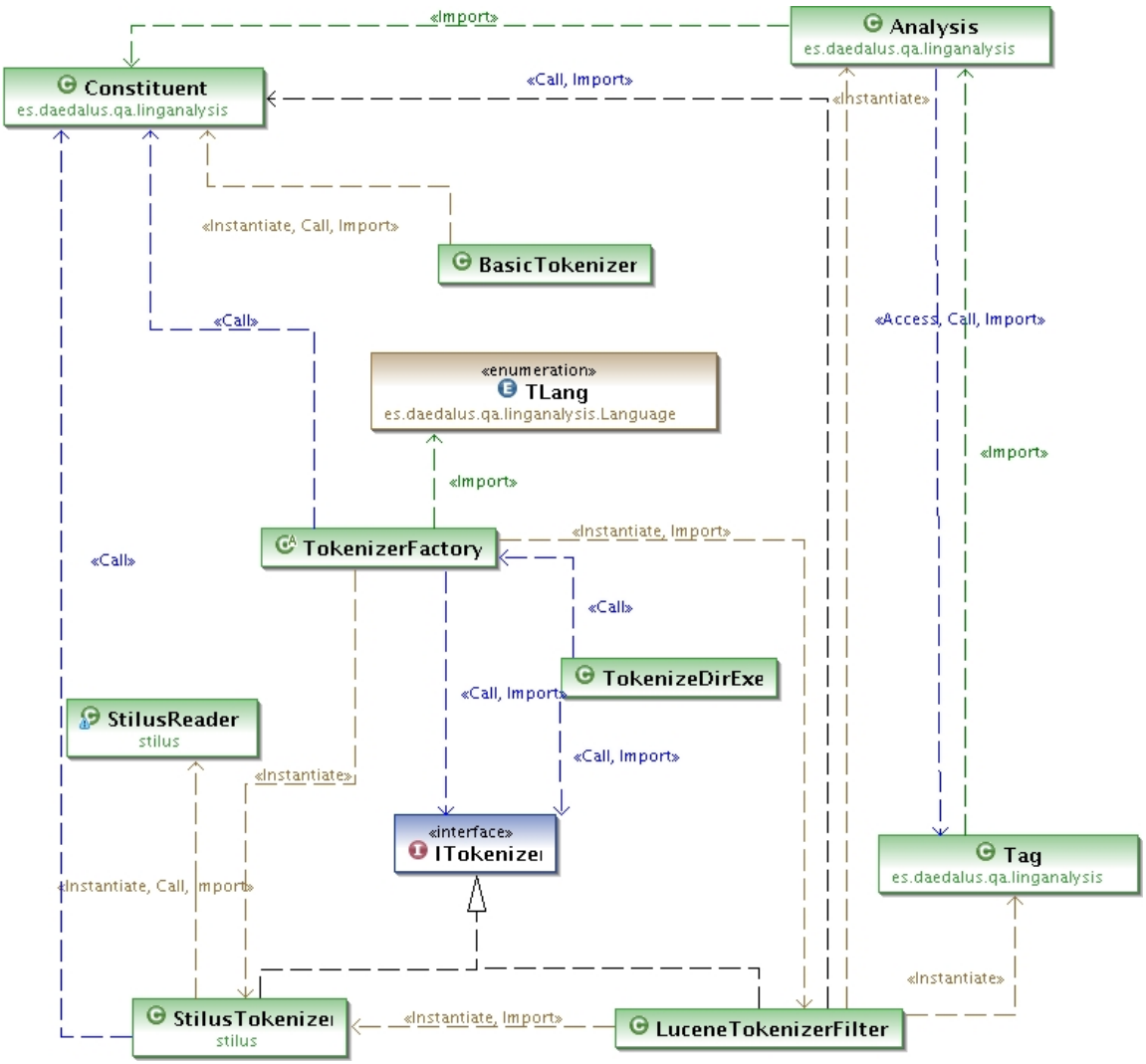

Figura 3.12: Diagrama de clases *tokenizer*

El paquete *stilus* implementa la interfaz ITokenizer definida en *tokenizer*, usando las clases definidas en este paquete para realizar la tokenización y devolver el resultado de ésta.

El tokenizador implementado en el paquete *stilus* está basado en la herramienta Stilus[2], desarrollada por Daedalus.

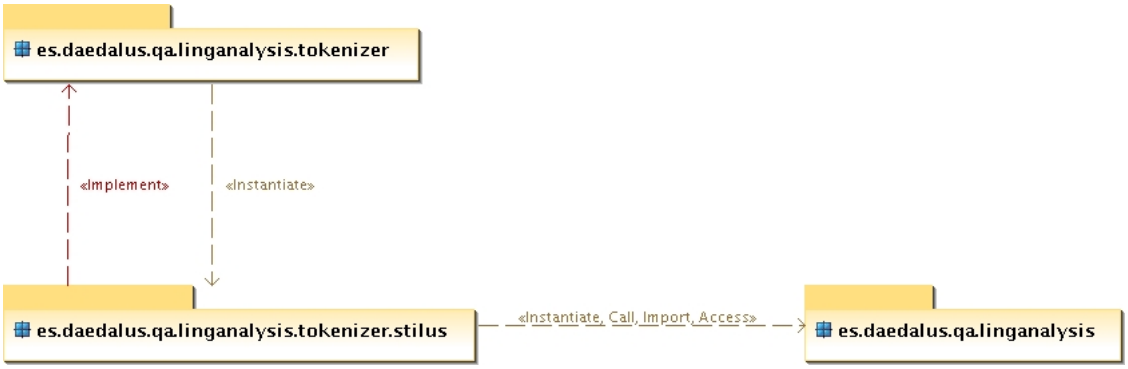

Figura 3.13: Dependencia de paquetes *stilus*

El paquete *stilus* contiene dos clases: StilusReader y StilusTokenizer. La primera está encargada de interpretar la salida del analizador lingüístico de Stilus, almacenando el resultado en las estructuras correspondientes. Para ello emplea expresiones regulares que interpretan las etiquetas devueltas por Stilus. También se interpretan las etiquetas devueltas con la información semántica. Para esto hace uso de las clases definidas en *linganalysis,* las cuales rellena con la información extraída del análisis sintático-semántico .

Por su parte, StilusTokenizer, implementa un tokenizador basado en Stilus. Esta implementación emula una llamada a consola al tokenizador e interpreta la salida por consola. La ruta en la que se encuentra el ejecutable de Stilus y la ruta en la que se encuentran los diccionarios son parámetros del tokenizador, que deberán de especificarse en el archivo de configuración del sistema. También deberá indicarse la ruta en la que deberán almacenarse los resultados de la tokenización siempre que se trate de documentos.

[2]Stilus es una herramienta lingüística capaz de realizar un etiquetado morfosintáctico y semántico de un texto, así como lematizar palabras. La herramienta está disponible en http://stilus.daedalus.es/stilus.php

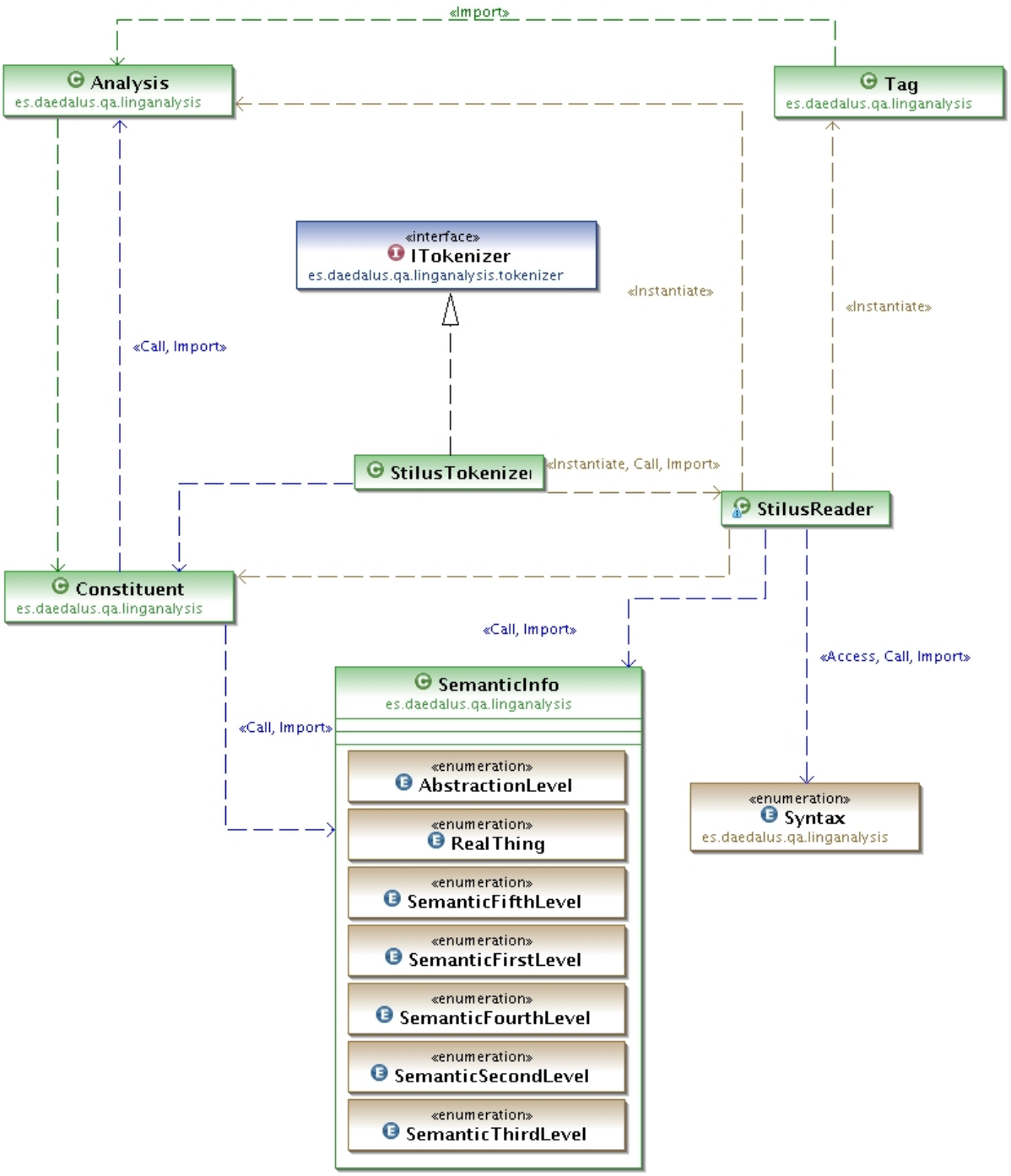

Figura 3.14: Diagrama de clases *stilus*

3.2.5. Reglas

El paquete *motorreglas* contiene las clases que especifican las reglas para la clasificación de la consulta y la extracción de la respuesta. A través de las reglas se pretende hacer más sencilla la ampliación del sistema. Este paquete hereda clases del paquete motorreglas que contiene reglas para el tratamiento de textos. También importa clases de los paquetes *answerextraction* y *questionanalysis*, que se encarga de rellenar con los datos obtenidos al aplicar las reglas. Utiliza también clases de *linganalysis* para manejar los datos extraídos.

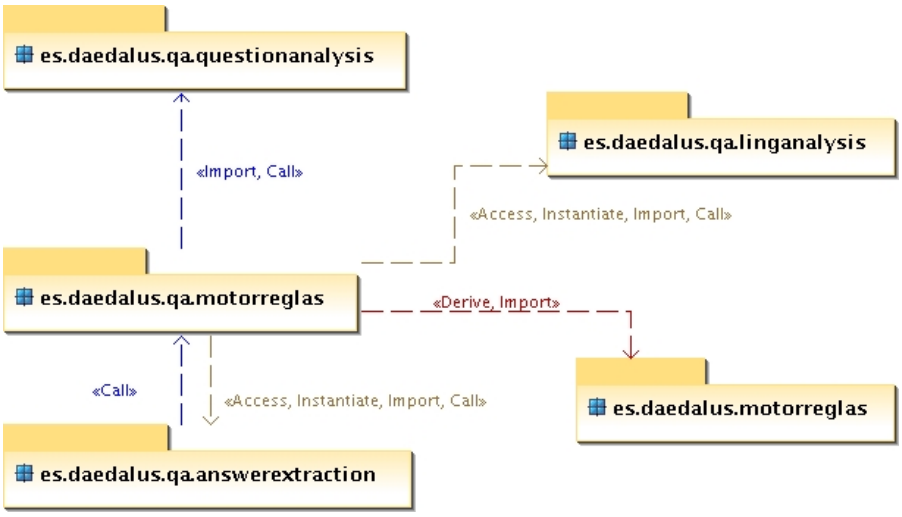

Figura 3.15: Dependencia de paquetes *motorreglas*

En MotorReglasQA se describen reglas básicas para el manejo del contenido de la pregunta y la respuesta del SBR. Estas pueden ser comprobar o asignar características propias de la pregunta. A su vez, esta clase hereda todas las reglas definidas en MotorReglas que contiene reglas de más bajo nivel, relacionadas con el tipo de la palabra (si es un número, una sigla, está en mayúsculas, etc.) o con su posición dentro del texto. De la clase MotorReglasQA heredan las clases RulesBasedQuestionClassifierES, con las reglas necesarias para clasificar la pregunta y MotorReglasExtraccion que define las funciones que utiliza ReglasExtraccionRespuestasES para extraer la respuesta. MotorReglasExtraccion tiene una instancia de ContextoReglasExtraccion, que contiene los datos relativos a una consulta: un objeto QuestionData con la pregunta, un objeto

AnswerDataList con las respuestas (que se irán añadiendo) y un objeto SupportData con los datos de soporte. Estos campos son los que se irán rellenando en función a las reglas que se cumplan. El diagrama de clases que describe estas relaciones se puede ver en la Figura A.8.

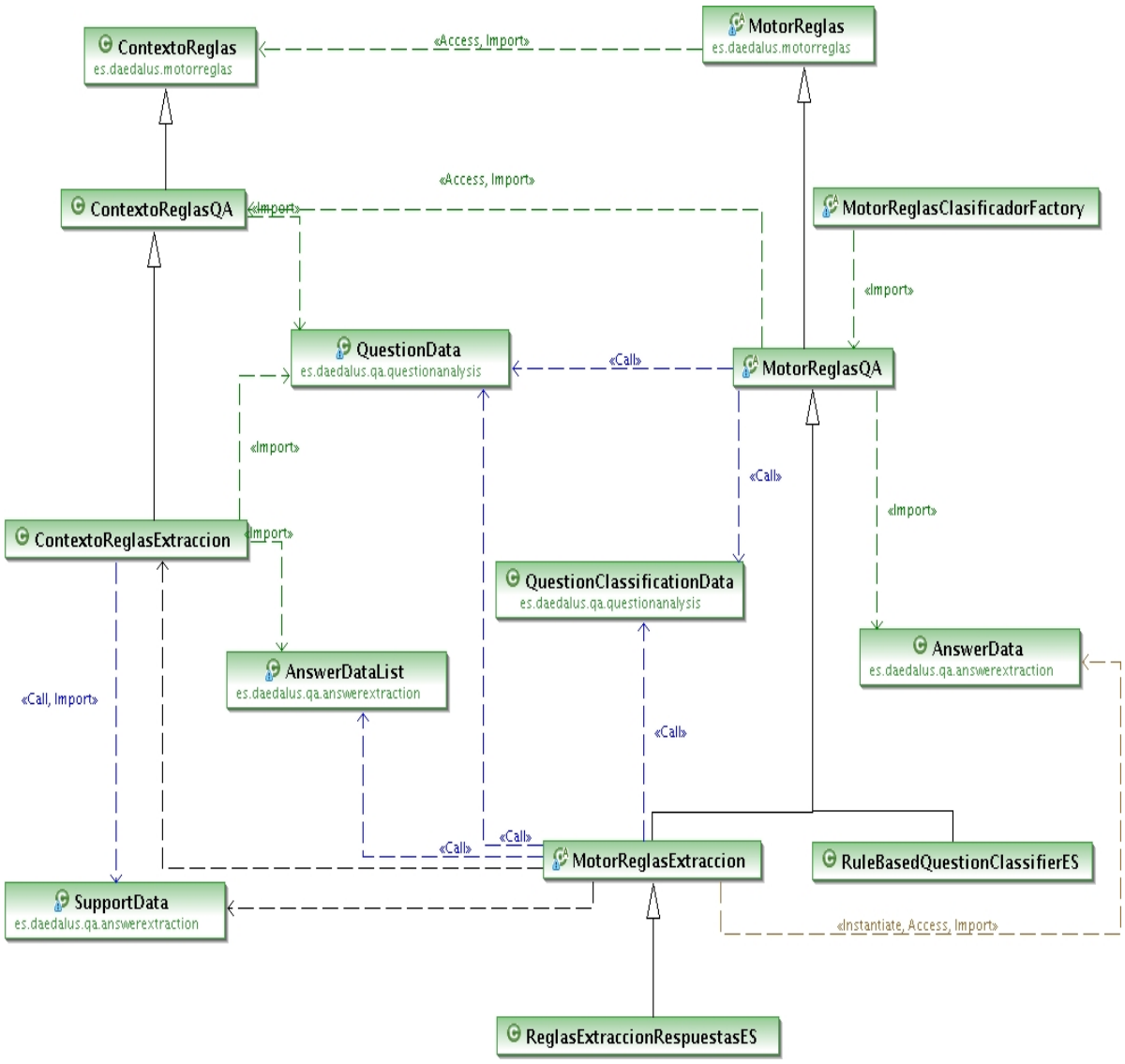

Figura 3.16: Diagrama de clases *motorreglas*

3.2.6. Extracción de la Respuesta

El paquete *answerextraction* es el encargado de extraer las respuestas y definir

la estructura de datos necesaria para almacenarlas junto con los documentos de soporte. Para realizar la extracción se utilizan las reglas para tal efecto descritas en el paquete *motorreglas* y la clase QuestionData del paquete questionanalysis.

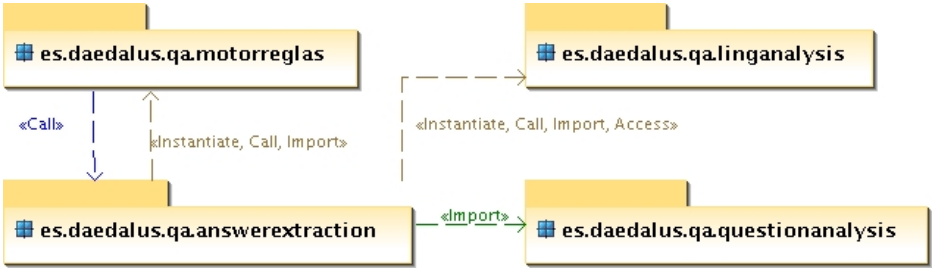

Figura 3.17: Dependencia de paquetes *answerextraction*

El diagrama de clases de este paquete se describe en la Figura A.9. El interfaz IAnswerExtractor define las operaciones que debe implementar cualquier módulo de extracción de respuestas. La clase RuleBasedAnswerExtractor implementa esta interfaz, especificando la función extractAnswers que crea un objeto de tipo MotorReglasExtraction (que será RulesBasedAnswerExtractionES para español). Este objeto ejecuta la función aplicarReglas, que se encarga de aplicar las reglas. La función recibe un objeto ContextoReglasQA, que contiene objetos de tipo QuestionData, AnswerDataList y SupportData. En el primer objeto se tiene información sobre la pregunta, función de la cual serán aplicadas unas reglas u otras. Los otros dos objetos, serán rellenados con la información sobre las respuestas y el soporte.

AnswerDataList define una lista de objetos AnswerData con la información de que se dispone para una respuesta generada por el sistema. Actúa a modo de plantilla a rellenar por los procesos de búsqueda de documentos y de extracción de respuestas.

SupportData incluye datos sobre frases de soporte y las fuentes (los índices) en las que aparecen. Es necesario relacionar cada frase con su fuente correspondiente, para poder asociarlas con la respuesta a la que dan soporte. Así, una misma respuesta vendrá acompañada de un conjunto de pares frase-fuente (uno o más), que servirán como justificantes de la respuesta extraída.

