Universidad Carlos III de Madrid
Escuela Politécnica Superior

Diseño de un sistema de extracción de información de artículos de Wikipedia

Proyecto Fin de Carrera
Ingeniería de Telecomunicación

Autor: Miguel Sáez Guerrero
Tutor: Julio Villena Román
Madrid, Octubre de 2009

Título: Diseño de un sistema de extracción de información de artículos de Wikipedia

Autor: Miguel Sáez Guerrero

Tutor: Julio Villena Román

EL TRIBUNAL

Presidente:
Jesús Arias Fisteus

Secretario:
Norberto Fernández García

Vocal:
Francisco Javier Calle Gómez

Realizado el acto de defensa y lectura del proyecto Fin de Carrera el 5 de octubre de 2009 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de:

Fdo. Presidente Fdo. Secretario Fdo. Vocal

Resumen

El objetivo del presente proyecto es el diseño de un sistema de extracción automática de información a partir de grandes corpus de textos. Más concretamente, el desarrollo del proyecto se ha centrado en la búsqueda de información específica dentro de artículos de personajes contenidos en Wikipedia.

El sistema diseñado tratará de establecer todas las relaciones posibles entre el artículo analizado y una serie de conceptos contenidos dentro del mismo (enlaces a otros artículos). Estas relaciones serán automáticamente clasificadas dentro de la categoría que se estime más adecuada (relación laboral, invención, lugar de residencia, etc.).

La implementación del sistema combina el uso de distintas técnicas de Procesamiento de Lenguaje Natural (incluyendo herramientas de análisis morfológico, sintáctico y semántico), la potencia de PHP para el procesamiento de textos de gran tamaño y la flexibilidad de las expresiones regulares tipo Perl.

Abstract

The objective of this project is the design of an Information Extraction (IE) system to gather specific information from large text files. Specifically, the design has focused on information search within Wikipedia articles about people.

The designed system will try to establish all the possible relationships between the analyzed article and a series of concepts appearing in it (links to other articles). These relationships will be automatically classified in the most suitable category (laboral relationship, invention, place of residence, etc.).

The implementation of the system combines the use of different techniques of Natural Language Processing (such as part-of-speech, syntactic and semantic analysis tools), the power of PHP to process large text files and the flexibility of Perl Compatible Regular Expressions.

Diseño de un sistema de extracción de información de artículos de Wikipedia

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. INTRODUCCIÓN

1.1. MOTIVACIÓN

En la actualidad es cada vez más sencillo localizar grandes cantidades de información prácticamente sobre cualquier tema o aspecto sobre el que se realice una búsqueda. La gran cantidad de posibilidades que ofrece Internet, entre ellas las herramientas colaborativas que permiten a cualquier persona compartir información (blogs, wikis, etc.), están provocando un crecimiento exponencial de información no estructurada.

Una de estas fuentes de información, un proyecto libre en crecimiento constante que está teniendo un gran éxito en los últimos años es Wikipedia (Wikimedia Foundation, 2009). La posibilidad para cualquier usuario de añadir y editar artículos sobre cualquier temática la ha convertido en una de las obras de consulta más populares del mundo. Sin embargo, a pesar de su inmensa utilidad, su contenido está fundamentalmente basado en información no estructurada[1] (artículos de gran extensión) que dificulta en muchas ocasiones localizar aspectos específicos de la misma que puedan resultar de interés.

La problemática comentada puede ilustrarse fácilmente con dos ejemplos prácticos: si fuese necesario localizar los inventos de Nikola Tesla[2] o los lugares visitados por Hernán Cortés[3], la única opción posible sería revisar los artículos completos, o bien realizar búsquedas manuales, corriendo el riesgo de pasar por alto información relevante. La Wikipedia no ofrece un mecanismo automático para realizar este tipo de consultas, lo que conlleva un gasto extraordinario de tiempo.

[1] Wikipedia sí dispone de ciertas capacidades de información estructurada, las llamadas *infoboxes* (cajas de información) en determinados artículos. Sin embargo, no están normalizados ni son obligatorios, lo que hace su utilización más difícil.
[2] http://es.wikipedia.org/wiki/Nikola_Tesla
[3] http://es.wikipedia.org/wiki/Hernan_Cortes

Dentro de cada artículo de Wikipedia, los conceptos de interés contenidos en los mismos suelen aparecer en forma de enlaces a otros artículos. No existe una manera automática de determinar qué relación existe entre el artículo principal y estos enlaces mientras no se lea el texto del artículo. Por este motivo, resultaría de gran utilidad contar con un sistema que clasificase y etiquetase automáticamente estas relaciones en diferentes categorías (por ejemplo, "relación laboral", "amistad", "lugar de residencia", etc.).

Los avances conseguidos en los últimos años en herramientas de Procesamiento de Lenguaje Natural (PLN) han hecho posible que las tareas de Extracción de Información (EI) tengan cada vez más aplicaciones prácticas. En distintas conferencias y programas como MUC (*Message Understanding Conference*) (Voorhees, 2001) o ACE (*Automatic Content Extraction*) (Doddington et al., 2004) se ha demostrado su utilidad para distintos casos de aplicación. En el presente proyecto se tratarán de aprovechar diferentes técnicas de extracción y PLN para tratar de implementar un sistema que dé solución a la problemática anteriormente expuesta.

1.2. OBJETIVOS

El objetivo de este proyecto será construir un sistema que sea capaz de extraer todas las relaciones existentes entre entradas de la Wikipedia, en concreto, entre un artículo dado frente a otros artículos, mediante los enlaces aparecidos dentro del mismo, y etiquetar semánticamente dichas relaciones dentro de distintas categorías. El diseño del sistema se ha centrado específicamente en artículos de personajes, ya que resultaría bastante costoso tratar de implementar un sistema de EI similar que funcionase para cualquier clase de artículo, pero las bases aquí presentadas son generalizables a otros conceptos.

El sistema tratará de englobar los conceptos de interés contenidos en el texto (enlaces) dentro de diferentes tipos de relaciones que aparezcan frecuentemente en artículos de personajes, como puedan ser amistades, relaciones laborales, personas conocidas, profesión, etc. Las relaciones que no puedan ser englobadas dentro de un

conjunto de tipos predefinidos serán etiquetadas dentro de una categoría genérica. No obstante, en estos casos se tratará de averiguar la relación existente analizando el fragmento de texto en las que aparezcan.

Para lograr implementar el sistema descrito, se necesitará el apoyo de herramientas de PLN. Una vez completado el diseño, se realizará una evaluación del sistema utilizando un corpus seleccionado de artículos de personajes en Wikipedia pertenecientes a distintas categorías.

1.3. ESTRUCTURA DEL DOCUMENTO

La presente memoria se ha dividido en 6 capítulos, cuyo contenido se detalla a continuación.

El Capítulo 1 (capítulo actual) se corresponde con la introducción del proyecto. Se explica la motivación que ha llevado a la implementación del sistema y los objetivos que se tratarán de conseguir con el diseño.

En el Capítulo 2 se analiza el concepto de EI, describiendo distintas técnicas y mecanismos utilizados en la actualidad para llevar a cabo la misma. Se explican además los diferentes tipos de medidas utilizadas para evaluar estos sistemas, haciendo también referencia a distintos programas y conferencias actuales sobre EI.