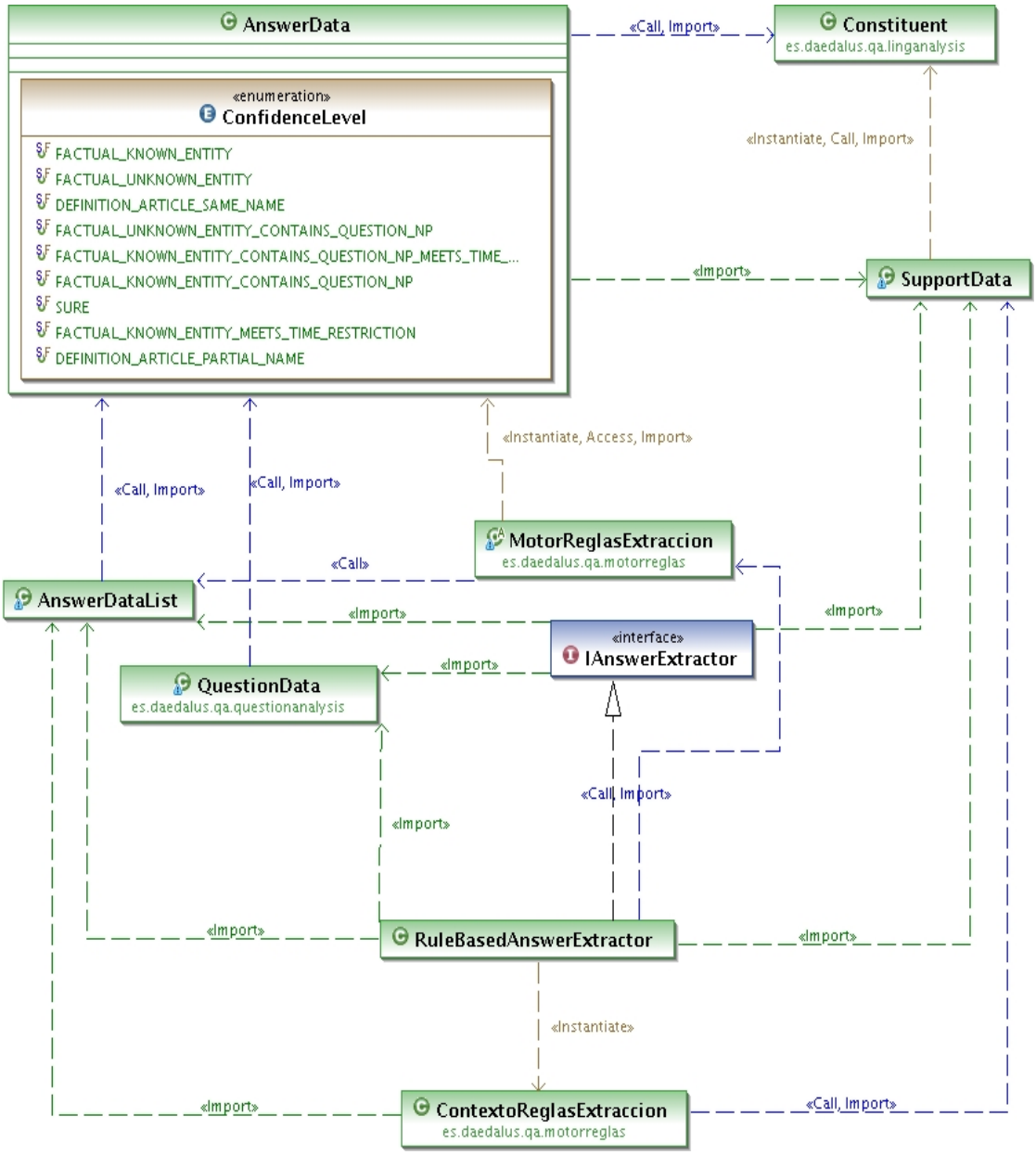

Figura 3.18: Diagrama de clases *answerextraction*

3.2.7. Ordenación de Respuestas

El paquete *ranking* es el encargado de la ordenación de las respuestas. Para ello utiliza los paquetes *questionanalysis*, *linganalysis* y *answerextraction*, de los que obtiene información de la pregunta, las respuestas, así como de sus análisis

sintácticos y semánticos. Con esta información establece la ordenación definitiva de las respuestas.

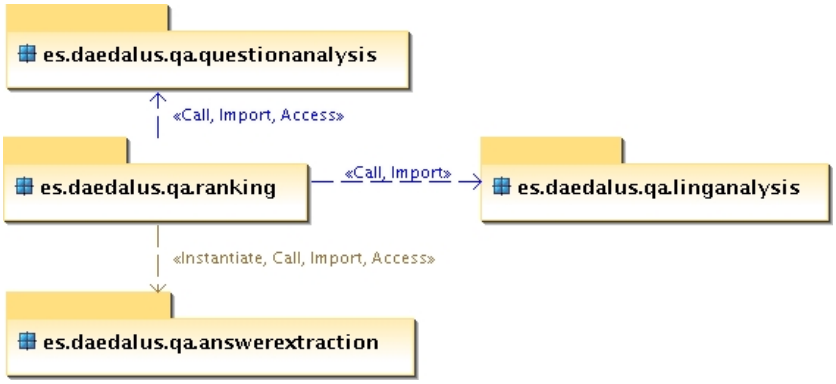

Figura 3.19: Dependencia de paquetes *ranking*

El interfaz IRanker define las operaciones que debe implementar cualquier módulo de puntuación de respuestas para el SBR. Las operaciones descritas son:

- rank: Método para efectuar la puntuación y ordenación de un conjunto de respuestas.
- consolidate: Método para combinar respuestas iguales. Por respuestas iguales se entiende aquellas que hacen referencia al mismo concepto pero que se expresan de forma diferente. Para determinar si dos respuestas son la misma se emplea el método 'compareAnswerString'.
- compareAnswerStrings: Método para determinar si dos respuestas hacen referencia al mismo concepto y deberían ser consideradas como una única respuesta.
- globalScore: Método para asignar pesos a las respuestas teniendo en cuenta el conjunto de respuestas obtenido. Este método puede apoyarse en el siguiente, 'localScore', para establecer el peso global. Este método será el empleado por el método 'ranker' para establecer la puntuación de las respuestas. Se incluye en la interfaz para poder modificar su comportamiento en diferentes implementaciones del ranker.

- localScore: Método para asignar un peso local a una pregunta. Se pretende que este peso local refleje la calidad de la respuesta atendiendo a parámetros internos a la respuesta, como su longitud, las frases de soporte de las que se ha extraído, etc.

La clase Ranker implementa esta interfaz, definiendo de la siguiente forma los métodos:

- localScore: Asigna un peso local a una respuesta. Este peso local sólo depende de la frase en la que se encuentra la respuesta. Se calcula a partir de:
  - numSemQuestion: El número de etiquetas semánticas válidas para la pregunta que aparecen en la frase de soporte.
  - numSemExpected: El número de etiquetas semánticas de la frase de soporte que aparecen en la pregunta.
  - numSem: El número de palabras con etiqueta semántica que hay en la frase de respuesta
  - numTermQuestion: El número de palabras de la frase soporte cuyo stem aparece en la pregunta
  - numTerm: Número de palabras relevantes de la frase soporte
  - La función de ordenación se puede ver a continuación, pSem y pWord, son el peso dado para la coincidencia semántica y de palabras relevantes respectivamente. La función prox(x) asigna un peso al término según su proximidad a la respuesta.

$$localScore = pSem * \frac{prox(numSemQuestion) + prox(numSemExpected)}{2 * prox(numSem)} + pWord * \frac{prox(numTermQuestion)}{prox(numTerm)}$$

- golbalScore: Asigna un peso global a cada respuesta. A la entrada de este método se tienen las mejores respuestas ordenadas por su peso local. Se juntan las respuestas que se consideran coincidentes y se calcula el score global como:

$$globalScore = \frac{\#respuestas\ coincidentes}{\#total\ respuestas\ sin\ consolidar}$$

- rank: Ejecuta la función localScore sobre cada respuesta a una pregunta, después realiza la función globalScore sobre la cuestión. El resultado final se almacena en el propio objeto QuestionData.

Para la ejecución de estos métodos Rank utiliza tanto datos de la clasificación de la pregunta, como datos del análisis lingüístico de la respuesta. La dependecia de clases se puede ver en el diagrama de la Figura A.10.

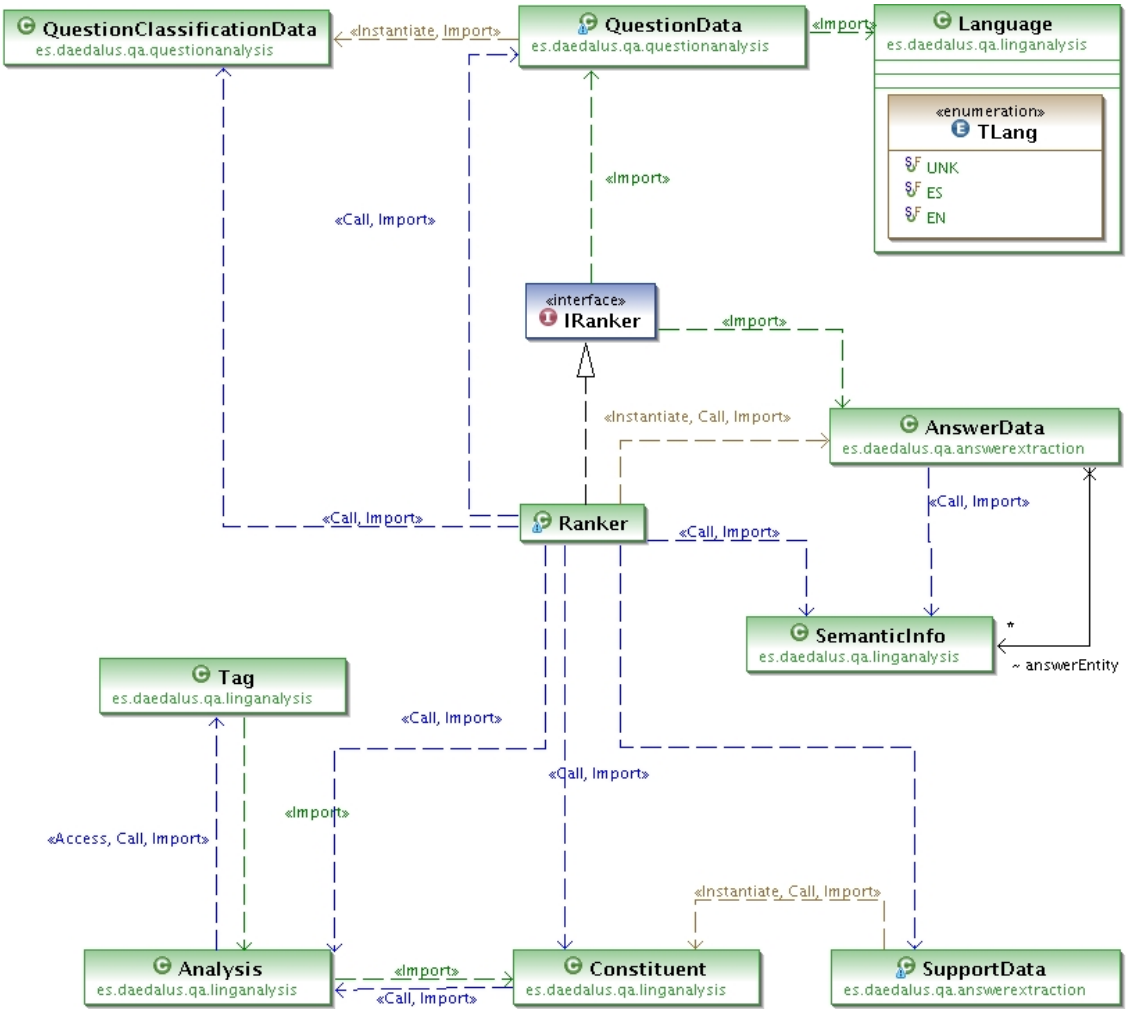

Figura 3.20: Diagrama de clases *ranking*

3.2.8. Motor del SBR

El paquete *engine* es el motor del programa, encargado de enlazar el resto de clases y métodos. Utiliza las estrutcuras de datos y métodos descritos en el resto de paquetes.

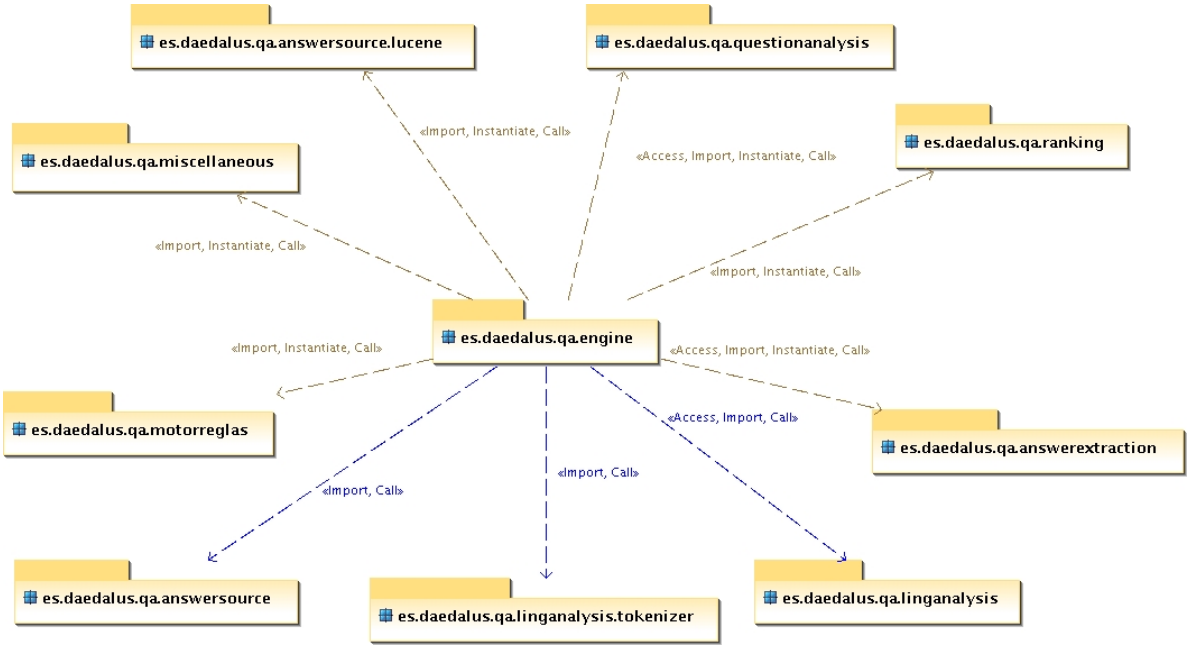

Figura 3.21: Dependencia de paquetes *engine*

Dentro de este paquete se encuentra la clase QAEngine, que es la encargada de combinar los elementos necesarios para implementar una búsqueda de respuestas a una pregunta dada. Esta clase cuenta con un objeto de las siguientes clases:

- CStopwords, que contiene una lista de palabras de parada CStopwords.
- ITokenizer, encargado de la tokenización de la pregunta y las fuentes. A través de la clase TokenizeFactory se selecciona el objeto tokenizador. Para el idioma español es seleccionado StilusTokenizer.
- MotorReglasQA, para clasificar la pregunta. Al igual que con el tokenizador, a través de la clase MotorReglasClasificadorFactory, se selecciona la clase para clasificar la pregunta. En nuestro caso es RuledBasedQuestionClassifierES.

- LuceneSource, objeto para realizar las búsquedas en un índice Lucene.
- IAnswerExtractor, para extraer las respuestas.
- Ranker, encargado de ordenar las respuestas.

El método encargado de realizar la búsqueda es searchQuestion, al que se le introduce una cadena de caracteres con la pregunta realizada por el usuario. Este método crea un objeto QuestionData, el cual será rellenado a lo largo del proceso de búsqueda de respuestas. Una vez creado este objeto, se inicializan los objetos descritos anteriormente. Los pasos seguidos para obtener la respuesta son los siguientes:

- La pregunta es analizada, de lo que se obtiene el tipo de pregunta, el tipo de respuesta esperado y los tokens de la pregunta.
- Se buscan documentos con posibles respuestas, utilizando para ello el objeto LuceneSource.
- Para cada documento, se recorre buscando posibles respuestas, utilizando IAnswerExtractor.
- Una vez seleccionadas las respuestas candidatas de todos los documentos se ordenan las respuestas utilizando el objeto Ranker.

El diagrama de clases con las dependencias de QAEngine se puede ver en la Figura A.11.

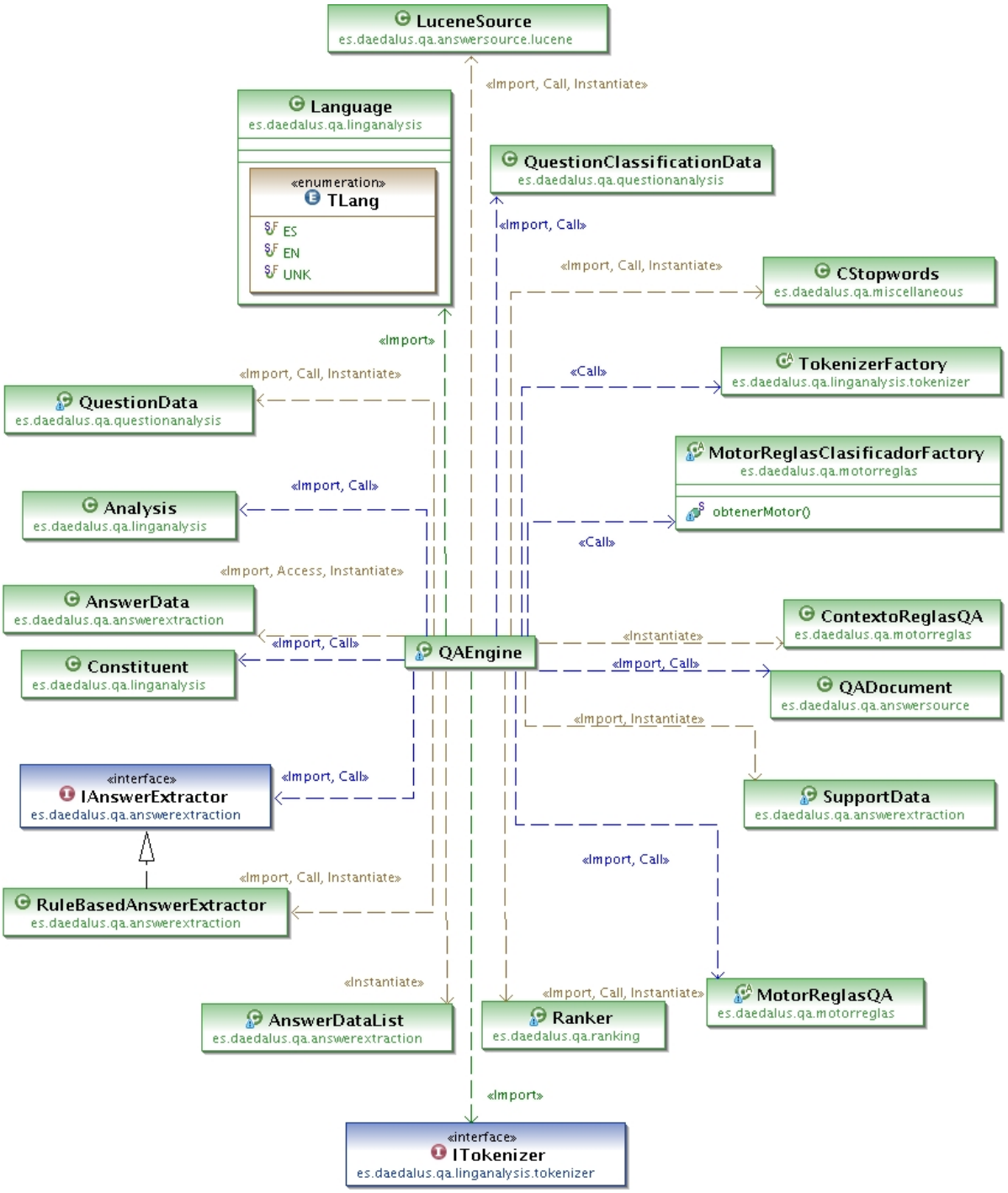

Figura 3.22: Diagrama de clases *engine*

3.2.9. Evaluación de las Respuestas

El paquete *evualation* es el encargado de evaluar las respuestas. La creación de este subsistema responde a la necesidad de evaluar los resultados obtenidos por el SBR de forma rápida y eficiente, permitiendo comparar diferentes versiones del sistema para ver si los cambios efectuados mejoran o empeoran el rendimiento.

Antes de adentrarse en la descripción de este subsistema es necesario definir dos conceptos utilizados como medidas de evaluación [de Pablo, 2003]:

- MRR (*Mean Reciprocal Rank*): Se utiliza para determinar un valor medio donde se encuentran las primeras respuestas correctas dentro de las respuestas que ofrece el sistema. Permite evaluar una lista ordenada de respuestas. El sistema recibe una puntuación inversamente proporcional a la posición en el ranking de la primera respuesta acertada y se realiza la media para todas las preguntas. La profundidad con que se buscan las respuestas es definida por el sistema. Viene definida por la expresión:

$$MRR = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{rank_i}$$

- CWS (*Confidence-Weighted Score*): Esta medida favorece a los sistemas que saben estimar la confianza en las respuestas ya que las respuestas se ordenan por la confianza antes de calcular. Viene definida por la función:

$$CWS = \frac{1}{Q}\sum_{i=1}^{Q}\frac{\#correctas\,hasta\,i}{i}$$

El paquete *evaluation* importa *answerextraction* y *answersource* para comparar datos de la respuesta y de datos de soporte, *questionnalysis* para obtener datos de la pregunta y *linganalisis* para el idioma. También utiliza la clase QAEngine del paquete *engine*, para resolver las preguntas que luego serán comparadas.

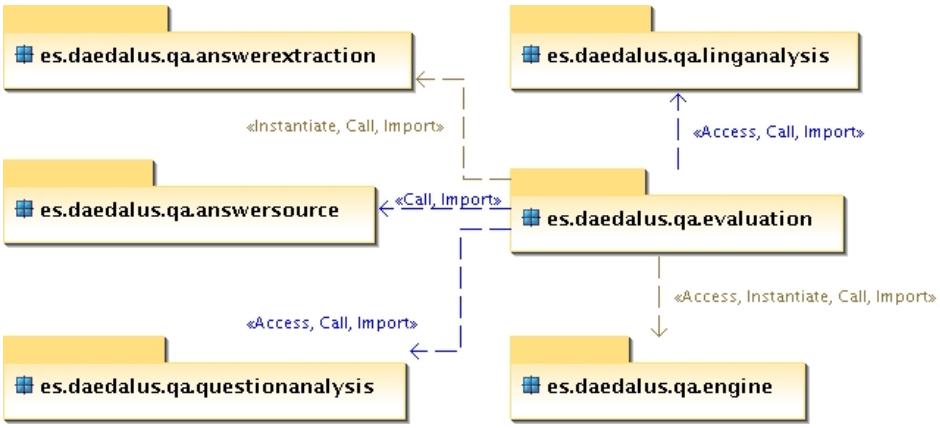

Figura 3.23: Dependencia de paquetes *evaluation*

El diagrama de clases está dividido en tres partes. Las dos primeras corresponden a un mismo diagrama dividido en dos, unidos por la clase EvalQAEngineXML. Esta clase es la encargada de ejecutar y coordinar el proceso de evaluación, el equivalente a QAEngine del paquete *engine*. El tercer diagrama muestra las relaciones de la clase Printer.

Las clases utilizadas para contener los datos de la evaluación de preguntas y respuestas son QuestionEval y AnswerEval, respectivamente. Los resultados del sistema se encapsulan en objetos QuestionEval y se comparan con los datos de evaluación disponibles.

Las clases LoadCLEFData y LoadXMLCLEFData cargan datos de preguntas y respuestas de distintas convocatorias de CLEF. La primera lee ficheros en texto plano y la segunda con formato XML. Se construyen objetos QuestionEval con los datos de las preguntas y respuestas de esos ficheros. Estos objetos son utilizados posteriormente para evaluar las respuestas obtenidas por el sistema a las mismas preguntas.

Las clases Comparator y Normalization definen una colección de métodos que permiten normalizar y comparar listas de respuestas.

En el diagrama de la Figura A.13, se muestran las clases encargadas de calcular y almacenar las medidas de evaluación (MRR y CWS). La clase TypeEvaluator permite definir el tipo de preguntas sobre el que se va a hacer la evaluación. En

SubTypeEvaluator se concreta sobre que elementos se halla el MRR (predefinido a respuestas, sentencias y documentos) y a que profundidad.

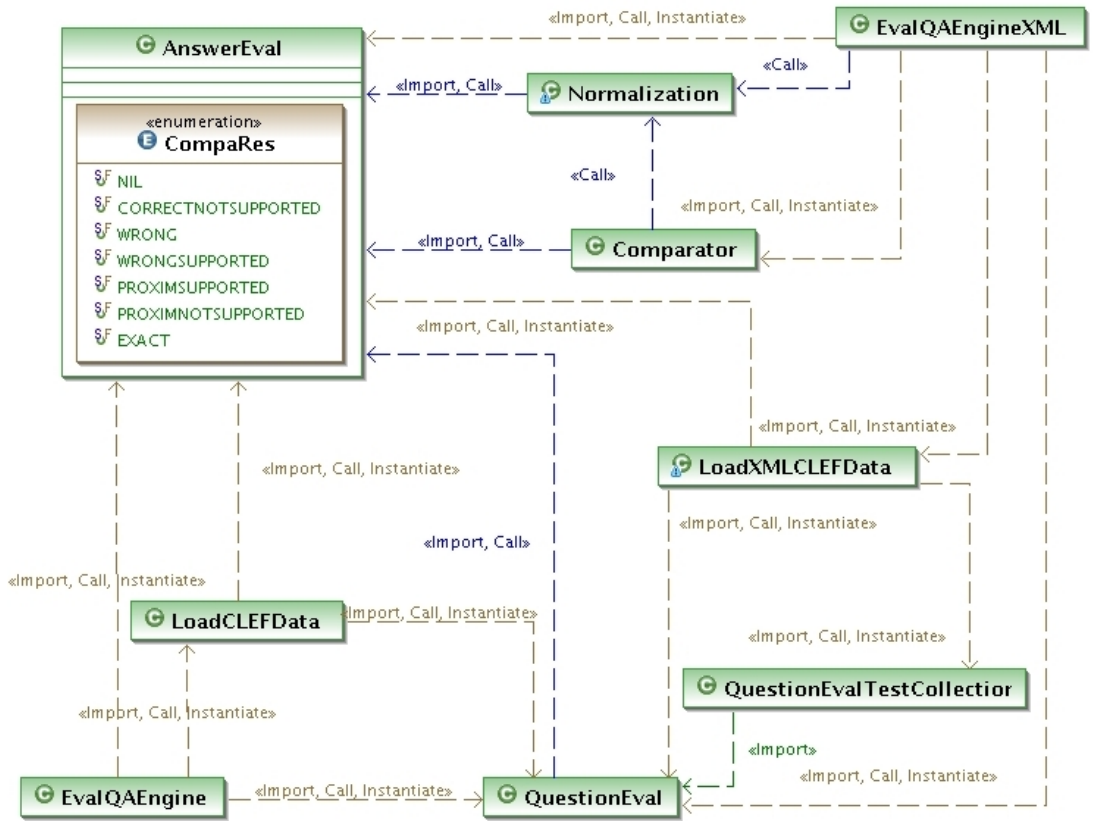

Figura 3.24: Diagrama de clases *evaluation* 1/3

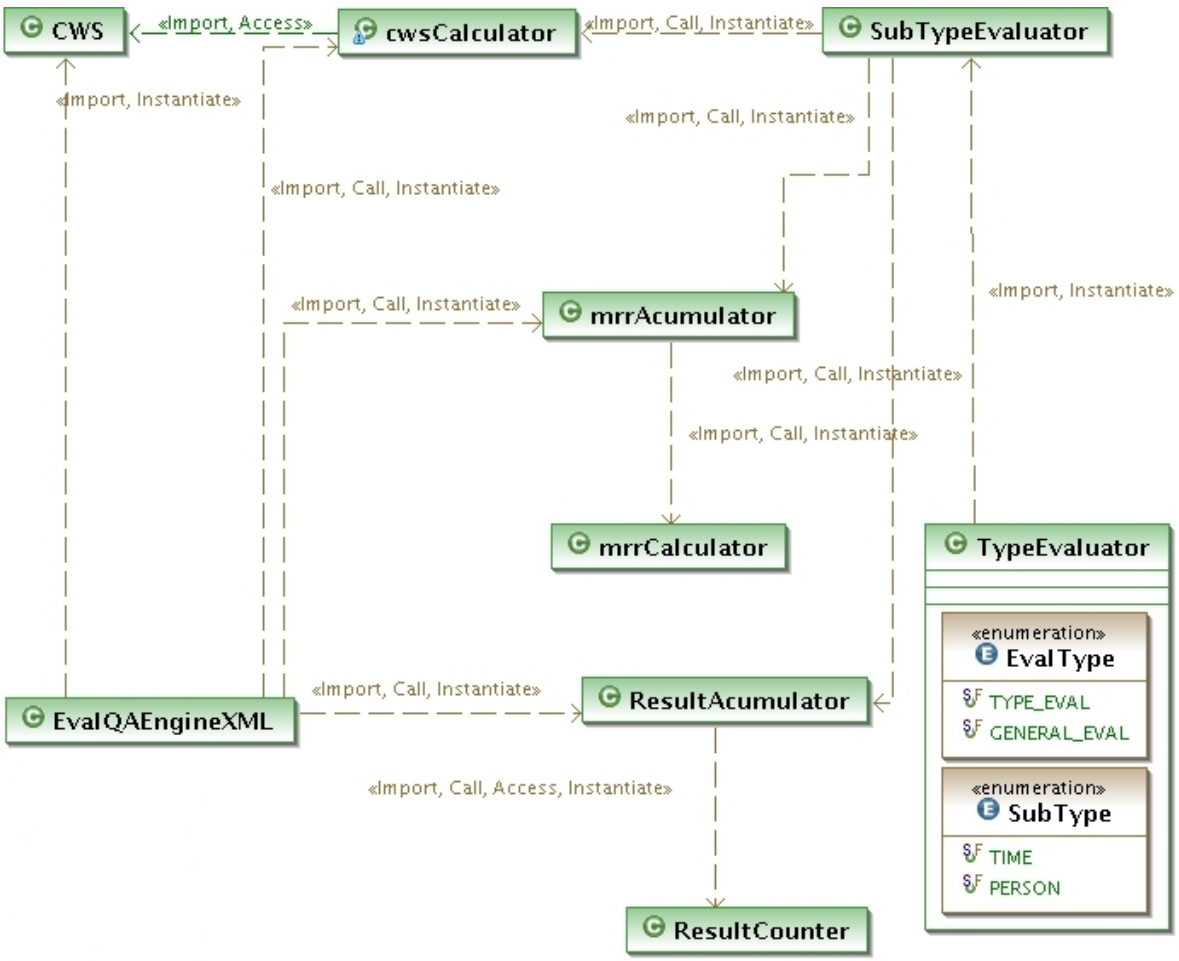

Figura 3.25: Diagrama de clases *evaluation* 2/3

El tercer diagrama de clases, que se muestra en la Figura A.14, muestra las relaciones de la clase Printer. Se muestra por separado, ya que esta clase tiene dependencias con la mayoría de las clases de este paquete y la inclusión en los diagramas anteriores genera muchas dependencias cruzadas que no permiten observar el diagrama con claridad.

Printer es la clase encargada de la impresión de resultados en un formato determinado. La salida del sistema de evaluación se compone de un mensaje por pantalla y de un documento XML asociado a una transformación XSLT.

Por cada conjunto de preguntas evaluadas se obtiene:

- El porcentaje de cada tipo de resultado. Los tipos posibles son:
  - Correcta.
  - Correcta sin documento de soporte.
  - Aproximada con documento de soporte.
  - Aproximada sin documento de soporte.
  - Erronea con documento de soporte.
  - Erronea sin documento de soporte.
- El CWS de las respuestas.
- El MRR de las respuestas para la profundidad total y para distintas profundidades.
- El CWS de las sentencias de soporte.
- El MRR de las sentencias para la profundidad total y para distintas profundidades.
- El CWS de los documentos de soporte.
- El MRR de los documentos de soporte, para la profundidad total y para distintas profundidades.

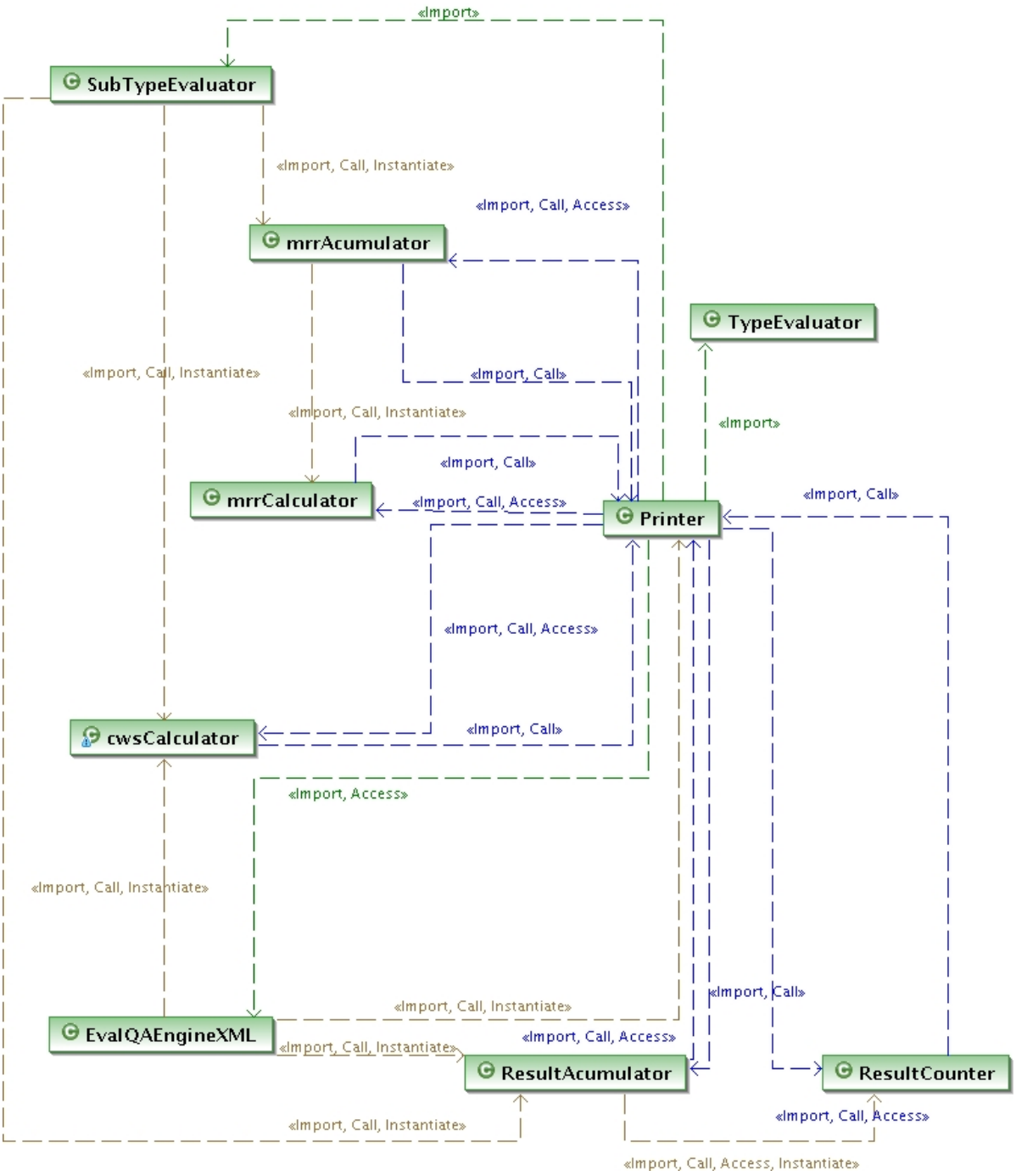

Figura 3.26: Diagrama de clases *evaluation* 3/3

3.3. Especificación de Casos de Uso

En la siguiente figura se especifica un diagrama con los distintos casos de uso, donde se muestra cuales son los usuarios del sistema y las interacciones que pueden realizar con éste.

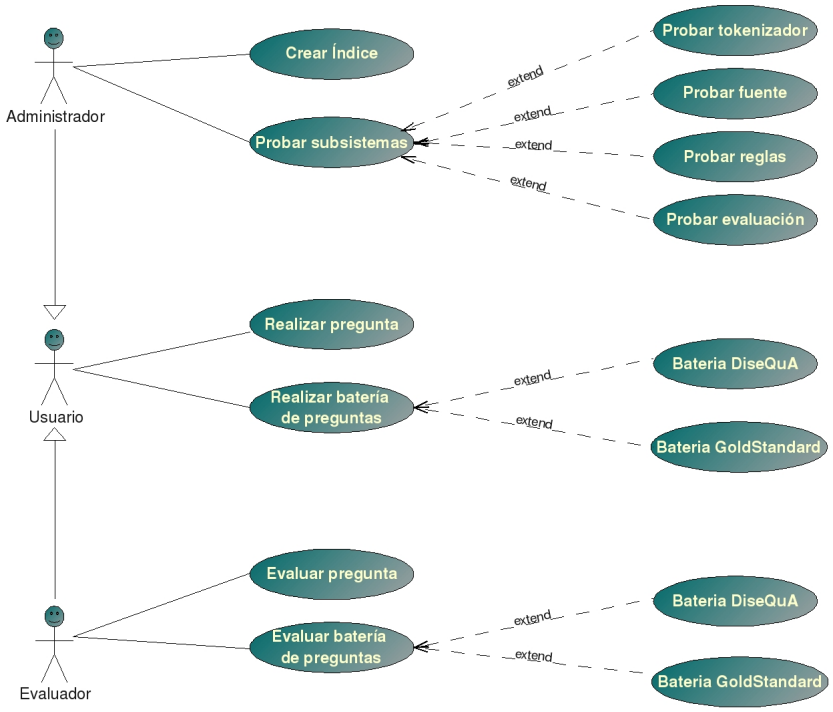

Figura 3.27: Diagrama de Casos de Uso

Los actores involucrados que se pueden ver en la Figura 3.27, son:

Administrador: Encargado de crear los índices sobre las colecciones, y probar los distintos subsistemas de forma independiente.

Evaluador: Cuyo objetivo es evaluar el sistema y modificar las propiedades de configuración para optimizar el sistema, dependiendo de los resultados obtenidos.

Usuario: Puede realizar una pregunta o varias contenidas en un fichero.

Los casos de uso representados en el diagrama se describen como sigue:

Caso de Uso	Descripción
Crear índice	Crear un índice de una colección sobre el cual realizar las búsquedas.
Probar subsistema	Probar alguno de los subsistemas.
Probar tokenizador	Probar el tokenizador sobre una pregunta, un texto o un fichero.
Probar fuente	Prueba para el indexador de Lucene.
Probar reglas	Pueba para comprobar la correcta clasifiación de preguntas.
Probar evaluación	Prueba de los distintos componentes del paquete evaluation.
Realizar pregunta	Responde a una pregunta introducida por el usuario.
Realizar batería de preguntas	Responde a una lista de preguntas introducidas por fichero.
Bateria DiseQuA	Responde a una lista de preguntas introducidas a traves de un fichero xml. Este fichero respeta la estructura definida en la convocatoria CLEF 2008
Bateria GoldStandard	Responde a una lista de preguntas introducidas a través de un fichero XML. Este fichero respeta la estructura definida en las convocatorias CLEF 2004-2005
Evaluar batería de preguntas	Halla la respuesta a una lista de preguntas, de las que se conoce a priori el resultado, y evalúa el resultado.
Betería DiseQuA	Realiza la evaluación sobre un fichero con la estructura definida en CLEF 2008
Batería GoldStandard	Realiza la evaluación sobre un fichero con la estructura definida en CLEF 2004-2005

Cuadro 3.1: Descripción de Casos de Uso

3.4. Dominio de búsqueda

El SBR respond.es es un sistema de dominio cerrado. Cuenta con dos colecciones sobre las que realizar la búsqueda de respuestas: artículos de la agencia EFE (de 1992 y 1993) y artículos de la Wikipedia. Como se ha visto en el apartado 2.4.2, en un dominio cerrado es más sencillo estudiar los distintos casos en los que se puede presentar la definición del término y crear patrones. En un dominio abierto la solución es más compleja, es necesario adentrarse más profundamente en el campo de la lingüística para dar con las respuestas dado que existen muchas formas en las cuales un concepto puede ser descrito en lenguaje natural. Obtener un conjunto completo de patrones lingüísticos es una tarea que requiere un avance en las herramientas de PLN.