El tercer capítulo describe el diseño del sistema de EI implementado. Se presenta el diagrama de bloques general y se explican los diferentes módulos que componen el sistema analizando el funcionamiento de cada uno de ellos.

En el Capítulo 4 se detalla la implementación del sistema. Se analizan los beneficios del uso de PHP y expresiones regulares para el desarrollo de la aplicación y se explica el funcionamiento de cada una de las clases y funciones implementadas.

En el Capítulo 5 se lleva a cabo una evaluación del sistema desarrollado. A partir de la salida del sistema para un conjunto de artículos, se calculan una serie de medidas para evaluar cada tipo de relación buscado.

En el Capítulo 6 se presentan las conclusiones obtenidas del desarrollo y evaluación del sistema, así como posibles desarrollos futuros para mejorar el funcionamiento del mismo.

Por último, en los anexos A y B se incluye el listado de artículos empleados en la evaluación así como el detalle de los resultados obtenidos. A continuación se detalla la bibliografía empleada.

2. ESTADO DEL ARTE

2.1. INTRODUCCIÓN

2.1.1 ¿Qué es la extracción de información?

Las tecnologías de Extracción de Información (EI, en inglés *Information Extraction*), son una disciplina que forma parte del Procesamiento de Lenguaje Natural (PLN, en inglés *Natural Language Processing*). Estas tecnologías han supuesto una revolución tecnológica en los ámbitos relacionados con la recuperación de información y su finalidad es facilitar la obtención de información útil por parte del usuario (Pazienza, 1997).

La manera tradicional de recuperar información a partir de documentos siempre ha sido el análisis y la extracción manual de la misma. Mediante la EI lo que se pretende conseguir es tomar estos documentos como entrada y producir una salida estructurada y de formato fijo. Los datos obtenidos pueden mostrarse directamente al usuario, o bien almacenarse en una base de datos para un posterior análisis. Otra posibilidad es utilizarlos para indexación en aplicaciones de Recuperación de Información (RI, en inglés *Information Retrieval*).

Resulta interesante hacer una comparación entre la EI y la RI. Mientras que la RI simplemente se basa en encontrar textos y presentarlos al usuario, las aplicaciones típicas de EI analizan textos y presentan sólo la información específica en la que se está interesado. Por ejemplo, un usuario de un sistema de RI llevaría a cabo una búsqueda introduciendo palabras relevantes y recibiría un conjunto de documentos apropiados. Por otro lado, un usuario de un sistema de EI podría, con una aplicación debidamente configurada, rellenar automáticamente una tabla con los resultados obtenidos.

Existen ventajas y desventajas de los sistemas de EI con respecto a los de RI. Los primeros son más difíciles de construir, y para la mayoría de las tareas son menos precisos que los lectores humanos. Además la EI es más compleja computacionalmente que la RI. Sin embargo, en aplicaciones donde hay grandes volúmenes de texto, la EI es

potencialmente mucho más eficiente que la RI por la posibilidad de reducir la cantidad de tiempo que los analistas invierten leyendo textos.

2.1.2 Datos no estructurados

La EI se basa normalmente en obtener alguna información a partir de datos no estructurados. El texto hablado y escrito, las imágenes, el vídeo y el audio son posibles formas de datos no estructurados. El término "no estructurados" no implica que los datos sean estructuralmente incoherentes (en cuyo caso no tendría sentido alguno), sino que su información está codificada de tal forma que hace difícil a los ordenadores interpretarla inmediatamente. La EI es el proceso que añade significado a datos no estructurados y sin tratar. En consecuencia, los datos se transforman en estructurados o semi-estructurados y se pueden procesar más fácilmente usando un ordenador.

2.1.3 Extracción de Información Semántica

La EI identifica información en textos aprovechando su organización lingüística. Cualquier texto consiste en una compleja estructura de capas de patrones recurrentes que forman una unidad coherente. Esto es consecuencia del *principio de composicionalidad* (Szabó, 2004), una idea general de la filosofía lingüística que sustenta muchos planteamientos modernos del lenguaje. Este principio afirma que el significado de cualquier expresión lingüística compleja es una función de los significados de las partes que lo constituyen. Una frase típicamente contiene un determinado número de partes (por ejemplo, un sujeto, un verbo, un objeto). Sus significados individuales, su orden y comprensión permiten determinar lo que la frase significa. Si un texto fuese totalmente irregular, sería sencillamente imposible para cualquier persona extraer su significado.

Los mecanismos de EI presuponen que aunque la información semántica en un texto no es computacionalmente transparente de manera directa, puede recuperarse si se tienen en cuenta las regularidades de su organización interna. Un sistema de EI utilizará un conjunto de patrones de extracción, que pueden construirse manualmente o

aprenderse automáticamente. Gracias a estos patrones la información puede extraerse de un texto y organizarse en un formato más estructurado.

El uso del término *extracción* implica que la información semántica está explícitamente presente en la organización lingüística de un texto. Esto significa que la información está fácilmente disponible en los elementos léxicos, las construcciones gramaticales y el orden pragmático del texto fuente. En este sentido, la extracción de información es diferente de las técnicas que deducen información de los textos a partir de reglas lógicas (inferencia lógica).

2.1.4 Extracción de Información Específica

La EI se aplica tradicionalmente en situaciones en las que se conoce de antemano la clase de información semántica que debe ser extraída. Por ejemplo, puede ser necesario identificar qué clase de eventos aparecen en un texto concreto y en qué momento estos eventos tienen lugar. Como los eventos y expresiones temporales pueden expresarse solamente de un limitado número de formas, es posible diseñar un método para identificar eventos específicos y su localización temporal correspondiente. Dependiendo de la necesidad de información, pueden construirse diferentes modelos para distinguir diferentes categorías semánticas. En algunas aplicaciones, por ejemplo, será suficiente con indicar que una parte de una frase es una expresión temporal, mientras que en otras podría ser necesario distinguir entre expresiones que indiquen pasado, presente y futuro.

La EI no presenta al usuario documentos completos, sino que extrae unidades textuales o elementos de los documentos, típicamente oraciones simples (Appelt e Israel, 1999), también llamadas *regiones de texto*. Por tanto, la EI es diferente de la generación de resúmenes basada en extractos, que normalmente recupera oraciones completas de textos para utilizarlas como resumen.

La especificidad implica que no sólo la naturaleza semántica de la información esté predefinida en un sistema de EI, sino también la *unidad* y *ámbito* de los elementos a ser extraídos. Algunas unidades de extracción típicas son palabras compuestas y

grupos nominales, pero en algunas aplicaciones podría ser oportuno extraer otras unidades lingüísticas, como grupos verbales, indicadores temporales, oraciones, etc. Mientras que la unidad de extracción tiene que ver con la granularidad de trozos individuales de información, el ámbito se refiere a la granularidad del espacio de extracción para cada solicitud individual de información. La información puede extraerse de una o múltiples oraciones examinando uno o más textos antes de ser devuelta por el sistema.

2.1.5 Clasificación y estructuración

Algo típico en la EI es que la información no sólo se extraiga de un texto sino que después también se clasifique semánticamente para asegurar su uso futuro en sistemas de información. Haciendo esto, la información no estructurada de textos fuente se transforma también en estructurada.