La colección de documentos con las que se trabajará en las preguntas de definición será la compuesta por los artículos de la Wikipedia. Las ventajas de esta decisión son:

- Wikipedia organiza la información por temas, cada documento concierne a un único elemento.
- El primer párrafo de cada documento contiene la descripción del tema.

Los artículos de la Wikipedia están escritos siguiendo las directrices de un lenguaje propio. Puesto que es una enciclopedia editada por los usuarios, no todos los artículos siguen un mismo patrón, no obstante, existe un manual de estilo[3], que permite que la mayoría de los artículos se realicen con unas características similares. La primera información de un artículo es una ficha o tabla, denominada *infobox*, en la que se encuentra la información más relevante sobre el artículo. Esta información es la más díficil de extraer, dado que no existen reglas expresas para su creación y la forma de representarlas. Tras el *infobox,* se encuentra el texto del artículo. En los primeros párrafos, suele encontrarse una breve descripción del tema del artículo, que más adelante se desarrolla. Cuando un artículo lo requiere, debido a su longitud, es dividido en

[3]http://es.wikipedia.org/wiki/Wikipedia:Manual_de_estilo

secciones. A lo largo del artículo también pueden encontrarse imágenes y tablas de contenido.

La organización Wikimedia, pone a disposición de los usuarios, de forma libre, la última versión de las entradas editadas en la Wikipedia[4]. Los documentos se encuentran agrupados en un único archivo XML. Para utilizar esta información es necesario procesarla previamente para eliminar las etiquetas XML y las etiquetas propias del lenguaje en el que se escribe la Wikipedia. Hay que notar que con esta acción, se pierde cierta información de los artículos, principalmente el contenido en tablas, ya que no están escritas de una forma homogénea, lo que dificulta la labor de extraer datos de ellas.

Para procesar el archivo XML con todas las entradas de la Wikipedia, se utiliza una herramienta desarrollada en Daedalus, que extrae los artículos uno a uno, y los almacena en archivos independientes formateados como texto plano.

Sobre la carpeta contenedora de los artículos en texto plano, se realiza una indexación para utilizar en la extracción de documentos. Para ello se utiliza la clase es.daedalus.qa.answersource.lucene.LuceneIndex. Cómo ya se ha reseñado en el análisis del subsistema Documentos Fuentes (3.2.3), los datos que se indexan son: Título, subtítulo -en este caso en blanco-, nombre de la colección, ruta del documento, idioma y texto.

Una vez generados los índices, el sistema puede realizar la búsqueda de la respuesta. No obstante, existe otra tarea que se puede realizar sobre la colección de documentos: el preprocesado lingüístico. La tarea del procesado lingüístico, es sin duda, la que más tiempo consume, y aunque es posible realizarlo en línea, realizarlo fuera de línea minimiza el tiempo de análisis lingüístico.

3.5. Preguntas de definición en el sistema respond.es

En el momento en el que se inicia este proyecto, el sistema respond.es no

[4]Se puede descargar en: http://download.wikipedia.org/eswiki/latest/eswiki-latest-pages-articles.xml.bz2

cuenta con una estrategia específica para las preguntas de tipo definición, a excepción de los acrónimos, que detecta y obtine su expansión apoyado en la herramienta Stilus. Introduce este resultado como respuesta, y busca documentos que contengan el acrónimo y su expansión, para adjuntarlos como soporte.

El sistema sí cuenta con reglas para la identificación de las preguntas de definición, pero éstas son tratadas de la misma forma que las preguntas factuales.

Con esta situación, se decide crear una estrategia específica para las preguntas de definición, añadiendo reglas para el caso. En el capítulo de diseño, se especifican las reglas que se crean con este objetivo, basadas en los patrones de los predicados de definición, utilizando herramientas de análisis sintáctico y semántico.

Capítulo 4

Diseño e Implementación del Sistema

4.1. Especificación del Entorno Tecnológico

Se puede definir el entorno tecnológico del sistema como la tecnología, tanto hardware como software, que se requiere para su desarrollo y funcionamiento. En este apartado se definen los distintos elementos de la infraestructura técnica que dan soporte al sistema. Aunque el desarrollo de este sistema está enmarcado en un proyecto de investigación, y en un principio no existe fecha para la puesta en explotación de éste, también se considera necesario tener un demostrador para mostrar los resultados que se consigan. El planteamiento de este demostrador es que el usuario pueda tener acceso a través de un navegador web, y el sistema se ejecute en el lado del servidor. El hardware y software necesario para el desarrollo y funcionamiento del sistema es:

- Las máquinas cliente, es decir, aquellas que se utilizarán para acceder al sistema, tendrán las siguientes características mínimas:
  - Pantallas con una resolución mínima de 640x480 píxeles
  - Conexión a Internet

    - Navegador web (optimizado para Internet Explorer 6 o superior, Safari, Firefox y Opera)

- Los equipos de desarrollo utilizados tendrán la siguiente configuración software instalada:

    - Eclipse 3.0 con Subversion
    - Stilus
    - Sistema gestor de base de datos MySQL 5.0
    - JDK 6

- La máquina servidor donde estará alojado el sistema junto a la aplicación tendrá las siguientes características mínimas:

    - Acceso remoto seguro (ssh o sftp)
    - Ancho de banda de 1Gbps
    - Linux Ubuntu Server Edition 8.04
    - Servidor Web Apache HTTP 2.2.9
    - Servidor Apache Tomcat
    - Sistema gestor de base de datos MySQL 5.0
    - Stilus
    - JDK 6

4.2. Diseño de Clases

El diseño de clases no se modifica con respecto al mostrado en el análisis. Para la introducción de reglas que respondan a las preguntas de definición se han modificado las clases *RuledBasedAnswerExtractor, AnswerData* y *Ranker*. La modificación realizada sobre éstas se describe en el apartado de implementación.

4.3. Diseño Físico de Datos

Con el objetivo de optimizar el tiempo de respuesta, se ha tomado la decisión de almacenar todas las preguntas realizadas al sistema. De esta forma, si la pregunta introducida ya ha sido contestada, el sistema no volverá a procesarla, sino que devolverá los resultados contenidos en la base de datos.

A continuación se muestra el diagrama entidad/relación de la base de datos implementada. Seguidamente se describen los campos almacenados.

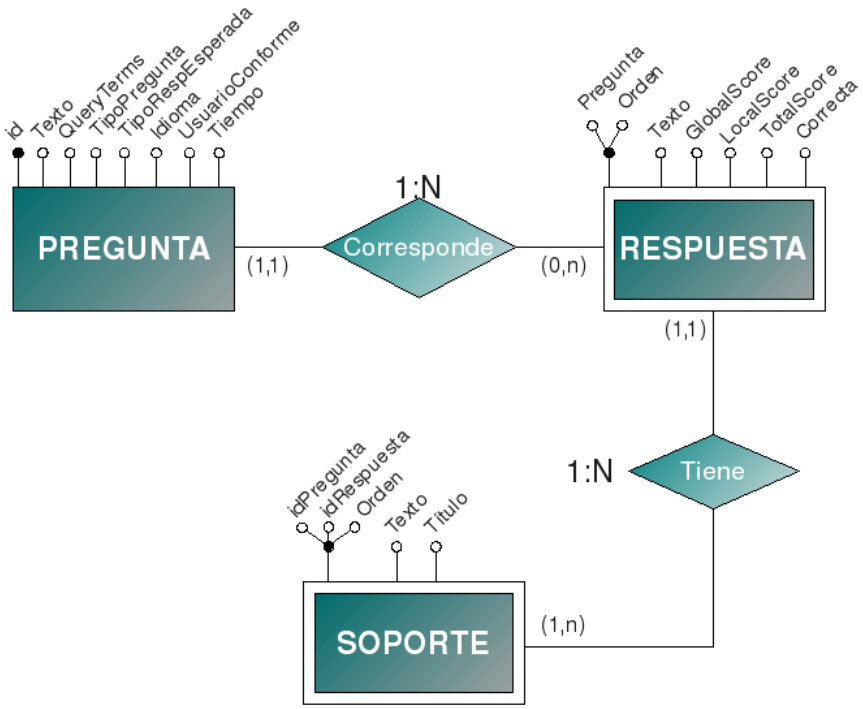

Figura 4.1: Diagrama ER

La base de datos almacena las preguntas, las respuestas y las frases de soporte. De las preguntas se almacena el texto tal cual es introducido por el usuario, los términos de consulta, el tipo de pregunta, el tipo de respuesta esperada y el idioma. Asimismo se almacenarán otros datos con fines de evaluación: una variable que recoge el resultado de si el usuario está conforme (Las opciones son ”si”, ”no” y ”-” si el usuario no valida la pregunta) y el tiempo que tarda el sistema en devolver la pregunta.

Cada pregunta puede tener de 0 a N respuestas. Las respuestas son identificadas por la pregunta a la que responden y la posición. Se almacena: el texto de la respuesta, la puntuación global y local, y un valor introducido por el usuario que identifica a la pregunta como correcta o no. Este último campo permite obtener una realimentación del sistema, que puede resultar muy útil a la hora de evaluarlo y en un futuro disponer de una base de datos con los resultados obtenidos, lo que permitiría utilizar nuevas estrategias antediendo a este valor.

Cada respuesta, a su vez, tiene uno o varios documentos de soporte. De este se almacenan el texto de soporte y el título del documento del cual ha sido extraído el texto.

4.4. Reglas de Definición

En este apartado se describen las reglas que se han diseñado para responder a las preguntas de definición. Antes, se describen los predicados de definición extraídos y las etiquetas semánticas utilizadas en la creación de las reglas.

4.4.1. Predicados de Definición

Una definición puede presentarse de muchas maneras en un texto. En este caso, los textos son los artículos de la Wikipedia. Para extraer los patrones que definan la forma de extraer las respuestas, se ha hecho un análisis de la forma en la que se presentan en la Wikipedia. Como ya se ha explicado, las definiciones se encuentran en el primer o primeros párrafos del artículo. Sobre estos párrafos se han estudiado las distintas formas en las que se encuentra la definición del tema del artículo. Durante el desarrollo del proyecto, se han ido realizando pruebas de las que han surgido predicados nuevos y se han añadido a las reglas.

A continuación se presentan en el Cuadro 4.1 los predicados de definición extraídos. Se muestra el patrón que siguen y a continuación un ejemplo[1], donde el

[1]Todos los ejemplos han sido extraídos de la Wikipedia en español: http://es.wikipedia.org

concepto está resaltado y la definición subrayada.

Verbo ser: <CONCEPTO> *verbo ser* <DEFINICIÓN>

La **globalización** *es* un proceso fundamentalmente económico que consiste en la creciente integración de las distintas economías nacionales en una única economía de mercado mundial.

Se define como: <CONCEPTO> *se define como* <DEFINICIÓN>

El **newton** *se define como* la fuerza necesaria para proporcionar una aceleración de 1 m/s^2 a un objeto de 1 kg de masa.

Se denomina a: *se denomina [como]* <CONCEPTO> *a* <DEFINICIÓN>

En la jerga militar *se denomina* **fuego amigo** *a* los disparos provenientes del propio bando.

Se describe como: <CONCEPTO> *se describe como* <DEFINICIÓN>

Tarasca *se describe como* una especie de dragón con seis cortas patas parecidas a las de un oso, un torso similar al de un buey con un caparazón de tortuga a su espalda y una escamosa cola que terminaba en el aguijón de un escorpión.

Se conoce como: *se conoce como* <CONCEPTO> *a/al* <DEFINICIÓN>

Se conoce como **risoterapia** *a* una estrategia o técnica psicoterapéutica tendiente a producir beneficios mentales y emocionales por medio de la risa.

Se refiere a: <CONCEPTO> *se refiere a* <DEFINICIÓN>

Un **astro** *se refiere a* cualquier cuerpo celeste con forma definida.

Se entiende por: *se entiende por* <CONCEPTO> *[a/al]* <DEFINICIÓN>

Se entiende por **firmamento** la bóveda celeste en que se encuentran aparentemente los astros.

Cuadro 4.1: Predicados de Definición

Aparte de los predicados descritos, hay que tener en cuenta que, en los artículos de personas, la definición puede presentarse de la siguiente forma:

<Nombre Completo> (<Fecha de Nacimiento> - <Fecha de Muerte>)[.|,]<Definición>

Este tipo de definiciones es tratado mediante las etiquetas semánticas. En el siguiente apartado se explica la clasificación semántica utilizada y la forma en la que se aplican las etiquetas semánticas para responder las preguntas de definición sobre personas.

4.4.2. Etiquetas Semánticas

El etiquetado semántico permite obtener información de una palabra dada como su temática, tipo de entidad, remisión a otras entidades o información geográfica, así como ampliar una cadena de texto añadiendo términos relacionados mediante sinonimia , antonimia o palabras relacionadas semánticamente. El uso de éstas en un SBR permite dos ventajas importantes: expandir los términos de consulta mediante sinonimia y detectar entidades esperadas como respuesta y compararlas con texto de los documentos fuente.

En concreto, para las preguntas de definición, las etiquetas semánticas son especialmente útiles a la hora de encontrar definiciones de personas. Normalmente, la definición de una persona viene dada por la vocación o cargo que desempeña o ha desempeñado. La clasificación semántica utilizada en el sistema respond.es, está basada en la de Sekine (Figura 4.2).

Las entidades utilizadas para la extracción de respuestas de definición son POSITION_TITLE y VOCATION. Cuando una pregunta contenga la entidad PERSON, se buscará alguna de estas dos entidades en los primeros párrafos del texto para extraer la respuesta.

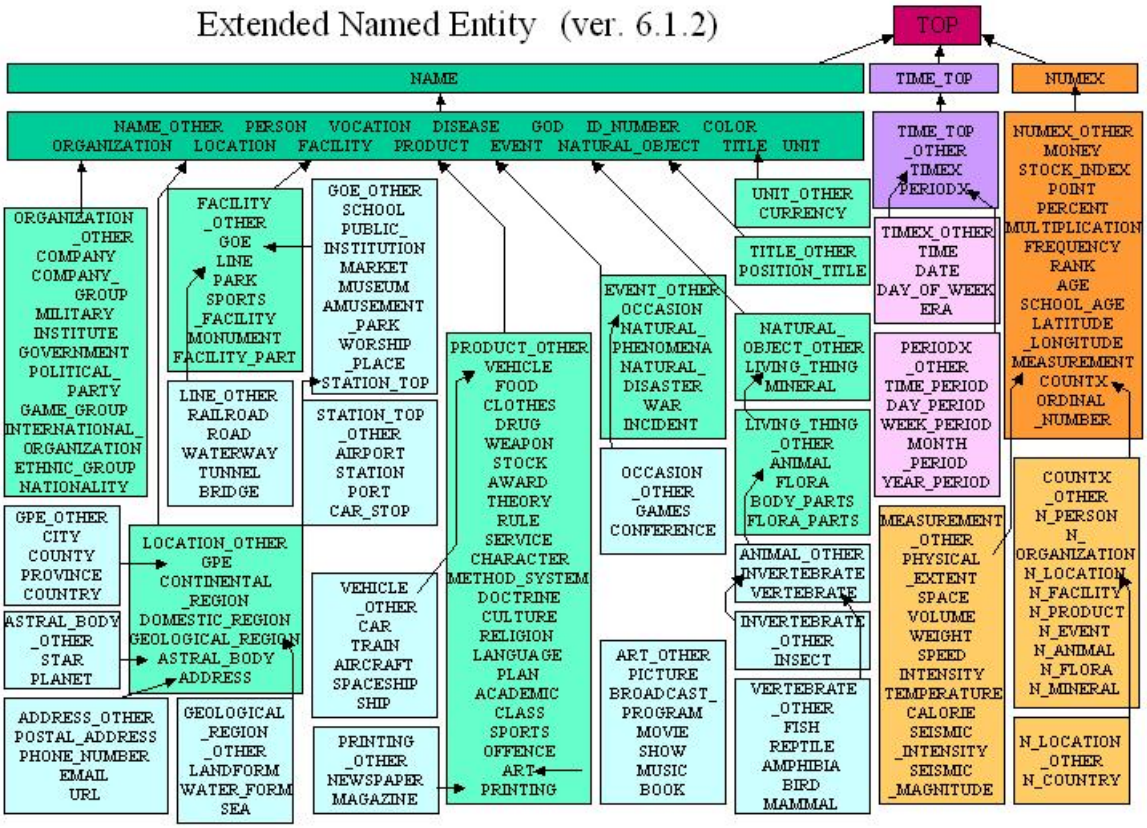

Figura 4.2: Clasificación de Entidades Semánticas de Sekine

4.4.3. Creación de Reglas

Teniendo en cuenta la estructura en la que se presentan las definiciones en la Wikipedia, y tras el análisis de los patrones de definición, se seleccionan las reglas para la implementación. Las reglas se han realizado contando con herramientas morfosintácticas, de forma que, al referirnos a un verbo, estamos haciendo referencia a su lema y cualquier conjugación del verbo generará una coincidencia.

Regla 1 Si se encuentra "ser" $\Rightarrow$ Extraer pregunta desde el verbo ser hasta primer símbolo de puntuación o fin de frase

Regla 2 Si se encuentra "se" $\wedge$ ("definir" $\vee$ "describir") $\Rightarrow$ Extraer respuesta desde el siguiente término después del primer "como" hasta el primer

símbolo de puntuación o fin de frase.

Regla 3 Si se encuentra "se" $\wedge$ ("denominar" $\vee$ "conocer" $\vee$ "referir") $\Rightarrow$ Extraer respuesta desde el siguiente término después del primer "a" $\vee$ "al" hasta el primer símbolo de puntuación o fin de frase.

Regla 4 Si se encuentra "se" $\wedge$ "entender" $\wedge$ ("por" $\vee$ "como) $\Rightarrow$ Extraer respuesta desde la siguiente posición del termino de consulta hasta el primer símbolo de puntuación o fin de frase.

Regla 5 Si la consulta contiene etiqueta PERSON $\Rightarrow$ Extraer respuesta desde la primera etiqueta (PERSON_TITLE $\vee$ VOCATION) hasta el primer símbolo de puntuación o fin de frase.

4.5. Implementación

4.5.1. Reglas Implementadas

El sistema respond.es cuenta con un motor de reglas. Las reglas de clasificación de preguntas y extracción de respuestas están escritas siguiendo el formato de la lógica de predicados, de forma que para entender una regla, basta con leerla. La inclusión, exclusión y modificado de las reglas es también más sencilla.

A la hora de compilación, mediante la herramienta Apache Ant[2], las reglas son traducidas a java, utilizando para ello expresiones regulares descritas en archivos de configuración XML.

[2]http://ant.apache.org/

```
//¿Qué es el Parlamento Europeo?
REGLA("definition")
    IDENTIFICADOR_FUENTE("wiki") Y
    TIPO_PREGUNTA_EXISTENCIAL("DEFINITION") Y
    (OBTENER_NUMERO_FRASE() <5 O
    FRASE_CONTIENE_ALGUN_LEMA_DE_PREGUNTA() Y
    FRASE_CONTIENE_ENTIDAD_COMPATIBLE_CON_ESPERADA()) Y
    LEMA_EXISTENCIAL(N,"ser")
    ENTONCES
        SI APARECE_EN_FRASE(POS_PRIMERA_PALABRA_I_DESPUES_DE(N,"\\.|,|;")-1)
        ENTONCES
            pos1=POS_PRIMERA_PALABRA_I_DESPUES_DE(N,"\\.|,|;")-1
        OSINO
            pos1=POS_ULTIMO_TOKEN();
        FIN
    SI !LEMAS_EXISTENCIAL_CONTENIDOS_EN_PREGUNTA(N,pos1)
    ENTONCES
        SI NOMBRE_DOCUMENTO_APARECE_EN_PREGUNTA()
        ENTONCES
            EXTRAER_RESPUESTA_CAMBIANDO_NIVEL_CONFIANZA(N,pos1,"DEFINITION_ARTICLE_SAME_NAME");
        OSI NOMBRE_DOCUMENTO_APARECE_PARCIALMENTE_EN_PREGUNTA()
        ENTONCES
            EXTRAER_RESPUESTA_CAMBIANDO_NIVEL_CONFIANZA(N,pos1,"DEFINITION_ARTICLE_PARTIAL_NAME");
        OSINO
            EXTRAER_RESPUESTA(N,pos1);
        FIN
    FIN
FIN
```

Cuadro 4.2: Implementación de Regla 1

En los Cuadros 4.2, 4.3, 4.4 se muestra la implementación de las reglas. Las reglas 2, 3 y 4 han sido agrupadas en una única expresión como se puede ver en el Cuadro 4.3. Todas las reglas realizan la comprobación de que el documento fuente sea la Wikipedia, y que la pregunta haya sido clasificada de tipo definición. También se comprueba que el número de frase sea menor que cinco, puesto que las definiciones se encuentran en las primeras frases del artículo. Las reglas seleccionan la palabra de inicio y de fin para luego extraerlas.