Cualquier proceso de clasificación requiere un esquema de clasificación semántica, es decir, un conjunto de clases semánticas que estén organizadas de alguna manera relevante (por ejemplo en una jerarquía). Estas clases se usan para categorizar los trozos de información en un cierto número de grupos según su significado. Existe una gran cantidad de esquemas de clasificación semántica que pueden usarse, unos más abstractos y otros más específicos.

Basándose en el enfoque general de información de un sistema, podemos hacer una distinción fundamental entre sistemas de EI de *dominio cerrado* y *dominio abierto*. Tradicionalmente, los sistemas de EI eran de dominio cerrado, lo que significa que estaban diseñados para funcionar en un dominio muy especializado y bien definido y usar unas reglas de clasificación muy específicas. Por otra parte, los sistemas de dominio abierto son capaces de manejar textos pertenecientes a dominios y temas heterogéneos. Estos sistemas normalmente usan esquemas de clasificación muy genéricos, que se podrían redefinir si se requiriese una identificación más específica.

Antes se mencionó que la EI básicamente convierte información no estructurada en estructurada. Para conseguirlo debe existir una estructura predefinida, una

representación, en la que pueda colocarse la información extraída. Aunque esta información puede etiquetarse únicamente para ser procesada por el sistema de información, en el pasado se desarrollaron muchos sistemas de extracción basados en plantillas. Estas plantillas consisten en un conjunto de pares atributo-valor, cada uno de los cuales representa un aspecto relevante del evento. Una tarea de EI típicamente intenta sacar información de un texto fuente y hacerla corresponder a un valor vacío de la plantilla definida.

Para saber qué información debe ir en cada hueco de la plantilla, una aplicación de EI utiliza un conjunto de reglas de extracción. Estas reglas establecen qué propiedades lingüísticas o formales debe poseer un trozo concreto de información para pertenecer a un grupo semántico determinado. Típicamente estas reglas se definen a mano, aunque en la actualidad el aprendizaje automático juega un papel importante en el paradigma de la extracción.

2.2. MECANISMOS PARA LA EXTRACCIÓN DE INFORMACIÓN

En este apartado se realizará una breve descripción de los distintos mecanismos que se han utilizado en los últimos años para las tareas de EI. En primer lugar, se dará una explicación de los distintos patrones que pueden aparecer en cualquier tipo de texto (léxicos, semánticos, sintácticos, etc.). Posteriormente se profundizará en los conceptos de aprendizaje supervisado y no supervisado, que en los últimos años han cobrado una gran importancia.

2.2.1 Reconocimiento basado en patrones

Como se ha visto anteriormente, la EI trata de la detección y el reconocimiento de cierta información en textos no estructurados, y depende de métodos de reconocimiento de patrones. La clasificación o reconociendo de patrones pretende clasificar datos basándose en un conocimiento adquirido a priori.

El reconocimiento basado en patrones clasifica objetos en un número de clases o categorías según el patrón observado (Theodoridis y Koutroumbas, 2003). Los objetos se describen con un número de atributos escogidos y sus valores. De esta forma un objeto ***x*** puede describirse como un *vector de atributos*:

$$\boldsymbol{x} = [x_1, x_2, \ldots, x_p]^T$$

donde p es el número de atributos medidos y x_i es cada uno de los atributos.

Los atributos abarcan un espacio multi-variable llamado espacio de medida o espacio de atributos. Los vectores no son la única forma de representar objetos textuales. También pueden usarse *objetos estructurados* presentaciones en lógica de predicados y grafos. Una representación que se adapta bien a los textos es un *árbol*.

Dependiendo del caso se elige un conjunto de atributos u otro. Normalmente no se utilizan todos los atributos presentes en un texto, sino que se escoge un número importante de ellos para la tarea de extracción. En entornos de dominio abierto, es importante que los atributos sean lo suficientemente genéricos. A continuación se explicarán los tipos de atributo más frecuentemente usados en tareas de extracción. Pueden clasificarse en *léxicos*, *sintácticos*, *semánticos* y *de discurso*.

2.2.1.1 Patrones léxicos

Las características léxicas se refieren a los atributos de las palabras de un texto, sin necesidad de tener en cuenta su contexto. Puede distinguirse entre las palabras de la unidad de información que va a analizarse, y las palabras de su contexto.

En el caso de reconocimiento de entidades con nombre, las características morfológicas de la información suelen ser importantes. Estas características pueden incluir convenciones especiales de caracteres o la existencia de dígitos o letras mayúsculas en las palabras. Como es difícil representar todas las composiciones posibles en un vector de atributos, las entidades muchas veces se hacen corresponder con lo que a veces se conoce como *tipos cortos* (Collins, 2002). Un *tipo corto* de una

palabra puede definirse, por ejemplo, reemplazando cualquier secuencia de letras mayúsculas con ‘A’, de minúsculas con ‘a’ y de dígitos con ‘0’, y conservando los caracteres no alfanuméricos. Por ejemplo, la palabra TGV-3 se convertiría en A-0.

Existen ciertos atributos que pueden detectarse usando reglas heurísticas sencillas. Por ejemplo, se puede identificar el título, nombre y apellidos de una persona y usarse como atributo en resolución de correferencia.

Es común que aparezcan palabras que puedan tener diferentes variantes a la hora de escribirlas. Especialmente, nombres propios como los nombres de persona pueden aparecer en un texto usando distintos alias. Los principales problemas a los que tienen que hacer frente los patrones léxicos para la extracción y recuperación son:

- Cambios en puntuación: UE vs. U.E.
- Cambios en capitalización: Cajamadrid vs. CAJAMADRID
- Cambios en espaciado: J.S. BACH vs. J. S. BACH
- Abreviaturas y acrónimos: "extracción de información" vs. EI
- Omisión de caracteres: Colin vs. Collin
- Adición de caracteres: MacKeown vs. McKeown
- Sustituciones: Kily vs. Kyly
- Cambio de orden: Pierce vs. Peirce

La capitalización y la puntuación pueden resolverse usando una normalización sencilla. Para las abreviaturas y acrónimos se deben usar tablas de traducción, o bien definir reglas de resolución de acrónimos.

Para los problemas de sustitución de caracteres, se utilizan algoritmos de cómputo de distancia. La semejanza entre dos cadenas de caracteres se basa en el coste asociado con convertir un patrón en otro. Si las cadenas son de la misma longitud, el coste está directamente relacionado al número de símbolos que difieren. Si no, deben añadirse o eliminarse caracteres. La distancia *D(A,B)* llamada “distancia de edición” se define como:

$$D(A,B) = \min_{j}[S(j) + I(j) + R(j)]$$

donde *S* es el número de caracteres sustituidos, *I* es el número de inserciones realizadas, *R* el número de letras eliminadas y *j* representa todas las posibles combinaciones de variaciones de símbolos para obtener B desde A. Hay diferentes definiciones, como la de Levenshtein (Cáceres, 2008).

2.2.1.2 Patrones sintácticos

El atributo sintáctico más común usado en EI es la *categoría léxica* (*part-of-speech, POS*) de una palabra. Existen multitud de herramientas de reconocimiento de categoría léxica que operan con gran precisión (Manning, 2007).

La categoría léxica de una palabra juega un papel importante a la hora de determinar los valores de otros atributos. Así, puede definirse la *determinación de una unidad de información* si ésta va precedida por un artículo determinado o un demostrativo (por ejemplo, *Ví un hombre y el hombre era viejo*; *Esa persona llevaba una ropa rara*). En este ejemplo *un hombre* es información indeterminada. La determinación puede separarse en dos atributos booleanos: determinado e indeterminado (Ng y Cardie, 2002). Esto permite describir casos que no estén en ninguna de las dos situaciones.

Detectar el *tipo de grupo* (por ejemplo, un grupo nominal como *el oso grande*, un grupo nominal preposicional como *en la fría montaña*) es importante en tareas de reconocimiento de funciones semánticas. El núcleo de un grupo es en este caso un atributo muy útil. El núcleo es la palabra cuya categoría léxica representa a la de todo el grupo (por ejemplo, *hombre* en el grupo nominal: *el gran hombre*).

La *voz de una oración* (pasiva o activa) es un atributo útil en tareas de extracción. Puede detectarse basándose en las expresiones y en las características léxicas de las palabras verbales. Otro atributo puede determinar si la oración es afirmativa o negativa, pero es más difícil de detectar con exactitud.

Muchos atributos sintácticos se basan en analizar sintácticamente la estructura de la frase. Desgraciadamente, los analizadores sintácticos no están disponibles en todos los idiomas. La función gramatical de un grupo dentro de una oración (sujeto, complemento directo, complemento indirecto) puede ser de gran importancia en un proceso de extracción. Las funciones sintácticas se detectan con la ayuda de reglas aplicadas en el árbol sintáctico de una frase.

2.2.1.3 Patrones semánticos

Los atributos semánticos se refieren a clasificaciones semánticas de unidades de información de una o más palabras. Actúan como atributos en otras tareas de clasificación semántica. Un ejemplo podría ser la oración *John Smith trabaja para IBM*, donde *John Smith* e *IBM* son clasificados como nombre de persona y de empresa respectivamente. Estos atributos más generales pueden usarse en el reconocimiento de la relación *trabaja para*. Hay muchas circunstancias donde la sustitución de palabras y términos por conceptos semánticos más generales es especialmente ventajosa, especialmente para clasificar semánticamente unidades más largas de información (Moreda, 2008).

En resolución de correferencia es muy importante utilizar clases semánticas como *hombre*, *mujer*, *persona* y *organización*, o *animado* e *inanimado* y encontrar antecedentes y referentes en estas clases. Los atributos semánticos pueden implicar identificación simple del nombre de un día o mes por las clases respectivas *día* o *mes*, el reconocimiento de categorías útiles como *nombre de persona*, *nombre de empresa*, *número* y *dinero*, y el reconocimiento de clases muy generales como el *locutor* en un proceso verbal.

Una ventaja adicional es que el etiquetado semántico de palabras individuales permite reglas más generales que las basadas exclusivamente en palabras exactas. Hay muchas maneras de identificar atributos semánticos. En primer lugar, pueden detectarse usando las técnicas típicas de extracción, como reconocimiento de entidades con nombre y reconocimiento de la función semántica. También podemos utilizar

diccionarios o léxicos externos leídos por máquinas, que pueden ser de dominio general o específico. Especialmente útil puede ser un léxico semántico usado para etiquetare palabras individuales con clases semánticas apropiadas al domino. Los léxicos semánticos pueden estar incompletos, y en aplicaciones prácticas normalmente deben complementarse con recursos específicos de domino.

2.2.1.4 Patrones de discurso

Los atributos de discurso son atributos cuyos valores se calculan usando fragmentos de texto, es decir, un discurso o escrito conectado más extenso que una frase. Muchos atributos de discurso son interesantes en un contexto de EI.

Un ejemplo muy simple es la *distancia de discurso*. En el reconocimiento de relaciones, la distancia entre dos entidades suele ser importante ya que se asume que es inversamente proporcional a la relación semántica. La distancia de discurso puede expresarse como el número de palabras o de frases que intervienen.

Los atributos de discurso tales como las *relaciones retóricas*, *temporales* y *espaciales* son importantes en la clasificación semántica de unidades de gran tamaño. El reconocimiento de expresiones temporales y su orden relativo se considera también una tarea de EI. En la actualidad, los experimentos relacionados con la clasificación automática de relaciones temporales son muy limitados (Mani et al., 2003).

2.2.2 Aprendizaje supervisado

El aprendizaje supervisado (Figura 1) es un planteamiento muy popular en cualquier tarea de extracción. Existen muchos algoritmos diferentes que aprenden patrones de clasificación a partir de un conjunto de ejemplos clasificados (Mitchell, 1997). Dada una muestra de ejemplos, la tarea consiste en modelar el proceso de clasificación y usar el modelo para predecir las clases de ejemplos nuevos no vistos anteriormente. En RI las técnicas supervisadas son muy populares para clasificar documentos en categorías, usando las palabras del documento como atributos

principales. En EI normalmente se clasifican unidades más pequeñas con una gran variedad de esquemas de clasificación, desde categorías genéricas hasta clases muy específicas.

La cantidad de ejemplos de entrenamiento está normalmente limitada, ya que estos ejemplos son difíciles de construir. Cuando el conjunto de ejemplos es pequeño, normalmente representa un conocimiento incompleto del modelo buscado. Este peligro está especialmente presente en datos de lenguaje natural, donde una gran variedad de modelos expresa el mismo contenido.

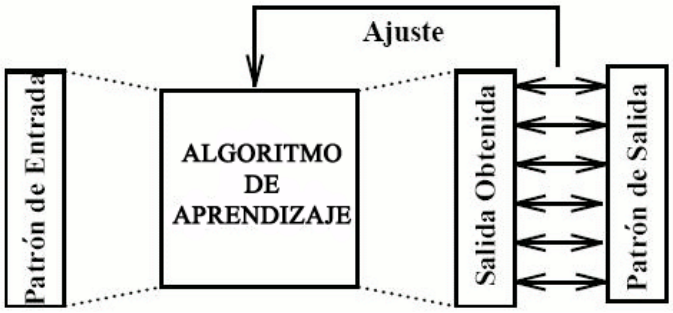


Figura 1. Aprendizaje supervisado

A la hora de implementar un sistema para extraer información de manera supervisada pueden usarse distintos métodos que se describirán a continuación.

2.2.2.1 Árboles de decisión

Son modelos de predicción usados en inteligencia artificial. Partiendo de una base de datos se elaboran diagramas de construcciones lógicas. Estos diagramas son similares a los sistemas de predicción basados en reglas (Mitchell, 1977), que categorizan una serie de condiciones para resolver un problema.

Las entradas del árbol pueden ser objetos o situaciones descritas a través de atributos (McCarthy y Lehnert, 1995). A partir de ellas se genera una respuesta o

decisión. Las entradas y salidas pueden tomar valores continuos o discretos, aunque estos últimos se usan más por simplicidad.

2.2.2.2 Redes neuronales

Este método de aprendizaje se desarrolló en los ámbitos del análisis estadístico y la IA. Se basa en extraer combinaciones lineales de los atributos obteniendo una serie de características, y modelar las diferentes clases como funciones de dichas características (Bishop, 1995). Concretamente, los modelos lineales de redes neuronales se han utilizado masivamente en aplicaciones estadísticas (Cherkassky, 1996).