Al realizar la extracción, si el título del artículo al que pertenece la frase aparece en la pregunta (ya sea de forma parcial o completa), la pregunta es extraída asignándole un valor de confianza. Este valor sirve para darle mayor prioridad a la hora de ordenar la respuesta. En el siguiente apartado se explica como se lleva a cabo este proceso.

```
REGLA("definition")
    IDENTIFICADOR_FUENTE("wiki") Y
    TIPO_PREGUNTA_EXISTENCIAL("DEFINITION") Y
    (OBTENER_NUMERO_FRASE() <5 O
    FRASE_CONTIENE_ALGUN_LEMA_DE_PREGUNTA() Y
    FRASE_CONTIENE_ENTIDAD_COMPATIBLE_CON_ESPERADA()) Y
    LEMA_EXISTENCIAL(N,"se") Y
    (
    ((LEMA_EXISTENCIAL(N+1,"definir|describir") Y
    APARECE_EN_FRASE(pos1=POS_PRIMER_LEMA_EXISTENCIAL_DESPUES_DE(N, "como")+1)
    ) O (
    (LEMA_EXISTENCIAL(N+1,"denominar|conocer|referir") Y
    APARECE_EN_FRASE(pos1=POS_PRIMER_LEMA_EXISTENCIAL_DESPUES_DE(N, "a|al")+1)
    ) O (
    (LEMA_EXISTENCIAL(N+1,"entender") Y
    LEMA_EXISTENCIAL(N+2,"por|como") Y
    APARECE_EN_FRASE(pos1=POS_PRIMER_LEMA_EXISTENCIAL_DESPUES_DE(N, "por|como")+2)
    )
    ENTONCES
        SI LEMAS_EXISTENCIAL_CONTENIDOS_EN_PREGUNTA(N,pos1)
        ENTONCES
            pos1 = pos1 + 1;
        FIN
        SI APARECE_EN_FRASE(POS_PRIMERA_PALABRA_I_DESPUES_DE(N,"\\.|,|;"))
        ENTONCES
            pos2=POS_PRIMERA_PALABRA_I_DESPUES_DE(N,"\\.|,|;")-1
        OSINO
            pos2=POS_ULTIMO_TOKEN();
        FIN
        SI NOMBRE_DOCUMENTO_APARECE_EN_PREGUNTA()
        ENTONCES
            EXTRAER_RESPUESTA_CAMBIANDO_NIVEL_CONFIANZA(pos1,pos2,"DEFINITION_ARTICLE_SAME_NAME");
        OSI NOMBRE_DOCUMENTO_APARECE_PARCIALMENTE_EN_PREGUNTA()
        ENTONCES
            EXTRAER_RESPUESTA_CAMBIANDO_NIVEL_CONFIANZA(pos1,pos2,"DEFINITION_ARTICLE_PARTIAL_NAME");
        OSINO
            EXTRAER_RESPUESTA(N,pos1);
        FIN
    FIN
FIN
```

Cuadro 4.3: Implementación de Reglas 2, 3 y 4

```
//¿Quién es Peter Sellers?
REGLA("definition")
    IDENTIFICADOR_FUENTE("wiki") Y
    TIPO_PREGUNTA_EXISTENCIAL("DEFINITION") Y
    (OBTENER_NUMERO_FRASE() <5 O FRASE_CONTIENE_ALGUN_LEMA_DE_PREGUNTA()) Y
    ENTIDAD_ESPERADA_EXISTENCIAL("@PERSON@") Y
    !TIENE_ETIQUETA_GEO(N) Y COINCIDE_NUMERO(N) Y
    (
    ABSTRACCION_EXISTENCIAL_COMPATIBLE_CON_ESPERADAS(N) Y
    ENTIDAD_EXISTENCIAL(N,"@PERSON@") O ABSTRACCION_EXISTENCIAL(N,"inst") Y
    (ENTIDAD_EXISTENCIAL(N,"@OTHER_ENTITY@VOCATION@") O
    ENTIDAD_EXISTENCIAL(N,"@OTHER_ENTITY@TITLE@POSITION_TITLE@"))
    )
    ENTONCES
        SI APARECE_EN_FRASE(POS_PRIMERA_PALABRA_I_DESPUES_DE(N,"\\.|,|;"))
        ENTONCES
            pos1=POS_PRIMERA_PALABRA_I_DESPUES_DE(N,"\\.|,|;")-1
        OSINO
            pos1=POS_ULTIMO_TOKEN();
    FIN
    SI !LEMAS_EXISTENCIAL_CONTENIDOS_EN_PREGUNTA(N,pos1)
    ENTONCES
        SI NOMBRE_DOCUMENTO_APARECE_EN_PREGUNTA()
        ENTONCES
            EXTRAER_RESPUESTA_CAMBIANDO_NIVEL_CONFIANZA(N,pos1,"DEFINITION_ARTICLE_SAME_NAME");
        OSI NOMBRE_DOCUMENTO_APARECE_PARCIALMENTE_EN_PREGUNTA()
        ENTONCES
            EXTRAER_RESPUESTA_CAMBIANDO_NIVEL_CONFIANZA(N,pos1,"DEFINITION_ARTICLE_PARTIAL_NAME");
        OSINO
            EXTRAER_RESPUESTA(N,pos1);
        FIN
    FIN

FIN
```

Cuadro 4.4: Implementación de Regla 5

4.5.2. Clases Modificadas

A lo largo del desarrollo de la práctica se han realizado modificaciones sobre el sistema respond.es para mejorar su rendimiento. En cuanto a las preguntas de definición, aparte de las reglas creadas, se ha modificado la forma de ordenación, tratándolas de forma específica. Se han definido dos nuevos niveles de confianza en la clase *AnswerData*. El nivel de confianza define la probabilidad de que la respuesta hallada sea correcta. La probabilidad de que un artículo con el mismo

nombre que el término de búsqueda contenga la definición, es a priori muy alta. El nivel de confianza definido en la clase *AnswerData* queda de la siguiente forma:

- AnswerData
  - ConfidenceLevel: {
    - SURE,
    - **DEFINITION_ARTICLE_SAME_NAME**,
    - **DEFINITION_ARTICLE_PARTIAL_NAME**,
    - FACTUAL_KNOWN_ENTITY_CONTAINS_QUESTION_NP_ MEETS_TIME_RESTRICTION,
    - FACTUAL_KNOWN_ENTITY_CONTAINS_QUESTION_NP,
    - FACTUAL_KNOWN_ENTITY_MEETS_TIME_RESTRICTION,
    - FACTUAL_KNOWN_ENTITY,
    - FACTUAL_UNKNOWN_ENTITY_CONTAINS_QUESTION_NP,
    - FACTUAL_UNKNOWN_ENTITY };

Al extraer la pregunta, como se puede ver en la implementación de las reglas, si el título del artículo es el mismo que el término de búsqueda, se establece la confinaza **DEFINITION_ARTICLE_SAME_NAME**, si el título contiene sólo algunas de las palabras que conforman los términos de consulta, la confianza asignada es **DEFINITION_ARTICLE_PARTIAL_NAME**. Si alguno de estos dos índices son encontrados a la hora de ordenar las respuestas, la puntuación asignada se calcula con la siguiente función:

$$Score = 0{,}9 * (1 - min(1, 0{,}2 * \#lineaEnDoc)) + 0{,}1 * (relevanciaLucene)$$

De esta forma, se tienen en cuenta las cinco primeras frases, obteniendo mejor puntuación, cuanto más cerca del inicio esté la frase.

Capítulo 5

Evaluación del Sistema

En este capítulo se evalúan los resultados del sistema a las preguntas de definición. Primero se muestra los resultado obtenidos tras la evaluación subjetiva las preguntas de definición extraídas de la colección de preguntas del foro QA@CLEF 2008 (introducido en el estado de la cuestion 2.4.1) que se muestran en el Anexo B.2. A continuación se muestran los resultados obtenidos en el foro QA@CLEF 2008 y una comparación con el resto de los sitemas participantes en lengua española. Asimismo se muestra la comparación entre los resultados del sistema respond.es en QA@CLEF 2007 y 2008.

5.1. Resultados de la Evaluación

Las respuestas obtenidas por el sistema han sido calificadas en alguna de las siguientes cinco categorías:

Correctas	Respuesta correcta, con documento de soporte.
Aproximadas	Respuesta no es completa o es inexacta.
En frase	Respuesta errónea pero se puede extraer de la frase de soporte
En documento	Respuesta errónea pero se puede extraer del documento de soporte
Erróneas	Respuesta errónea y no se encuentra en el documento de soporte

El sistema identifica 21 preguntas de definición, el resultado de la evaluación de estas preguntas es:

- 17 respuestas correctas
- 1 respuesta aproximada,
- 3 respuestas erróneas

Para evaluar las respuestas se utilizan los índices MRR (a profundidad 1 y 3) y CWS. Los resultados obtenidos son:

$$CWS = 0,9487151$$

	MRR-1	MRR-3
Respuestas	0,8095238	0,8253968
Documentos	0,8571429	0,8809524

Cuadro 5.1: MRR de las tres primeras respuestas

Del primer dato se puede concluir que las preguntas de definición del sistema tienen un alto índice de confianza. Del segundo dato podemos obtener que nuestro sistema **responde correctamente a un 80,95 %** de las preguntas de definición realizadas. Si ampliamos el rango hasta tres preguntas, el porcentaje asciende a un 82,53 %. También se puede deducir del valor del MRR para los documentos, que en la mayoría de las preguntas, si se cuenta con el documento que contiene la respuesta, el sistema es capaz de extraerla.

Se ha analizando las respuestas erróneas, comparando con las respuestas correctas proporcionadas por CLEF. Los errores se han producido por :

- La respuesta no está en la Wikipedia, sino en la colección de artículos EFE.
- Falta de etiqueta semántica.
- El nombre de la pregunta difiere con el nombre del artículo y no se resuelve bien el alias.

5.1.1. Resultados del sistema 2008 sobre preguntas 2007

El sistema respond.es también fue evaluado con las preguntas de QA@CLEF 2007. La misma colección de preguntas -que se puede ver en el Anexo B.1- se ha respondido en el sistema con las nuevas reglas implementadas para las preguntas de definición. Los resultados de la evaluación de las preguntas de definición en QA@CLEF 2007, con la versión de respond.es de 2007 y 2008 se muestran en la Figura 5.1.

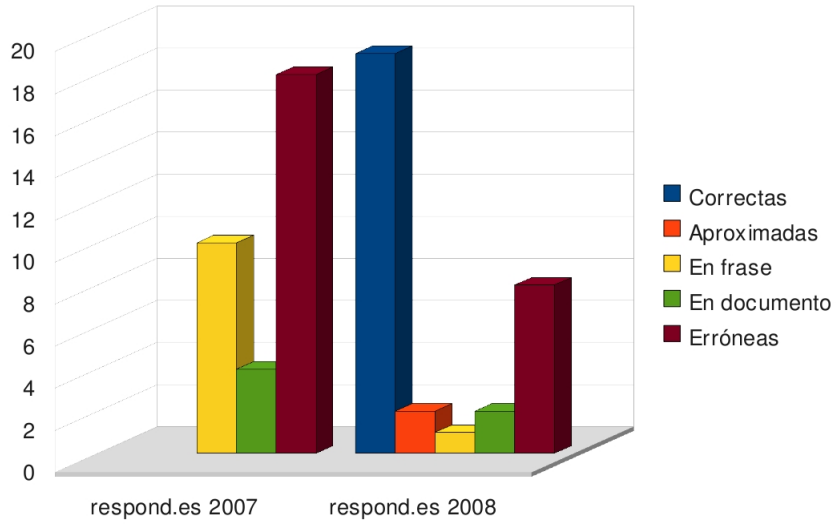

Figura 5.1: Resultados respond.es 2007 con motor 2007 y 2008

Se aprecia una sustancial mejora en los resultados de 2008 frente a los de 2007. El nuevo sistema responde correctamente a 19 preguntas, mientras que el sistema anterior no conseguía ninguna respuesta correcta. Además, se ha reducido el número de respuestas erróneas de 18 a 8.

5.2. Resultados QA@CLEF 2008

Los resultados obtenidos en el foro QA@CLEF permiten medir de una forma objetiva las respuestas obtenidas. La forma de evaluar del foro difiere ligeramente de la expuesta anteriormente. Los valores que pueden tomar las respuestas son:

Correcto, Incorrecto, Inexacto y Sin soporte. Los resultados que se obtuvieron fueron:

- Para un total de 19 definiciones:
  - 14 Correctas
  - 3 Incorrectas
  - 1 Inexacta
  - 1 Sin soporte
  - Precisión calculada sobre definiciones = 14/19 = 73.684 %

El primer dato que difiere de nuestra evaluación es el número de respuestas de definición. Nuestro sistema detecta 21 respuestas de definición, mientras que en el foro sólo detectan 19 de este tipo. Las dos preguntas en cuestión son *¿Qué es el oricalco?* y *¿Quién era Edgar P. Jacobs?.* Analizando las respuestas otorgadas por el foro, clasifica las dos preguntas como factuales. En principio, este problema parace mas un error por parte del sistema de corrección del foro QA@CLEF que un error de nuestro sistema.

Nuestra evaluación detecta 17 preguntas correctas, frente a las 14 del foro. A parte de las dos respuestas que no son clasificadas como definición por CLEF y si por nuestro sistema -lo cual llevaría al sistema a 15 preguntas correctas sobre 19-, una respuesta que nosotros calificamos como correcta, en el foro es calificada como sin soporte. Esto es debido a que la frase de soporte contiene sólo la respuesta sin hacer referencia a la pregunta. Analizando la frase de soporte, se comprueba que ésta sólo contiene la mrespuesta, por lo que habría que estudiar los casos en los que la frase de soporte fuera demasiado corta y concatenar la frase anterior a la propia respuesta para devolver como frase de soporte.

5.2.1. Comparación de Resultados

El foro QA@CLEF también nos ofrece los resultados de otros sistemas. Estos se muestran a continuación en forma de gráfico comparativo. Aparte de respond.es los sistemas que participaron en español fueron: inaoe, prib y qaua.

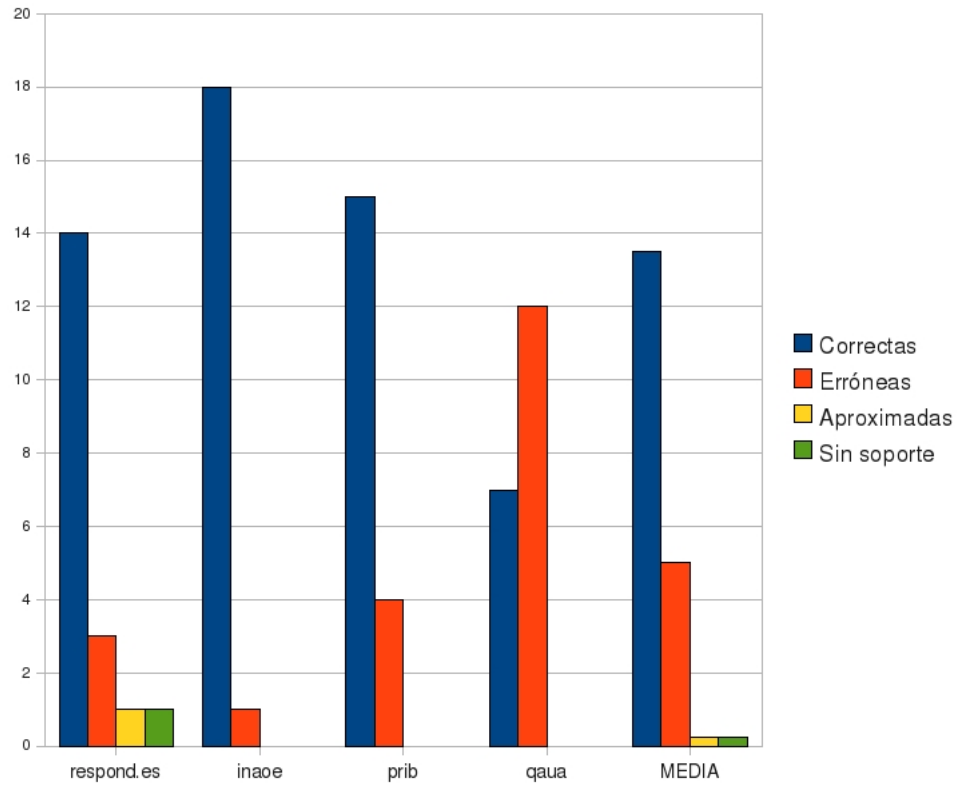

Figura 5.2: Comparación de resultados entre sistemas QA@CLEF 2008

El sistema respond.es se encuentra por encima de la media de aciertos y por debajo de la media de errores. Se puede apreciar que los resultados no se encuentran muy lejos de los resultados de los mejores sistemas.

Por último se muestra una comparativa entre los resultados del sistema respond.es en QA@CLEF 2008 y 2007. El número de preguntas de definición en el año 2007 fue de 32 mientras que en 2008 sólo ha habido 19 preguntas de este tipo.

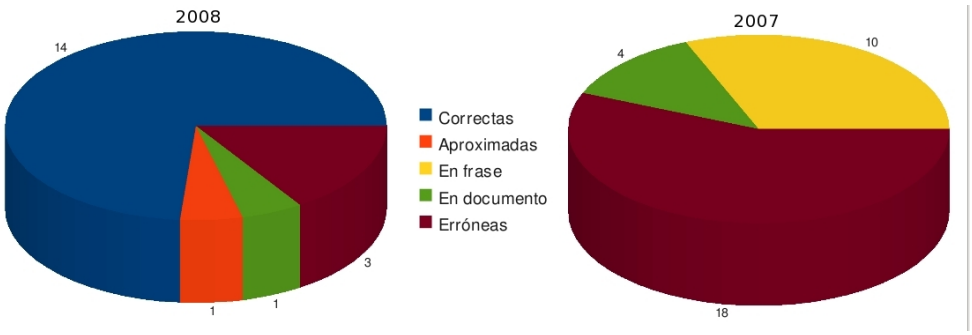

Figura 5.3: Evaluación QA@CLEF 2008 vs 2007

Se puede apreciar una gran mejoría entre los resultados de 2007 y 2008. Hay que tener en cuenta que en el sistema de 2007 no se había tomado la decisión de buscar las preguntas de definición exclusivamente en artículos de la Wikipedia, se puede ver que de un total de 32 preguntas, 18 preguntas ni siquiera encontraron el artículo donde se encontraba la definición.

Capítulo 6

Conclusiones y Líneas Futuras

6.1. Conclusiones

De la idea de simplificar al máximo la búsqueda a los usuarios, nacen los SBR. Estos sistemas, a pesar de que tengan una trayectoria de casi medio siglo, se han puesto en auge con la inclusión de herramientas de PLN. La inclusión de estas técnicas mostró en un principio resultados muy buenos, pero cuando se han intentado implantar en sistemas abiertos, los problemas propios del entendimiento del lenguaje, tales como la ambigüedad o la anáfora y la complejidad propia de cada lenguaje, ha hecho que sea una tarea todavía inalcanzable. El reconocimiento del lenguaje natural requiere un gran conocimiento sobre el mundo real. Además, este avance, al estar fuertemente ligado con el idioma, no se ha producido a nivel global, haciendo que existan grandes diferencias entre idiomas y no sean exportables los avances de unos idiomas a otros, y, aunque muchas reglas del inglés pueden traducirse al español, otras tantas son propias de cada idioma. Las herramientas de PLN en español ya han superado su infancia, pero todavía no han alcanzado el nivel de madurez que permitan llegar a un completo entendimiento del lenguaje natural.

El sistema respond.es se encuentra en este camino, plantando los cimientos sobre la cual crear un SBR. El sistema es sólido y permite la inclusión de nuevos módulos, como por ejemplo otro motor de búsqueda, ampliar las colecciones

de búsqueda e implementar nuevas reglas. Además, mediante el módulo de evaluación del sistema se obtienen datos facilmente sobre si las modificaciones introducidas mejoran o empeoran el sistema.