Los modelos lineales son especialmente útiles cuando los atributos son numéricos, ya que de esta manera cada clase puede representarse como una combinación lineal de atributos. Sin embargo, tienen ciertos problemas, como la naturaleza lineal de las relaciones, y que el número de patrones debe ser mucho mayor al de atributos.

2.2.2.3 Modelos ocultos de Markov

Son modelos estadísticos en los que se asume que el sistema a modelar es un proceso de Markov de parámetros desconocidos. Se pretende determinar estos parámetros ocultos usando los parámetros observables. Se conoce como *proceso de Markov* un proceso estocástico discreto en el que el pasado es irrelevante para conocer el futuro dado el presente (Rabiner, 1989).

Los tres problemas fundamentales a resolver en el diseño de un modelo oculto de Markov son: la evaluación de la probabilidad de una secuencia de observaciones dado un modelo específico; la determinación de la mejor secuencia de estados del modelo; y el ajuste de los parámetros del modelo que mejor se ajusten a los valores observados.

2.2.2.4 Algoritmos del vecino más cercano (K-Nearest Neighbours)

Son métodos de aprendizaje basados en ejemplos. Se utiliza una función de distancia para determinar qué patrón del conjunto de entrenamiento está más cerca del patrón a clasificar (Dasarathy, 1991).

2.2.3 Aprendizaje no supervisado

En EI se han empleado muchos algoritmos supervisados que se entrenan usando ejemplos etiquetados, obteniendo gran éxito en sus resultados. Sin embargo, muchos estudios indican que este funcionamiento es sólo adecuado en dominios limitados y cerrados. Cuando se trabaja en dominios abiertos, un cuello de botella importante es la falta de ejemplos etiquetados suficientes. (Duda et al., 2001).

Las tecnologías de aprendizaje no supervisado se basan en explotar los datos no etiquetados, por lo que no necesitan supervisión externa. Son capaces de modificar sus parámetros internamente.

El fundamento en el que se basan es aprovechar la redundancia que existe en el lenguaje natural. De esta manera se extraen relaciones semánticas, se distinguen expresiones superfluas, se descubren clases, etc. Su ventaja principal radica en que no es necesario un método de clasificación manual. Sin embargo, se necesita disponer de grandes cantidades de información para poder sacar relaciones y redundancias de forma apropiada.

Los sistemas no supervisados tienen una serie de características en común. Deben identificar si existe algún orden o jerarquía en los datos extraídos. Además, deben realizar un mapa topológico de los datos de entrada, de forma que patrones parecidos produzcan respuestas similares. También deben detectar qué componentes de los datos de entrada tienen un mayor valor para la recuperación, y obtener prototipos de la información que se pretende buscar.

Existen distintas técnicas para llevar a cabo un aprendizaje no supervisado. Las principales se describen a continuación.

2.2.3.1 Agrupamiento o clustering

Es una técnica estadística multivariable que permite la generación automática de grupos en los datos. Los vectores de atributos de los ejemplos no etiquetados se agrupan usando una función que calcula la distancia numérica o similitud entre dos pares de objetos. El resultado es un particionamiento de la colección de objetos en grupos de objetos relacionados. Existen varios libros que dan una interesante visión sobre el agrupamiento, como el de Kaufman y Rousseeuw (1990) y el de Theodoridis y Koutroumbas (2003).

En el agrupamiento influyen diversos factores:

- Las *propiedades* o *atributos* de los objetos que representan el conjunto de datos.

- La *función matemática* que mide la distancia entre dos objetos. Típicamente se usan la distancia euclídea o el producto escalar.

- Las *restricciones* a las que está sujeto el conjunto de datos, principalmente la elección del número de clusters.

En EI, el agrupamiento es útil cuando no existen ejemplos de entrenamiento, la información cambia dinámicamente o se pretende extraer ciertas propiedades o clases de la información. Se distinguen dos aplicaciones muy útiles:

- *Extracción de correferencias en los nombres de una frase*. Se trata de relacionar nombres y pronombres que se refieren a la misma persona, cosa, lugar, fecha, etc. Inducir estas relaciones a partir del texto es un problema complejo, pero

mediante técnicas de agrupamiento y añadiendo restricciones, se puede resolver con bastante menor coste computacional, tanto en tiempo como en memoria.

- *Correferencias de frases en diferentes documentos*. Recuperar cadenas de caracteres en distintos ficheros y que hagan referencia al mismo contexto. Esto es de gran utilidad para los buscadores y recuperadores de información que actualmente incluyen algunos sistemas operativos.

2.2.3.2 Auto-entrenamiento (Self-training)

Son técnicas de aprendizaje supervisado que incrementalmente aprenden un clasificador. Este clasificador se basa en un conjunto origen de ejemplos etiquetados y otro de ejemplos no etiquetados. Los no etiquetados son etiquetados progresivamente con el clasificador actual hasta que el clasificador entrenado alcanza un cierto nivel de precisión sobre el conjunto de test (McCallum et al., 1999).

2.2.3.3 Auto-entrenamiento paralelo (Co-training)

Dos o más clasificadores se entrenan usando el mismo conjunto origen de ejemplos etiquetados, pero cada clasificador se entrena con un subconjunto distinto de atributos (Blum y Mitchell, 1998). Los atributos normalmente se separan de tal modo que atributos de diferentes conjuntos son condicionalmente independientes. Suele decirse que estos subconjuntos son distintos puntos de vista que los clasificadores tienen del conjunto de entrenamiento.

2.2.3.4 Aprendizaje competitivo (active learning)

El aprendizaje competitivo es un planteamiento prometedor en los contextos de EI. Este método requiere algo más de supervisión o implicación humana. Con esta técnica un humano es el encargado de etiquetar todos los ejemplos, pero será la máquina la que elija cuidadosamente dicho conjunto limitado de ejemplos (Shen et al., 2004).

2.2.3.5 Mapas autoorganizativos

Son un tipo de redes neuronales artificiales. Se diferencian dos capas, una primera *capa de entrada* y una segunda *capa de competición*. La primera recibe el vector de información de entrada y propaga por las conexiones para que llegue la información a la capa de competición. En la segunda, cada célula que la compone produce una salida (típicamente basada en el cálculo de distancias, por ejemplo, la distancia euclídea). Todas estas salidas se comparan entre sí, para seleccionar aquella que produzca la mejor de ellas. Ésta será la ganadora y un posible valor futuro. Este proceso se repite hasta conseguir un cierto nivel de eficiencia o seguridad (Kohonen, 2001).

2.3. EL PROCESO DE EXTRACCIÓN

2.3.1 Arquitectura de un sistema de Extracción de Información

En esta sección se profundizará en los componentes típicos de un sistema de EI, los tipos de información que puede ser extraída y cuáles son los fundamentos teóricos existentes para asumir que es posible extraer estos tipos de información.

La Figura 2 muestra que la arquitectura de un sistema de EI típicamente tiene dos fases distintas: una fase de entrenamiento y una fase de despliegue. En la *fase de entrenamiento*, el ingeniero de conocimiento o el sistema adquiere los patrones de extracción necesarios, manualmente o usando aprendizaje automático respectivamente. En un primer paso, se elige un conjunto de textos representativo.

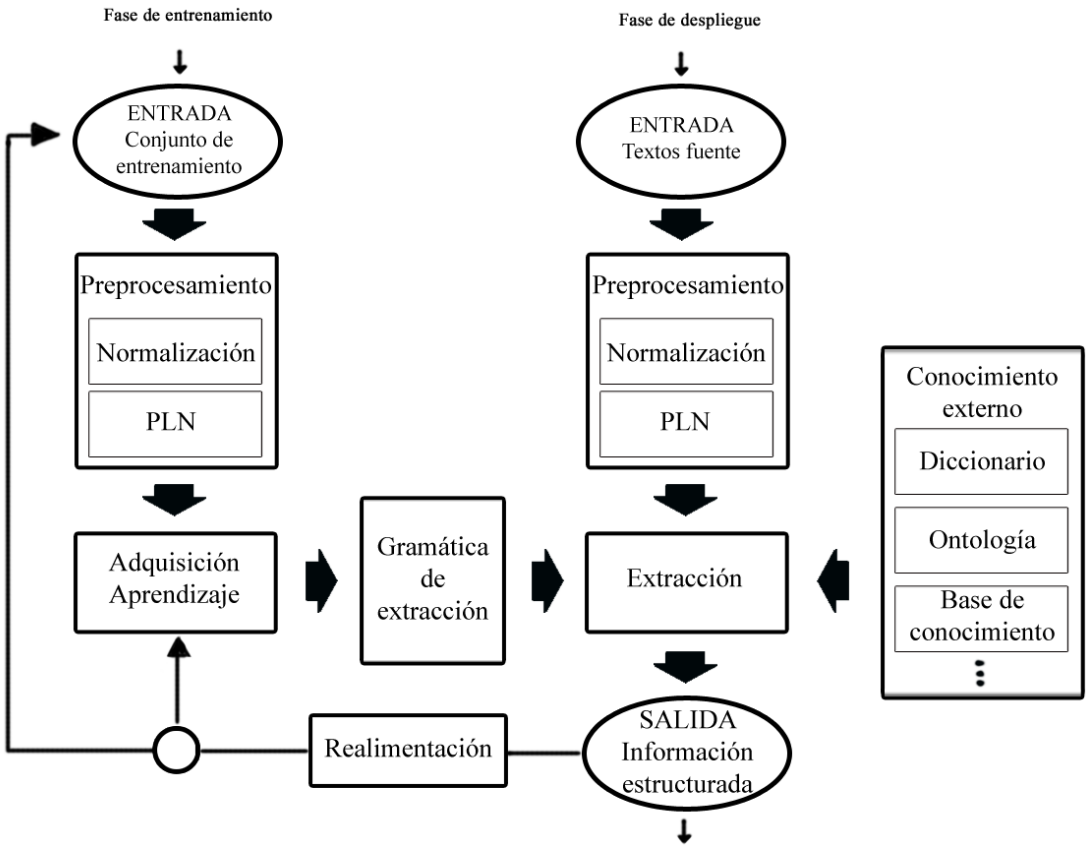


Figura 2. Un sistema de extracción de información típico

Antes de que los textos puedan utilizarse para extrapolar reglas de extracción, normalmente pasan una fase de preprocesamiento en la que sus características formales se normalizan. Otro paso perteneciente al preprocesamiento es el enriquecimiento de los datos textuales con metadatos lingüísticos que se usarán como parámetros en el proceso de adquisición. Para este fin pueden usarse distintas herramientas de PLN. La arquitectura típica de este bloque puede verse en la Figura 3 (Appelt e Israel, 1999).

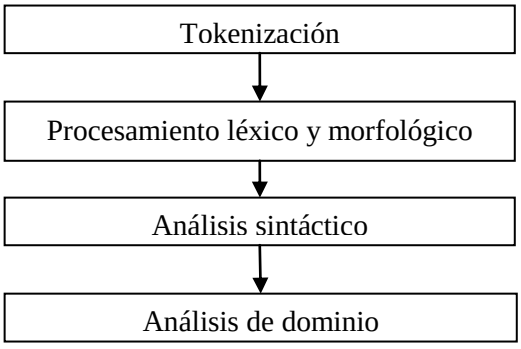


Figura 3. Arquitectura del bloque de Procesamiento de Lenguaje Natural

Si el problema se aborda de forma manual, un especialista utilizará el conjunto de textos preprocesados como base para escribir una *gramática de extracción*. En el caso de aprendizaje automático, el conjunto de textos se etiqueta normalmente de forma manual para indicar qué elementos en los textos son relevantes para la tarea de extracción. El sistema usará estos textos en la fase de aprendizaje para inducir automáticamente la gramática de extracción. Esta gramática puede aparecer en forma de función matemática que predecirá la clase de un ejemplo. También es posible que el conjunto de entrenamiento no se etiquete manualmente para utilizar técnicas de aprendizaje no supervisado.

En la fase de despliegue, el sistema de EI identifica y clasifica información semántica relevante en textos nuevos, es decir, textos que no se incluyeron en el conjunto de entrenamiento. El componente de preprocesamiento en la fase de despliegue es tan similar como sea posible al de la fase de aprendizaje. Después del preprocesamiento los textos de entrada pasan a la fase de extracción, que usa la gramática de extracción que se aprendió en el paso de aprendizaje, y posiblemente algún conocimiento adicional. De esta forma se determina qué elementos en los textos de entrada son relevantes para la tarea de extracción y cómo se relacionan con ciertas clases semánticas.

Los elementos textuales se extraen, clasifican y expresan en un formato estructurado. Algunos sistemas también tienen un mecanismo de realimentación, en el que la salida final del sistema se corrige y se emplea para reentrenar el componente de aprendizaje (aprendizaje incremental).

Es obvio que la tarea principal de un sistema de EI es la extracción de información semántica de textos, y esta información se define de antemano en el proceso de extracción. En cualquier texto pueden extraerse diferentes clases de información semántica, dependiendo del tamaño de las unidades lingüísticas buscadas y el contexto lingüístico cubierto por el sistema.

2.3.2 Algunas tareas de Extracción de Información

Hay un cierto número de tareas típicas de EI que se han investigado profusamente en los últimos años. Estas tareas incluyen reconocimiento de entidades con nombre, resolución de correferencia, reconocimiento de función semántica, reconocimiento de relación entre entidades y reconocimiento de expresiones temporales. Esta no es una lista exhaustiva de tareas de extracción, pero se utilizan frecuentemente para ilustrar el funcionamiento de distintos algoritmos de extracción.

2.4.2.1. Reconocimiento de Entidades con Nombre (Named Entity Recognition)

La tecnología más simple y más fiable de EI es la de Reconocimiento de Entidades con Nombre. Estos sistemas identifican todos los nombres de persona, lugares, organizaciones, etc.

Ejemplo:

John Smith trabaja para *IBM*.

↓ ↓

Persona Compañía

2.4.2.2. Resolución de correferencia (Coreference Resolution)

Dos o más grupos nominales son correferentes cuando se refieren a la misma situación descrita en un texto. Muchas referencias en un texto están codificadas como referencias fóricas, es decir, elementos lingüísticos que en lugar de codificar directamente el significado de una entidad, se refieren a una descripción directa de la entidad aparecida antes o después en el texto. Se las conoce como referencias anafóricas y catafóricas respectivamente.

Ejemplo:

Juan vendrá mañana a verme. Tengo algo para *él*.