Partiendo de esta idea se describe una estrategia sobre un dominio cerrado, analizando los artículos de la Wikipedia y creando patrones que describan los predicados de definición. El hecho de que sea un dominio cerrado significa, que esta estrategia nos permitiría en el mejor de los casos responder a 521.115 preguntas de definición -estos son los artículos de la Wikipeda a fecha de Octubre de 2009-, no obstante, este parece un punto de partida para la creación de un sistema que responda a las preguntas de definición, pudiendo crear sin mucha dificultad reglas para adaptarse a otros dominios o incluso con la incorporación de herramientas más potentes poder realizar la búsqueda en un dominio abierto.

La inclusión de las reglas para las preguntas de definición, ha conseguido que el sistema pasase de no contestar correctamente a ninguna pregunta a obtener un **73,68 %** de respuestas correctas en la evaluación realizada en el foro QA@CLEF 2008 y un **80,95 %** en los resultados de la evaluación propia. Estos resultados evidencian la necesidad del sistema de reglas específicas para las preguntas de definición, y establecen una buena base sobre la que ampliar estas reglas.

6.2. Líneas Futuras

Existen dos ramas a mejorar: aumentar los resultados correctos y minimizar el tiempo de respuesta.

Una de las medidas que se podría adoptar para mejorar los tiempos de respuesta es la anotación predictiva. Esta consiste en buscar pares conceptos definición sobre la colección de documentos y almacenarlos. Otra medida que se puede tomar para optimizar el tiempo de respuesta es el preprocesado lingüístico. Aunque ya se realiza en el sistema propuesto, el acceso a los datos ya analizados -almacenados en un fichero- no mejora mucho los tiempos de respuesta. La mejora debería incidir en este aspecto, almacenando el procesado lingüístico de la colección de ficheros fuentes en bases de datos, de forma que para acceder a los análisis, se hiciera mediantes consultas a la base de datos.

Para mejorar el porcentaje de respuestas correctas, se puede sopesar la utilización de la información contenida en los infobox de la Wikipedia. Esta medida sería más beneficiosa para las preguntas factuales que para las de definición, puesto que la información contenida en las tablas es más concreta, aun así se para las definiciones de personas sería una buena solución. Asimismo, aunque la política de la Wikipedia es crear artículos independientes para cada concepto, existen casos, como la descripción de personajes de libros o películas, en los cuales las definiciones pueden encontrarse en tablas. Procesando las tablas y se pueden construir nuevas estrategias para la obtención de respuestas. El problema con el que nos encontraríamos en ese caso es que las tablas y los infobox no tienen una estructura homogeneizada, lo que dificulta la extracción de información. Sobre este tema existe un proyecto denominado DBPedia que encauza sus esfuerzos en extraer información estructurada de la Wikipedia.

Con el fin de encontrar nuevos patrones para las preguntas de definición, se puede crear una herramiente que analice los primeros párrafos de todos los artículos de la Wikipedia y obtener aquellos que no se correspondan con ninguno de los patrones ya elaborados. Estudiar los predicados de definición sobre esta nueva colección dará información para la creación de nuevos patrones.

Bibliografía

[Amit, 2001] Singhal, A. (2001). *Modern Information Retrieval: A Brief Overview*. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering.

[Denicia-Carral, 2007] Denicia-Carral, M. C. (2007). *Respondiendo Preguntas de Definición mediante el Descubrimiento de Patrones Léxicos*. Tesis INAOE.

[Hernández-Rubio, 2008] Hernández-Rubio, G. (2008). *Recuperación de Pasajes Orientada a la Resolución de Preguntas con Restricción Temporal*. Tesis INAOE.

[Lehnert, 1977] Lehnert, W. (1977). *Human and Computational Question Answering*. Cognitive Science: A Multidisciplinary Journal,1:1,47 — 73.

[Lin et al., 2003] Lin, J., Katz, B. (2003). *Question answering from the web using knowledge annotation and knowledge mining techniques*. New York, USA. ACM Press.

[Llopis et al. 2002] Llopis, F., Vicedo J.L., Ferrández, A. (2002). *Selección de pasajes para facilitar el proceso de búsqueda de respuestas*. PLN, Num. 29 , pp. 273-280.

[Magnini et al., 2003] Magnini, B., Romagnoli, S., Vallin, A., Herrera, J., Peñas, A., Peinado, V., Verdejo, F., de Rijke, M. (2003). *The Multiple Language Question Answering Track*. CLEF 2003.

[de Pablo, 2003] de Pablo, C. (2003). *Aprendizaje Semisupervisado de Patrones para la Extracción de Respuestas en Sistemas de Búsqueda de Respuestas*. Anteproyecto de Tesis Doctoral. Universidad Carlos III.

[Salton, 1989] Salton, G. A. (1989). *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison Wesley. New York.

[Sekine et al., 2002] Sekine, S., Sudo, K., Nobata, C. (2002). *Extended named entity hierarchy*. LREC-2002 Conference.

[Vicedo et al., 2003] Vicedo, J. L., Rodríguez, H., Peñas, A., Massot, M. (2003). *Los sistemas de Búsqueda de Respuestas desde una perspectiva actual*. Revista de la Sociedad Española para el Procesamiento del Lenguaje Natural. Num.31.

Bibliografía Complementaria

Vicedo, J.L. (2003). *La Búsqueda de Respuestas: Estado Actual y Perspectivas de Futuro*. Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. Num. 20 , pp. 34-52.

Srihari, R., Li, W. (1999). *Information Extraction Supported Question Answering*.

Castelo, D., Isi, J., Martínez, S. (2006). *WebQA - Respuesta automática a preguntas*. Proyecto de Grado. Universidad de la República.

Olaziregi, G. (2008). *Modelado de Conocimiento Emocional para Interacción Natural*. Proyecto Fin Carrera. Universidad Carlos III

Apéndice A

Diagramas de Clases Detallados

A continuación se muestran los diagramas de clases en detalle, con los métodos y atributos de cada clase y sus relaciones y dependencias.