Juan y *él* se refieren en este texto a la misma entidad. En este caso, *él* es una referencia anafórica.

Este proceso es menos relevante para los usuarios que otras tareas de extracción. Mientras que otras tareas producen una salida que es de obvia utilidad para el usuario de la aplicación, esta tarea es más relevante para las necesidades del desarrollador de la aplicación. Es un proceso más impreciso que el de reconocimiento de entidades con nombre.

2.4.2.3. Reconocimiento de función semántica

El reconocimiento de la función semántica trata de asignar funciones semánticas a los componentes (sintácticos) de una oración. Se consideran ciertas acciones o estados, sus participantes y sus circunstancias. Las funciones semánticas pueden definirse de manera muy general o más específica.

Ejemplo:

Ella alzó su mano para saludar.

Agente Parte del cuerpo Causa

2.4.2.4. Reconocimiento de relación entre entidades

Se detecta la relación entre dos o más entidades y se clasifica con una función semántica.

Ejemplo:

John Smith trabaja para IBM.

Persona Relación Compañía

trabaja para

2.4.2.5. Reconocimiento de expresiones temporales

Otra tarea importante es la detección de expresiones temporales en el texto. En este grupo están incluidas tanto las expresiones absolutas (*17 de julio de 1999, 12:00, verano del 69*) como las relativas (*ayer, la semana pasada, el próximo milenio*). También son dignas de atención las duraciones (*una hora, dos semanas*), expresiones ancladas a eventos (*dos días antes de salir*) y repeticiones (*todas las semanas*). A partir de las expresiones temporales reconocidas puede reconstruirse la línea temporal de diferentes eventos. Algunas relaciones temporales básicas son: X antes de Y, X es igual a Y, X coincide con Y, X se solapa con Y, X durante Y, X empieza en Y, X termina en Y, etc. El reconocimiento de una línea temporal implica formas sofisticadas de razonamiento temporal.

Ejemplo:

El *16 de abril de 2005* aprobé mi examen final. *Las tres semanas anteriores* estudié mucho.

26 de marzo de 2005 → 15 de abril de 2005: Estudiar
16 de abril de 2005: Examen

Las tareas de extracción que se han explicado son en general bastante independientes del dominio. Sin embargo, permiten identificar muchos detalles de un evento (por ejemplo, tiempo o localización). Pueden definirse tareas de extracción dependientes del dominio para completar la descripción (por ejemplo, el número de víctimas de un ataque terrorista, los síntomas de una enfermedad de un paciente, etc.).

En este nivel, la EI se interesa principalmente en extraer información sobre eventos individuales, el estado de los participantes y su situación espacial, temporal o causal.

Los eventos en el mundo real nunca aparecen aislados, sino que son parte de eventos más complejos que están enlazados causalmente unos con otros. Los humanos reconocen estos eventos enlazados como eventos complejos porque típicamente ocurren en cierto orden. A estos eventos estereotipados se los conoce como *escenarios* complejos. El objetivo final de la EI es reconocer escenarios y enlazarlos con modelos abstractos que reflejen eventos complejos del mundo real.

Los eventos complejos no son las estructuras semánticas más largas que pueden encontrarse en datos textuales. Generalmente son parte de estructuras multi-evento que pueden ordenarse cronológicamente para expresar una historia completa. Estas estructuras normalmente abarcan múltiples textos, y se distinguen de los escenarios en que la causalidad no suele ser tan importante como la cronología. Al final, debería ser posible extraer cronologías complejas de eventos a partir de textos completos y localizar escenarios, eventos aislados y entidades participantes en dichos eventos en su marco temporal y causal.

2.4. EVALUACIÓN DE TECNOLOGÍAS DE EXTRACCIÓN DE INFORMACIÓN

2.4.1 Introducción

Cuando se diseñan tecnologías, es deseable poder evaluar los sistemas con el objetivo de ver cómo se comportan con respecto a un criterio de referencia, y en comparación a otras tecnologías existentes. Esto también ocurre para la EI. Dependiendo de la aplicación, se miden unos resultados u otros. Por ejemplo, en algunos casos obtener resultados de alta precisión es de primordial importancia, como cuando los resultados de la extracción no se controlan manualmente. En otros casos, donde la extracción automática sólo realiza un filtrado inicial de la información que al final se selecciona manualmente, es más importante una alta cobertura.

Una alta *precisión* (en inglés *precision*) significa que la información extraída no contiene o contiene muy pocos errores. Alta *cobertura* (en inglés *recall*) se refiere a la situación en la que toda o casi toda la información que debe ser extraída es realmente extraída. Un ejemplo de precisión podría ser obtener el precio de un vuelo de la Web. Un ejemplo de cobertura puede ser un caso en el que el analista quiera encontrar tanta información válida como sea posible sobre las localizaciones de un crimen concreto, para procesarse después junto con otras evidencias.

La EI normalmente no es una meta final, sino que ayuda en otras tareas tales como la RI, la generación de resúmenes o la minería de datos. Muchas medidas de evaluación se centran en una *evaluación intrínseca*, es decir, se mide el rendimiento de la tarea de extracción. También podría ser útil llevar a cabo una *evaluación extrínseca*, en la que se mide el rendimiento de otra tarea en la que la EI forma una parte integral. En lo sucesivo sólo se considerarán mecanismos de evaluación del primer tipo.

La EI es una tarea de clasificación. Las clases asignadas pueden compararse con la asignación de clases ideal, que normalmente es determinada por un experto. En muchas tareas de EI las clases pueden se asignan objetivamente y rara vez hay discusión sobre qué clases asignar (por ejemplo, resolución de entidades con nombre, resolución de correferencia). Sin embargo, existen tareas para las que la asignación no está tan clara.

2.4.2 Medidas clásicas de evaluación

La EI utiliza las medidas típicas de evaluación para tareas de clasificación de textos, siendo éstas la cobertura y precisión, su combinación formando la medida-F, y la exactitud.

La efectividad de la asignación automática de clases semánticas se computa directamente comparando los resultados de la asignación automática con las asignaciones manuales hechas por un experto. Cuando las clases no son mutuamente excluyentes, las decisiones de clasificación binaria son las más apropiadas.

Antes de pasar a explicar las medidas clásicas de evaluación, es necesario conocer la nomenclatura utilizada (Vilares, 2008):

- **Claves**: conjunto de registros extraídos por un experto (registros de referencia).
- **Respuestas**: conjunto de registros extraídos por el sistema (registros a evaluar). Según la evaluación dada, se distinguirán los siguientes tipos de respuesta:
  - Correcta → Respuesta = Clave
  - Parcial → Respuesta $\cong$ Clave
  - Incorrecta → Respuesta $\neq$ Clave
  - Espuria → SI respuesta, NO clave
  - Perdida → NO respuesta, SI clave

Las medidas que se podrán calcular serán las siguientes:

- **Subgeneración (*undergeneration*)**: porcentaje de registros sin extraer.

$$UG = \frac{\#\,perdidas}{\#\,claves}$$

- **Sobregeneración (*overgeneration*)**: porcentaje de respuestas "de más".

$$overgeneration = \frac{\#\,espurias}{\#\,respuestas}$$

- **Precisión (*precision*)**: porcentaje de respuestas correctas. Mide la capacidad para extraer sólo registros correctos[4].

$$P = \frac{\#\,correctas + \#\,parciales / 2}{\#\,respuestas}$$

[4] Para los cálculos se precisión y cobertura, se están teniendo en cuenta tanto respuestas correctas como parciales, aunque estas últimas sólo contabilizan la mitad (Vilares, 2008). Esta forma de cálculo no es un estándar. Según el contexto, puede ocurrir que este tipo de respuestas no sean consideradas o que sean contabilizadas con un peso menor.