A.1. Análisis de la Pregunta

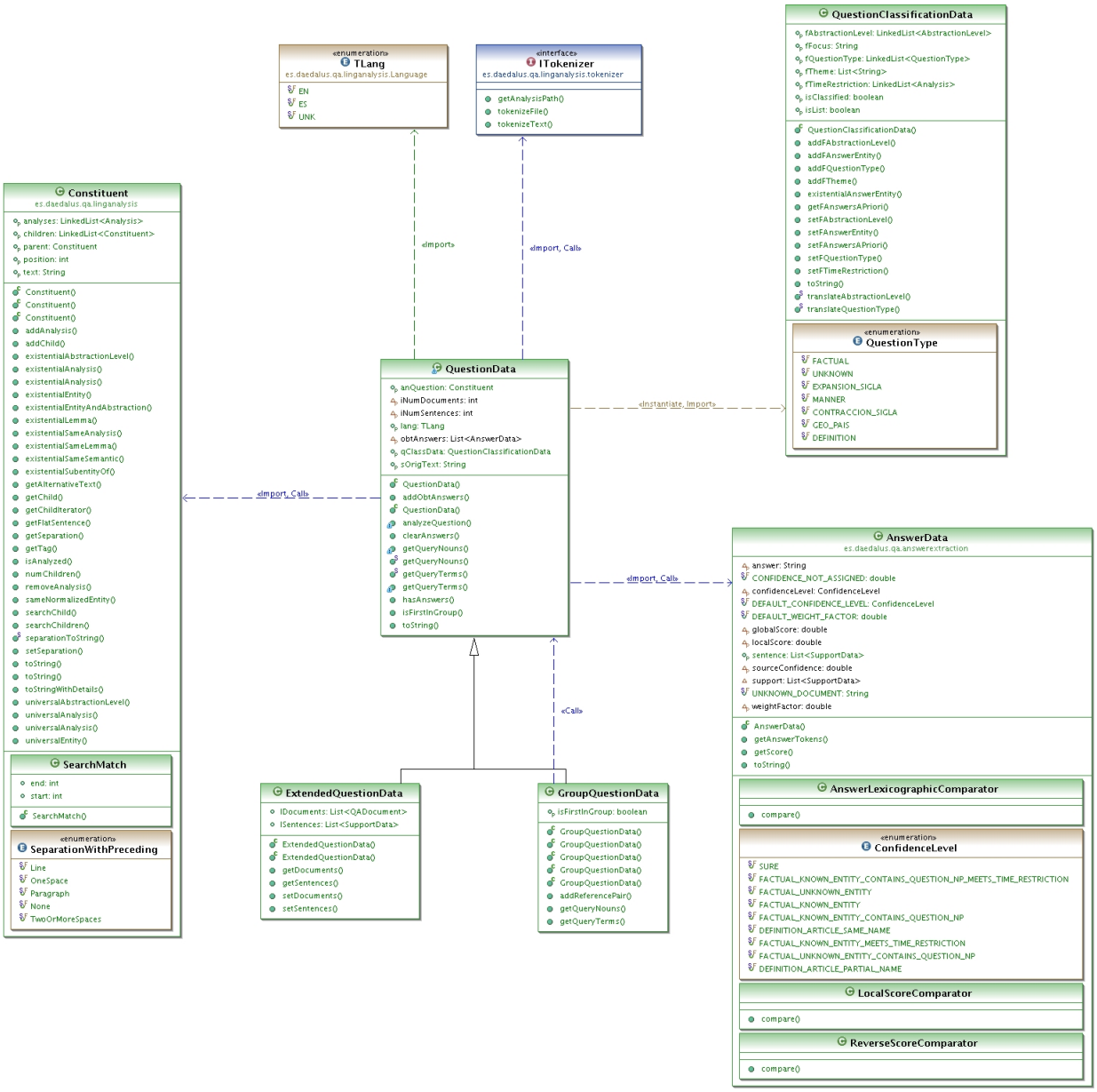

Figura A.1: Diagrama detallado *questionanalysis* 1/2

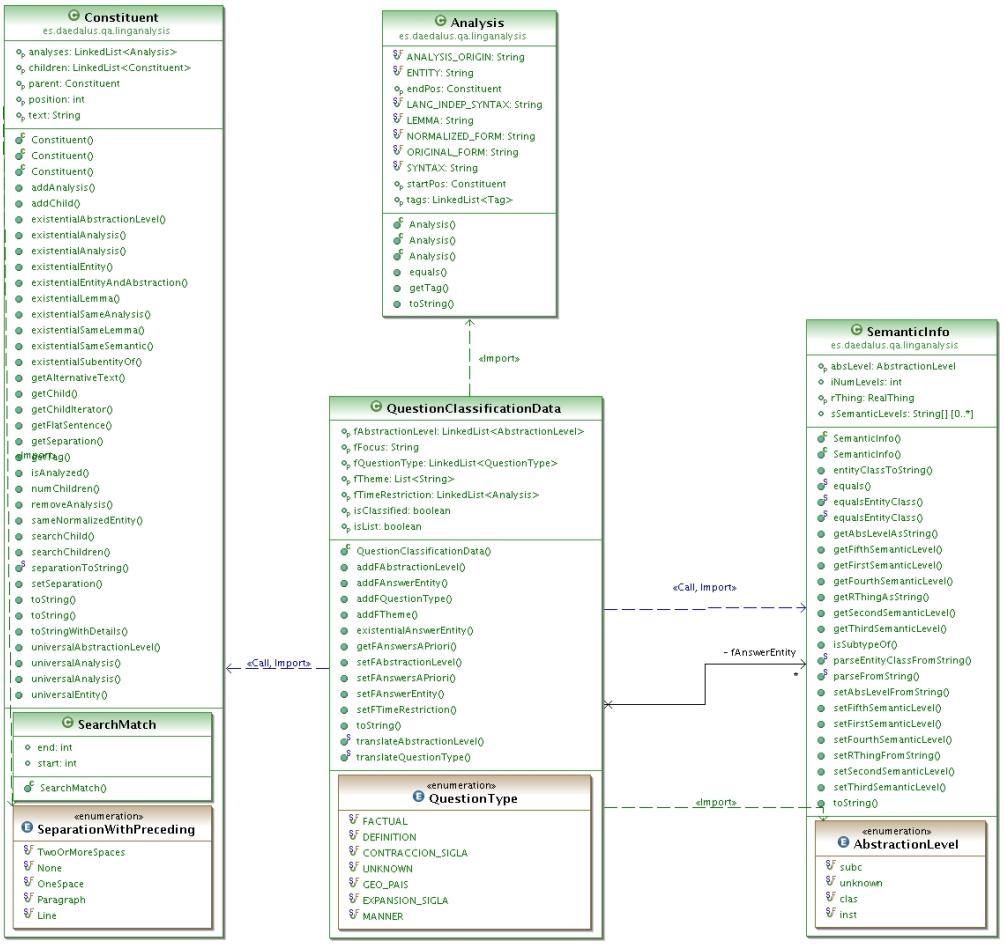

Figura A.2: Diagrama detallado *questionanalysis* 2/2

A.2. Documentos Fuente

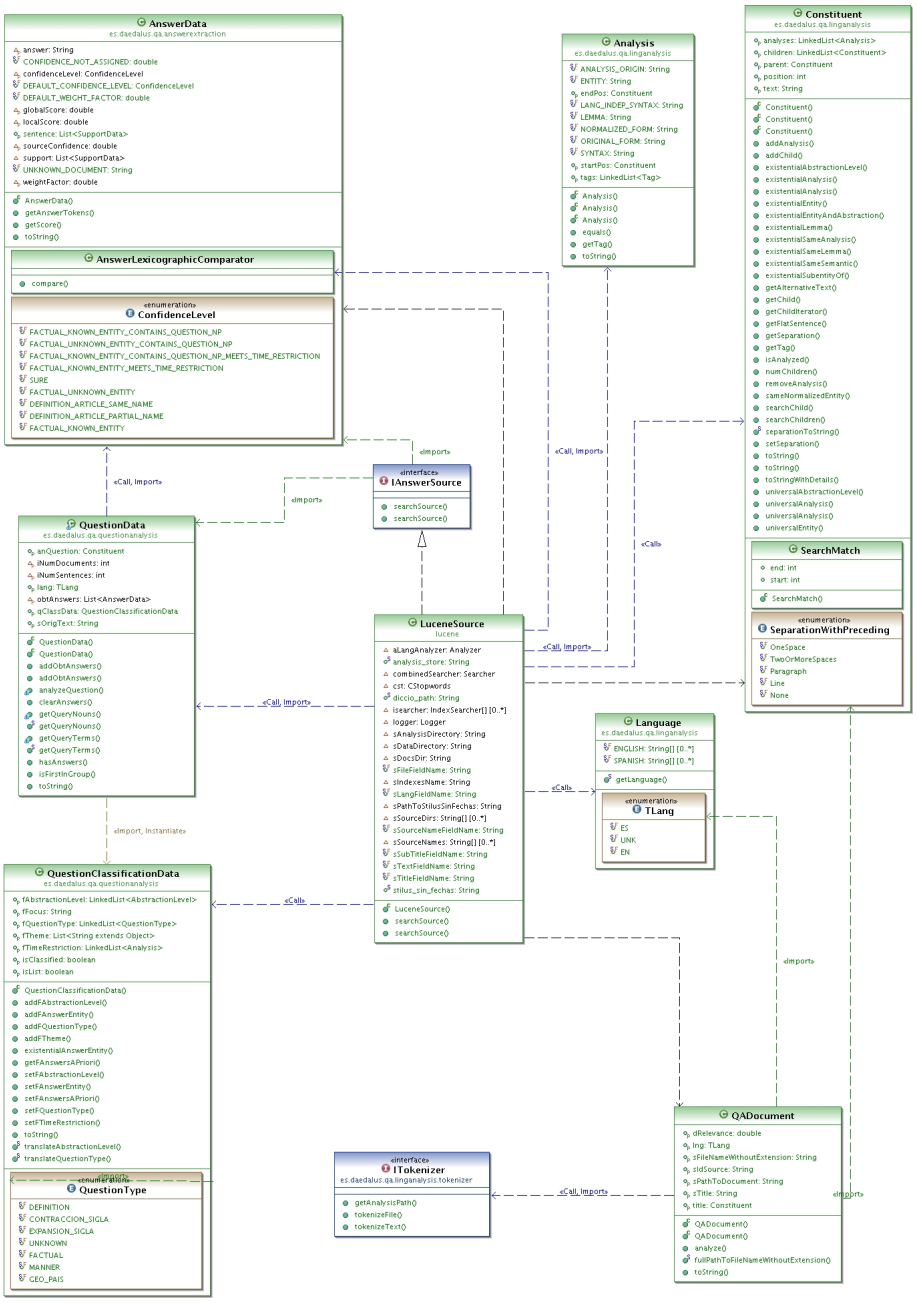

Figura A.3: Diagrama detallado *answersource*

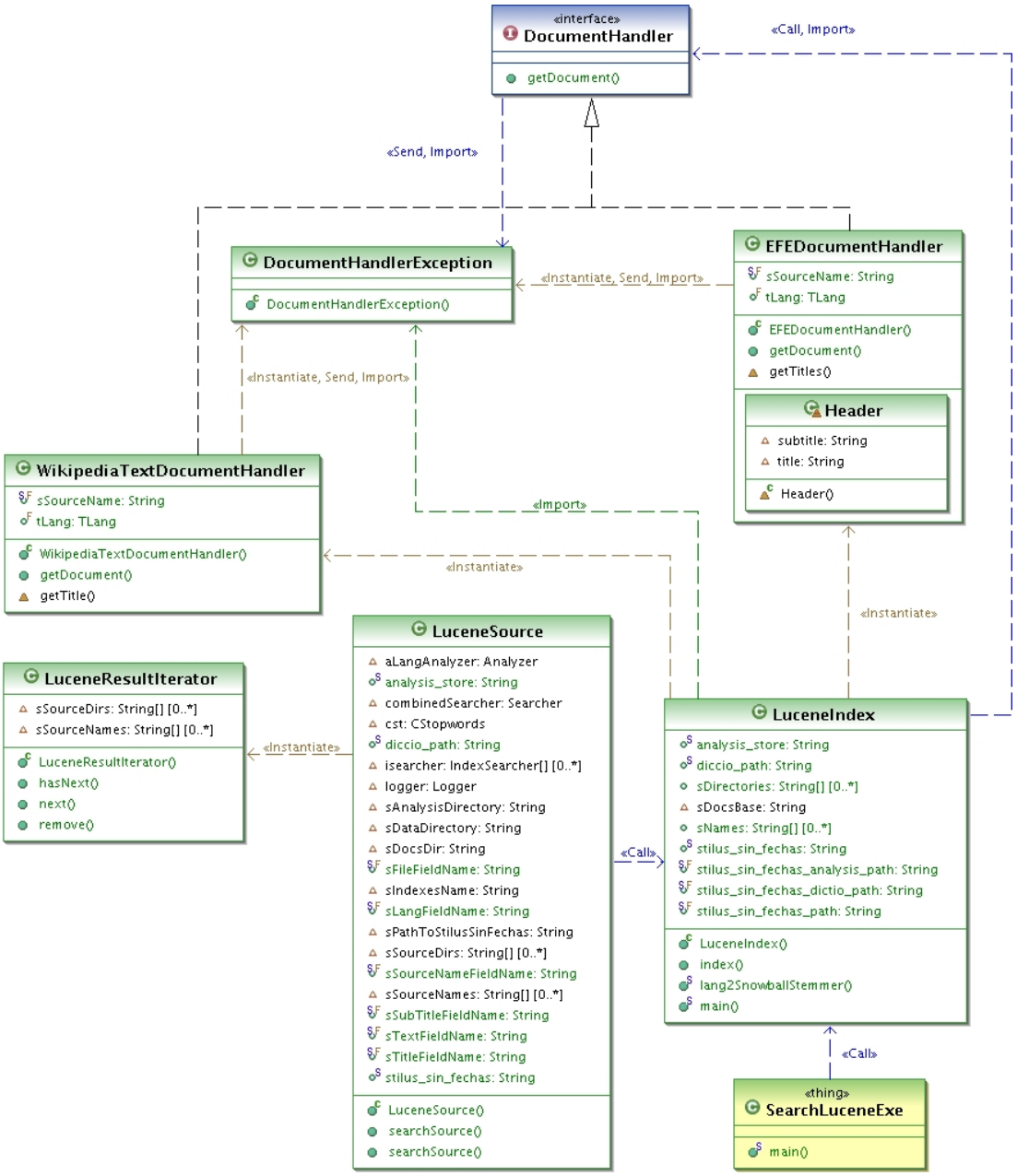

Figura A.4: Diagrama detallado *lucene*

A.3. Análisis Ligüístico

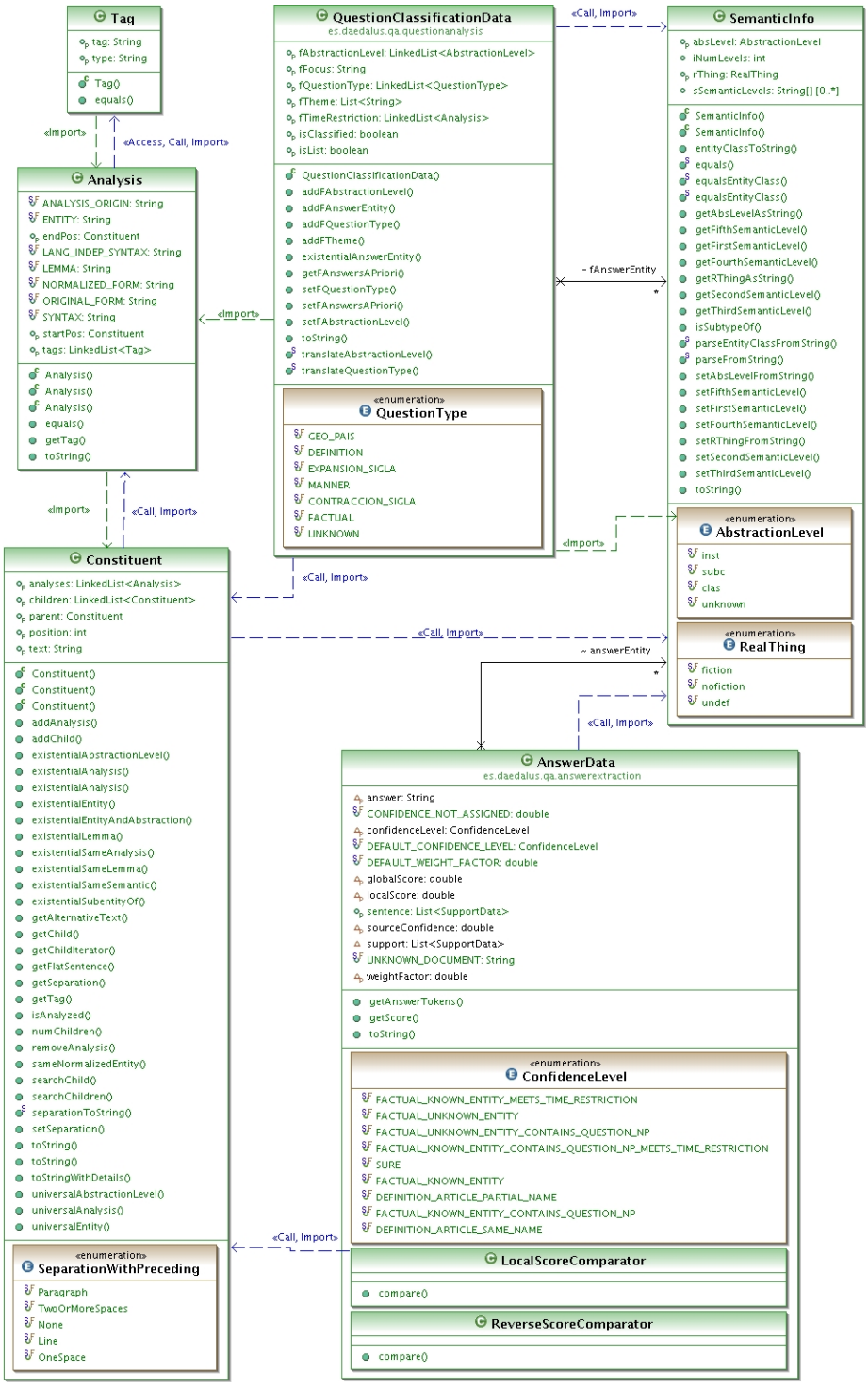

Figura A.5: Diagrama detallado *linganalysis*

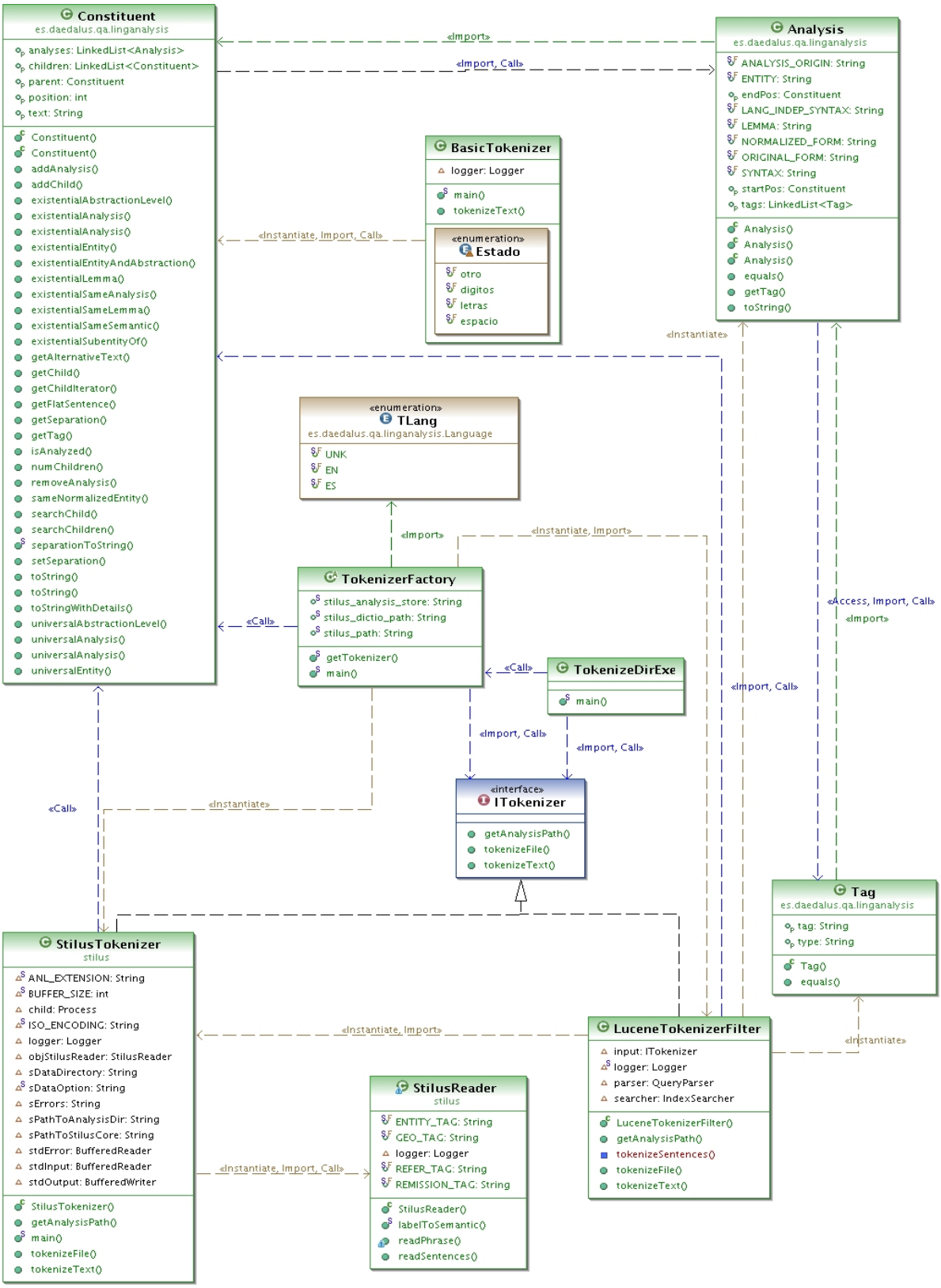

Figura A.6: Diagrama detallado *tokenizer*

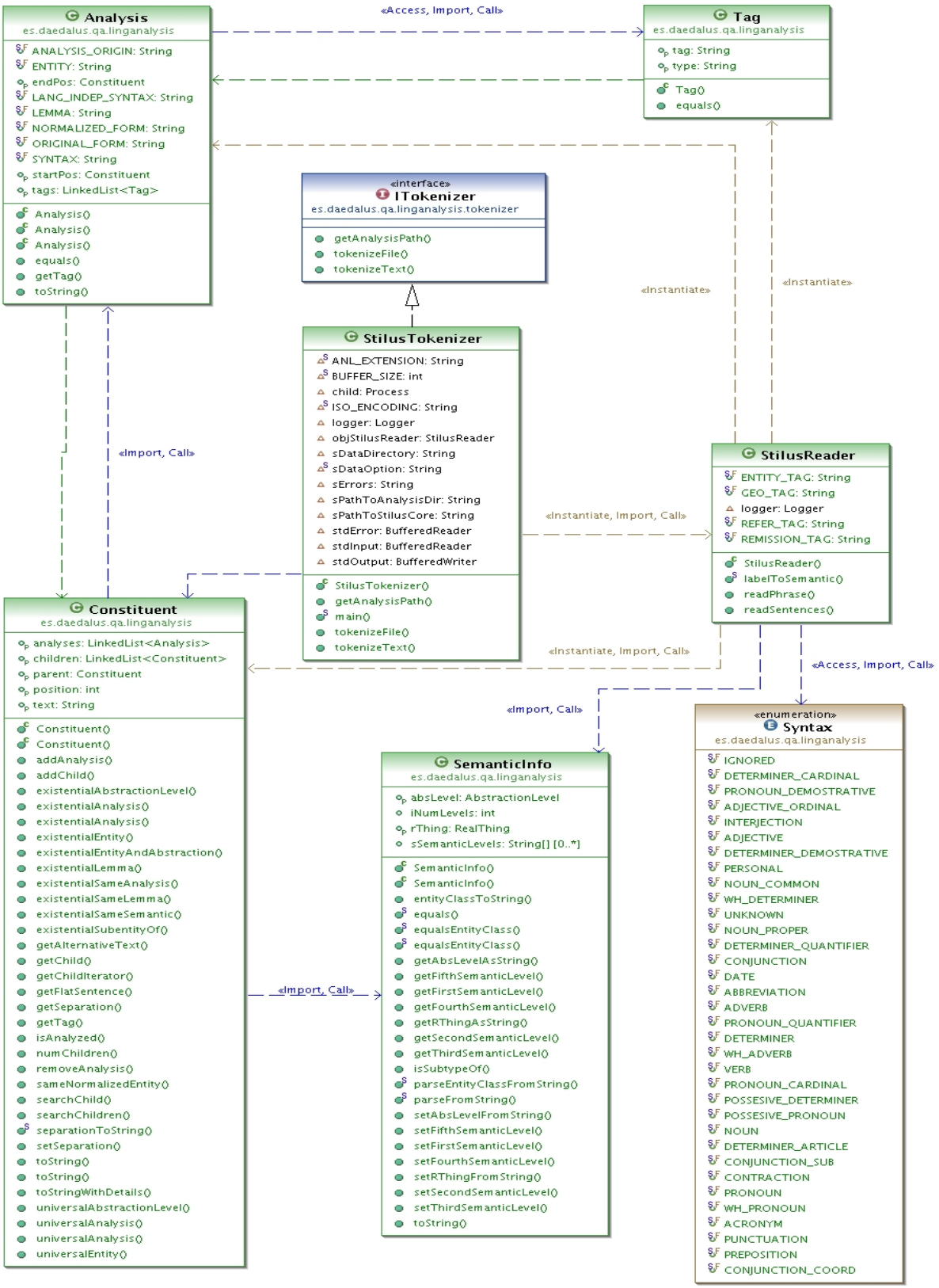

Figura A.7: Diagrama detallado *stilus*

A.4. Reglas

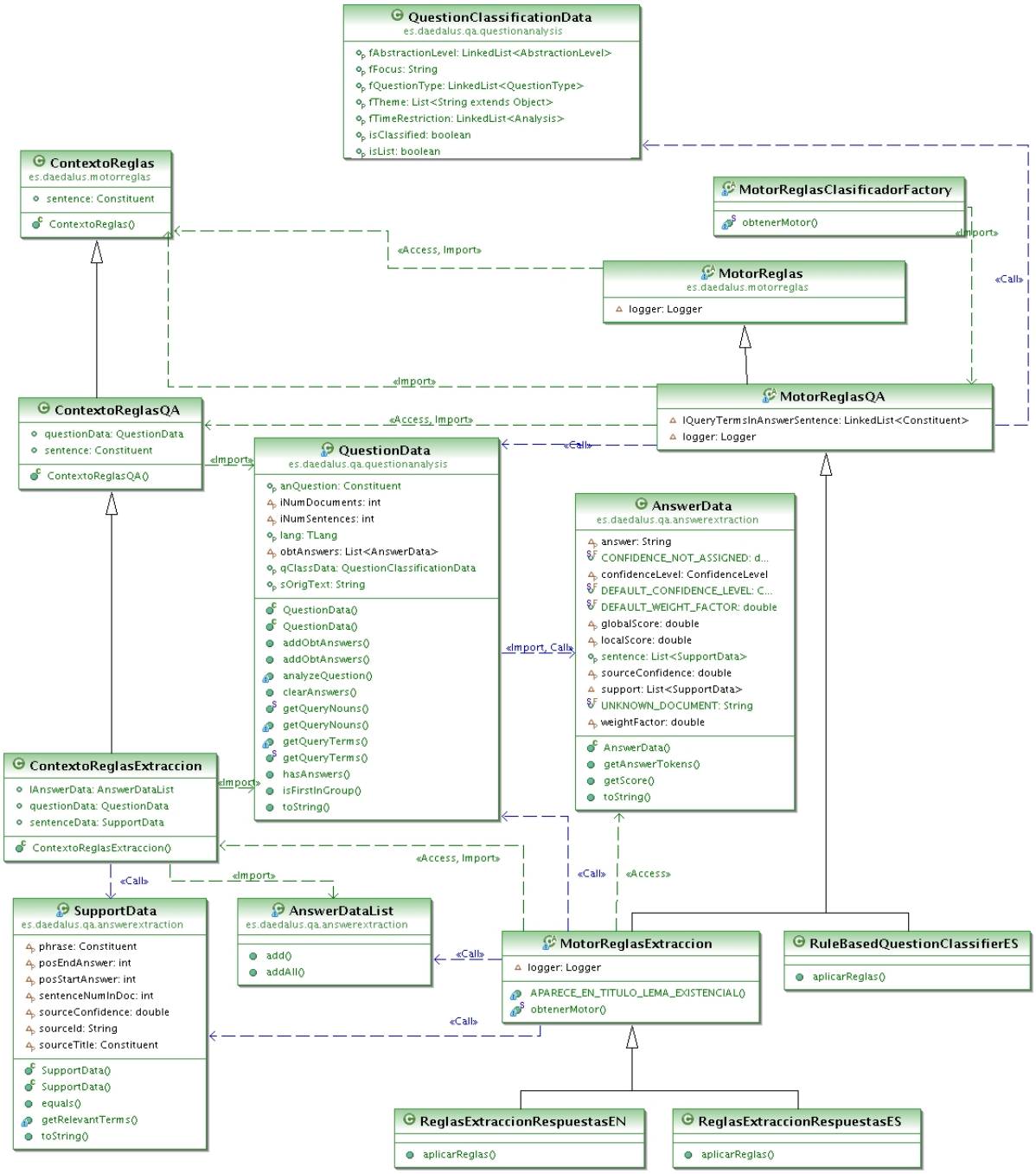

Figura A.8: Diagrama detallado *motorreglas*

A.5. Extracción de la Respuesta

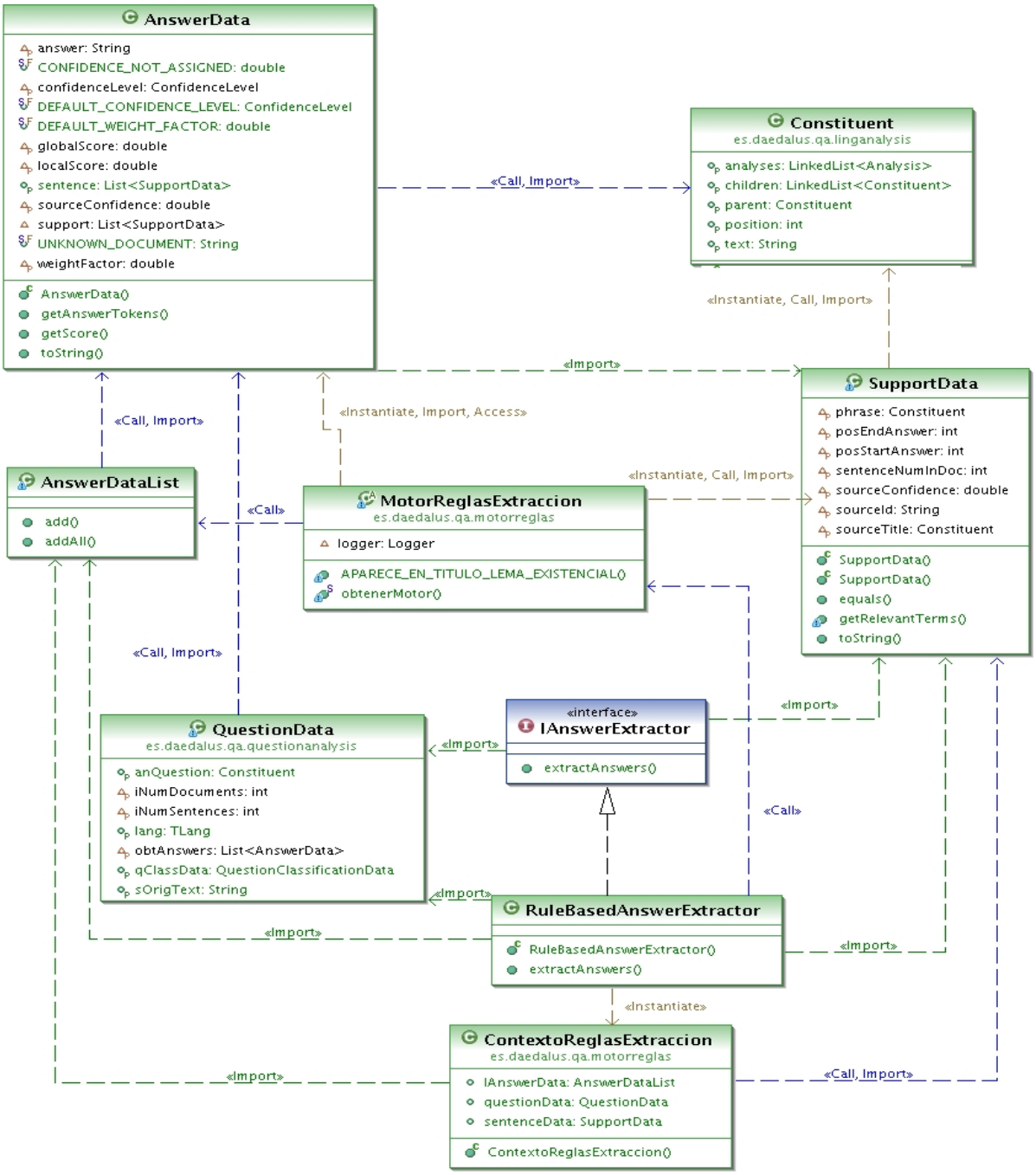

Figura A.9: Diagrama detallado *answerextraction*

A.6. Ordenación de Respuestas

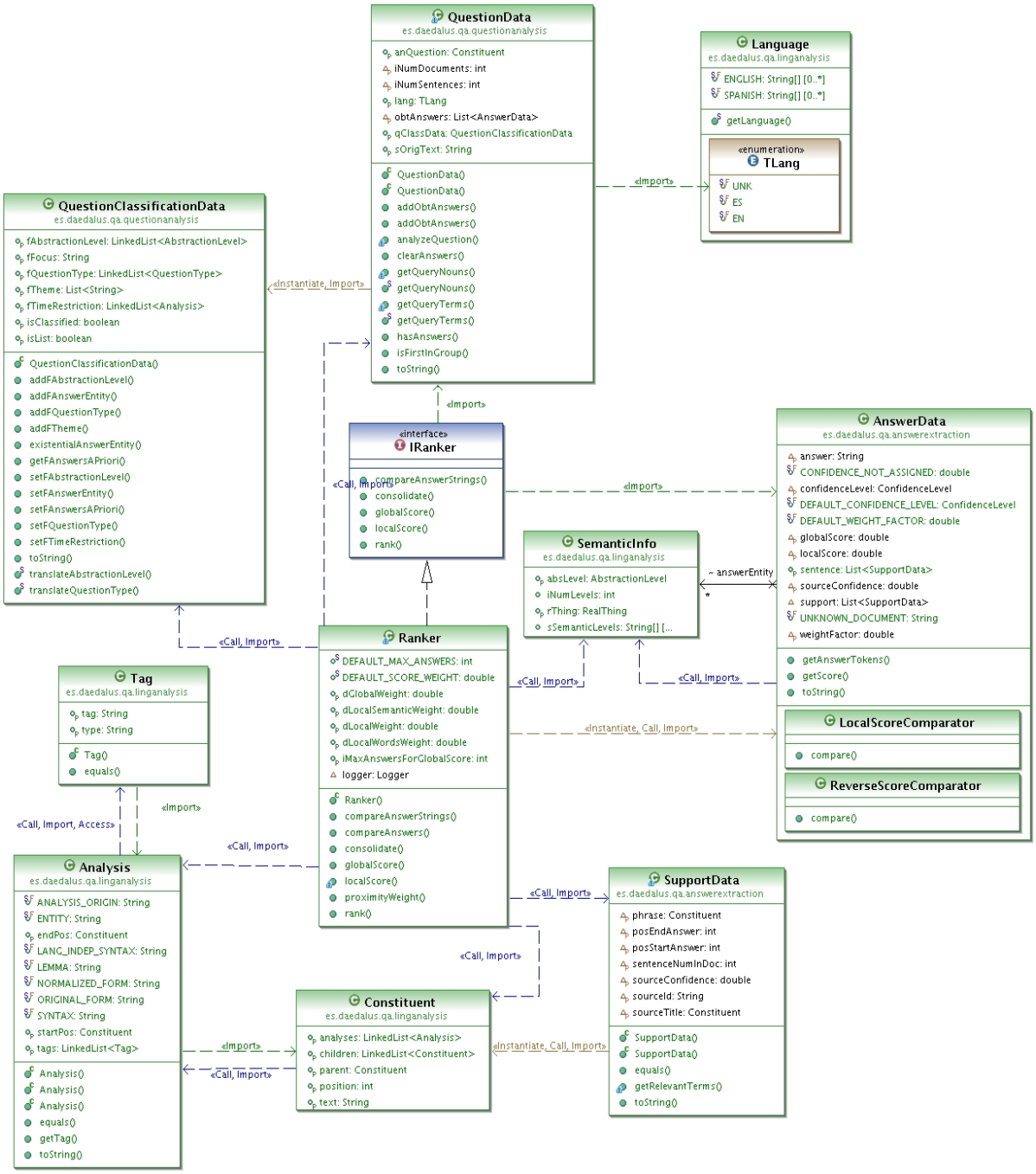

Figura A.10: Diagrama detallado *ranking*

A.7. Motor del SBR

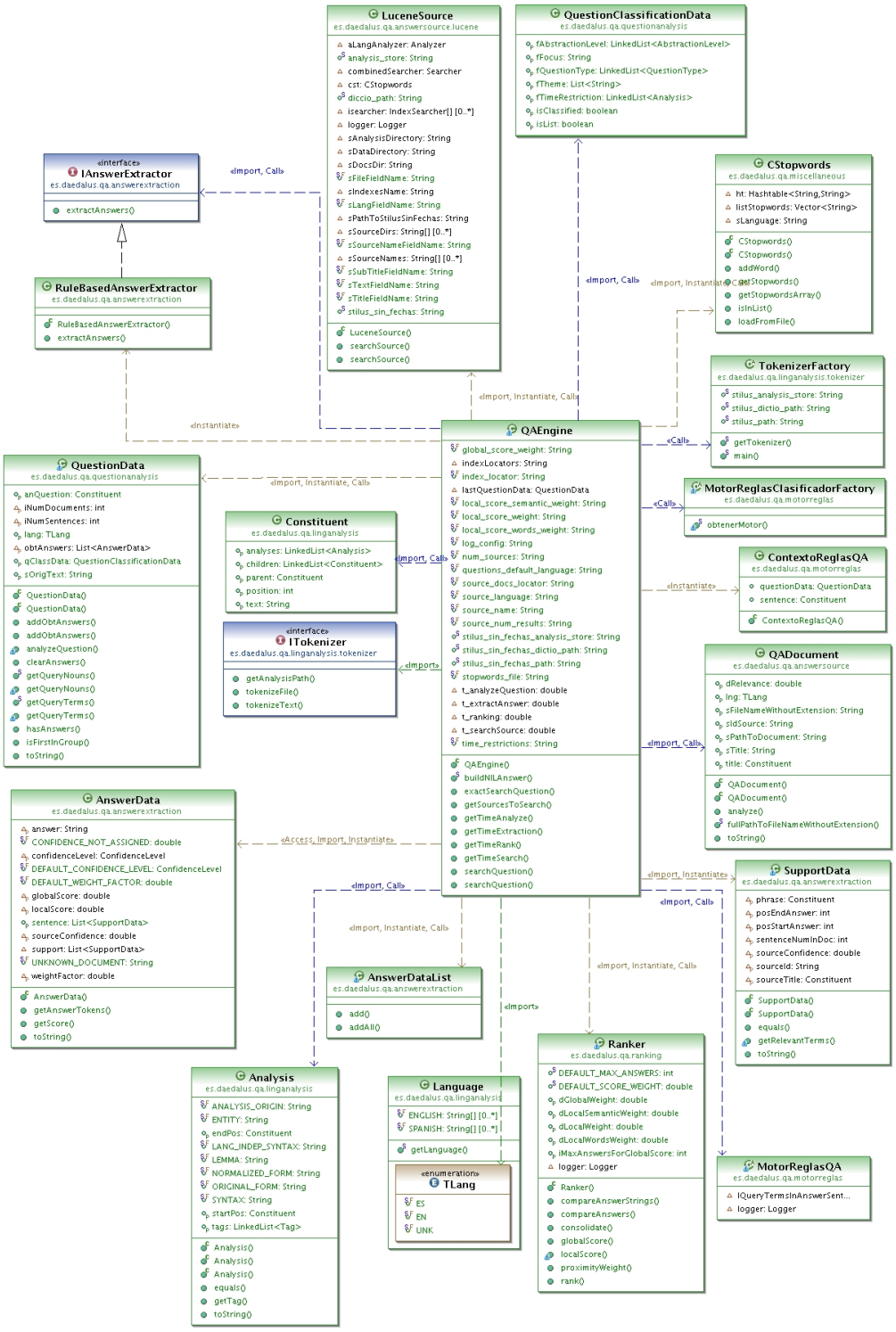

Figura A.11: Diagrama detallado *engine*

A.8. Evaluación de las Respuestas

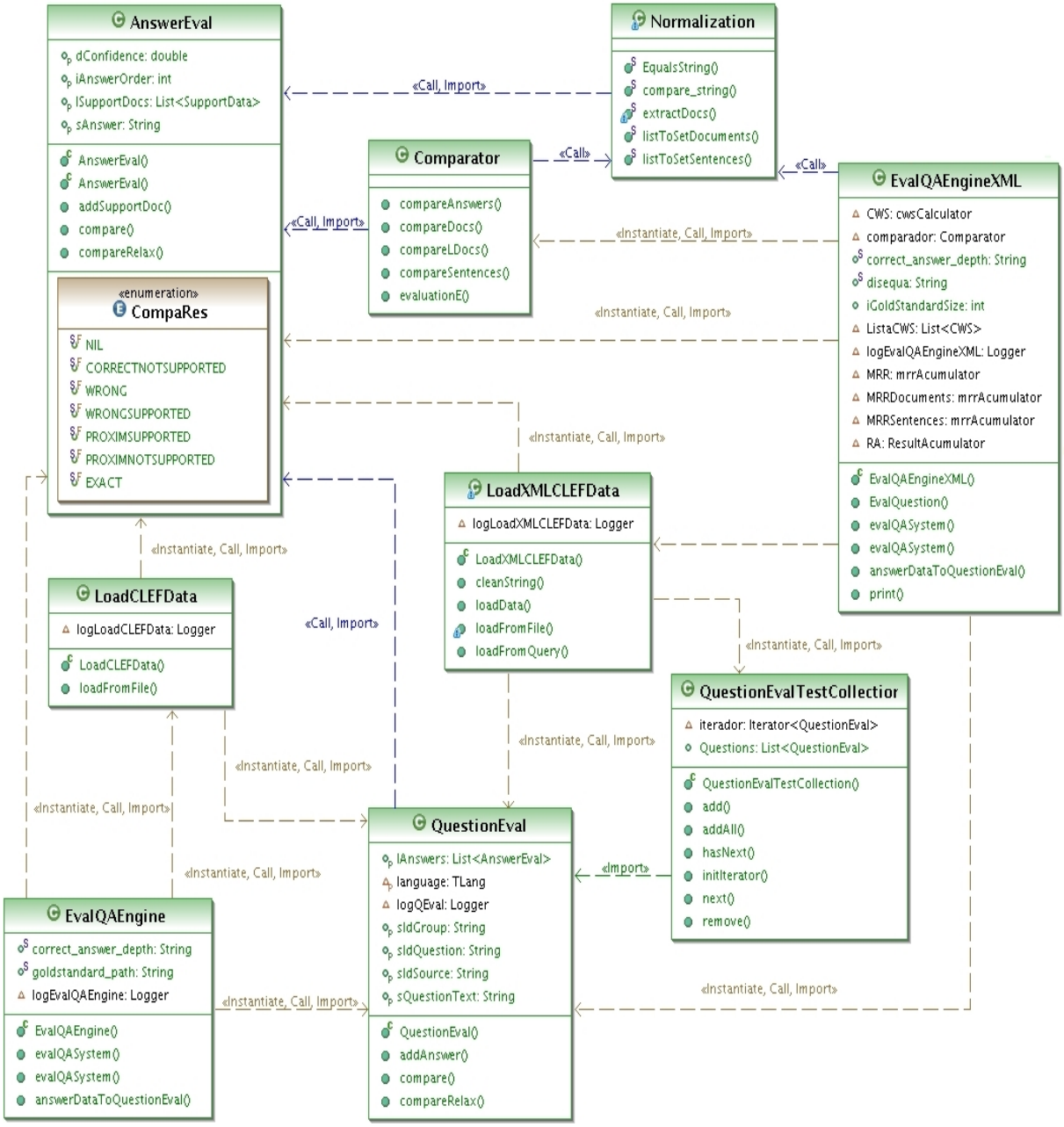

Figura A.12: Diagrama detallado *evaluation* 1/3

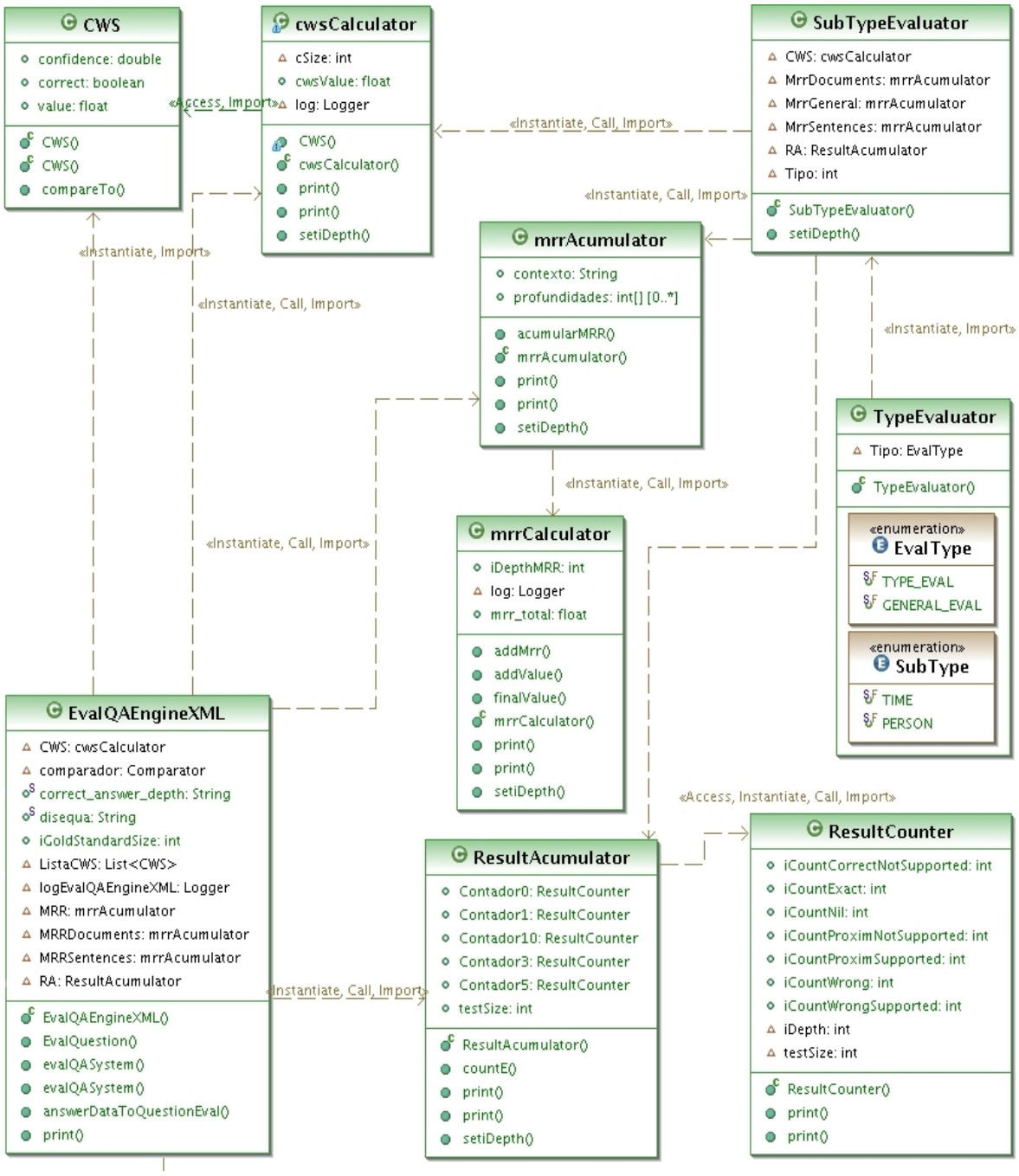

Figura A.13: Diagrama detallado *evaluation* 2/3

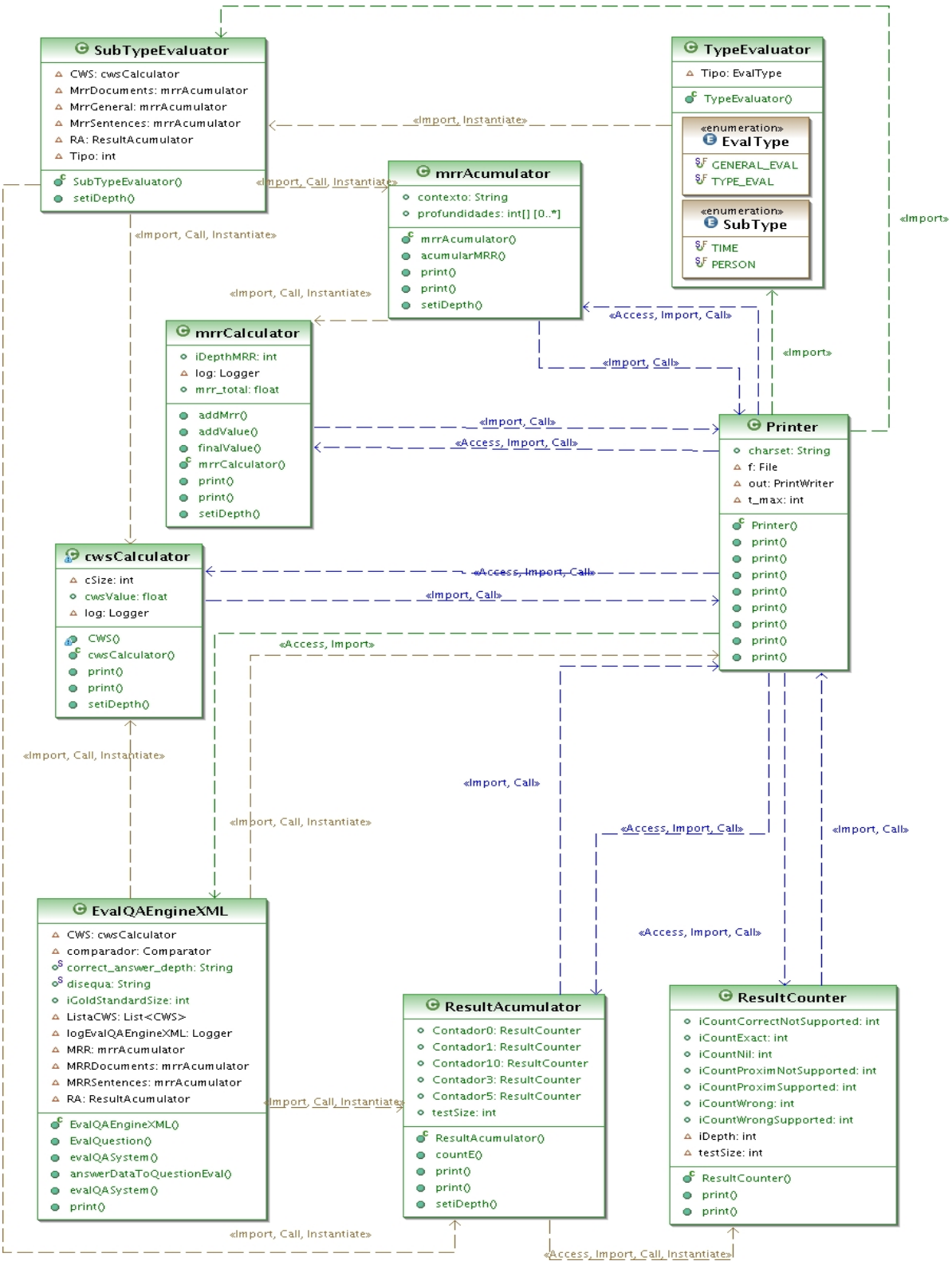

Figura A.14: Diagrama detallado *evaluation* 3/3

Apéndice B

Preguntas de Definición CLEF

B.1. QA@CLEF 2007

¿Qué es el Arca de Noé?

¿Qué es una tarantela?

¿Qué es una macana?

¿Qué es un kinnor?

¿Qué es un odómetro?

¿Qué es un tacógrafo?

¿Qué es la obsidiana?

¿Quién fue Juan Manuel Fangio?

¿Qué es un pasodoble?

¿Qué es la realidad virtual?

¿Qué es un agujero negro?

¿De qué se encarga la ONUDI?

¿Qué era la Gestapo?

¿A qué se dedica Industrial Light and Magic?

¿Cuál es la función de la Organización Meteorológica Mundial?

¿Qué fue La Mano Negra?

¿Qué es la Organización Internacional para la Estandarización?

¿Qué era el Rainbow Warrior?

¿Qué es INTASAT?

¿Qué es una Organización No Gubernamental?

¿Qué es INTELSAT?

¿Para qué se usa un dragaminas?

¿Quién fue Marco Pantani?

¿Qué fue la Revolución de los Claveles?

¿Quién fue António de Oliveira Salazar?

¿Cómo son los peces espada?

¿Quién fue Le Corbusier?

¿Quién era Flaubert?

¿Qué fue la Revolución de Terciopelo?

¿Quién era Bertha von Suttner?

¿Quién fue Hermann Emil Fischer?

¿Quién es Philip J. Fry?

B.2. QA@CLEF 2008

¿Quién era Edgar P. Jacobs?

¿Quién era Eleanor Roosevelt?

¿Quién fue Saul Bass?

¿Qué es un incensario?

¿Qué era ODESSA?

¿Qué es el oricalco?

¿Qué era la Max Azul?

¿Qué es la Carta de Londres?

¿Qué es la chatarra?

¿Qué es un adobe?

¿Qué es la vexilología?

¿Qué es el arrabio?

¿Qué es un deluminador?

¿Quién es Rick Deckard?

¿Qué son los replicantes?

¿Qué es el Kuomintang?

¿Qué es Opel?

¿Qué son los pellets?

¿Quién fue Chiang Kai Chek?

¿Quién fue Rafael Azcona?

¿Qué es Polaroid?