- **Cobertura (*recall*)**: porcentaje de registros extraídos. Mide la capacidad para extraer todos los registros correctos.

$$R = \frac{\#correctas + \#parciales/2}{\#claves}$$

- **Medida-F (*F-measure*)**: combina cobertura y precisión en una única medida.

$$F = \frac{(\beta^2 + 1) \cdot P \cdot R}{\beta^2 \cdot P + R}$$

donde β es un factor que indica la importancia relativa de la cobertura y la precisión. Si en la evaluación se considera que la precisión y cobertura tienen igual importancia, se tomará β = 1. En este caso la medida-F es conocida como F_1 (*media armónica*):

$$F_1 = \frac{2 \cdot P \cdot R}{P + R}$$

Idealmente, la cobertura y la precisión están cerca de 1 (así como la medida-F) y la sobregeneración y subgeneración cerca de 0. Los errores de cobertura suelen conocerse como *falsos negativos*, mientras que los de precisión son *falsos positivos*.

Cuando se comparan dos clasificadores, es deseable tener una única medida de eficiencia. La medida-F se utiliza comúnmente para combinar los valores de cobertura y precisión en una única medida:

$$F = \frac{(\beta^2 + 1) \cdot P \cdot R}{\beta^2 \cdot P + R}$$

donde β es un factor que indica la importancia relativa de la cobertura y la precisión. Cuando β es igual a 1, es decir, la cobertura y la precisión tienen la misma importancia, y la medida se conoce como *media armónica* (medida-F_1).

En EI deben evaluarse tanto la detección como la clasificación de información. Para ambas tareas normalmente se emplean las mismas medidas de evaluación. El resultado de la EI es frecuentemente una asignación probabilística. Ninguna de las medidas anteriores tiene en cuenta la probabilidad de la asignación.

2.4.3 Medidas alternativas de evaluación

En los casos en los que la información se clasifica agrupando los *tokens* o unidades, se han diseñado medidas de evaluación adecuadas que son una variación de las clásicas medidas de precisión y cobertura. Estas medidas normalmente se ilustran con la tarea de resolución de correferencia. Una opción para construir cadenas de correferencia de grupos nominales puede ser realizar agrupamientos de grupos nominales.

Cuando se evalúa el agrupamiento en EI, a menudo se utiliza la métrica de Vilain (métrica oficial usada en MUC) o la métrica B-cubo (Bagga y Baldwin, 1998). En ambos casos, los *clusters* construidos manualmente por un experto se comparan con los *clusters* que son generados automáticamente.

El *algoritmo de Vilain* (Vilain, 1995) tiene en cuenta el número de entidades que se deberían añadir 1) a la salida automática para llegar al agrupamiento manual y 2) a la salida manual para llegar a la automática. El primer número influye en la medida de cobertura *R*, mientras que el segundo influye en la medida de precisión *P*.

El *algoritmo B-cubo* (Bagga y Baldwin, 1998) tiene cierto parecido con el anterior. Se tiene en cuenta el número de entidades que se deberían añadir 1) a la salida automática para llegar a la manual y 2) a la salida manual para llegar a la automática. Formalmente, dados *n* objetos, se define para cada objeto *i*:

$$R_i = \frac{co_i}{mo_i}$$

$$P_i = \frac{co_i}{ao_i}$$

donde co_i = número de objetos correctos en el cluster construido automáticamente que contiene al objeto *i*

mo_i = número de objetos en el cluster construido manualmente que contiene al objeto *i*

ao_i = número de objetos en el cluster construido automáticamente que contiene al objeto *i*

La cobertura final *R* y la precisión final *P* considerando a los *n* objetos del agrupamiento se calculan respectivamente de la siguiente forma:

$$R = \sum_{i=1}^{n} w_i \cdot R_i$$

$$P = \sum_{i=1}^{n} w_i \cdot P_i$$

donde los w_i son pesos que indican la importancia relativa de cada objeto (por ejemplo, en resolución de correferencia el pronombre *i* podría tener un peso distinto que el nombre *i*). Todos los w_i deberían sumar 1. Normalmente se les da el valor 1/*n*.

Tanto el algoritmo de Vilain como el de B-cubo incorporan alguna forma de subjetividad al medir la validez de los clusters. La métrica de Vilain se centra en "¿qué necesito para conseguir el resultado correcto?", y no en "¿es el resultado que el sistema obtiene correcto o no?". El algoritmo de Vilain sólo recompensa a los objetos que están implicados en alguna relación, y todos los objetos se tratan de forma similar. En el algoritmo B-cubo, la relación de un objeto con otros en su cluster puede reforzarse más o menos con distintos pesos.

2.5. CONFERENCIAS Y PROGRAMAS DE INVESTIGACIÓN

2.5.1 MUC (Message Understanding Conferences)

Las conferencias MUC (*Message Understanding Conferences*) fueron iniciadas y financiadas por DARPA (*Defense Advanced Research Projects Agency*) a finales de los 80 para fomentar el desarrollo de nuevos y mejores métodos de EI (Voorhees, 2001). El carácter de estas conferencias (muchos equipos de investigación compitiendo unos contra otros) requirió el desarrollo de distintos estándares de evaluación, como por ejemplo la precisión y la cobertura. Los resultados de estas evaluaciones se presentaron en conferencias, en las que desarrolladores y evaluadores compartían sus descubrimientos y especialistas del gobierno describían sus necesidades (National Institute of Standards and Technology, 2001).

A mediados de los 90 las conferencias MUC comenzaron a proporcionar datos preparados y definiciones de tareas. Además facilitaron software de evaluación totalmente automatizado para medir el rendimiento, tanto de humanos como de los sistemas implementados. Sólo en la primera conferencia (MUC-1) los participantes pudieron elegir el formato de salida para la información extraída. A partir de la segunda el formato estaba predefinido.

La complejidad de las tareas definidas creció, pasando de la simple elaboración de una base de datos de eventos encontrados en noticias a la producción de múltiples bases de datos de información compleja extraída de múltiples fuentes de noticias en muchos idiomas. Para cada tema se daban varios campos, con el objetivo de rellenarlos usando información del texto. Algunos campos típicos eran, por ejemplo, la causa, el agente, el lugar y la hora de un evento, las consecuencias, etc. El número de campos fue creciendo de conferencia en conferencia.

Los temas y textos fuente que se procesaron en estas conferencias se centraban en temas civiles y militares.

Tabla 1. Temas de las distintas conferencias MUC

Conferencia	*Año*	*Texto fuente*	*Tema*
MUC-1	1987	Informes militares	Operaciones de flota
MUC-2	1989	Informes militares	Operaciones de flota
MUC-3	1991	Noticias	Actividades terroristas en América Latina
MUC-4	1992	Noticias	Actividades terroristas en América Latina
MUC-5	1993	Noticias	Microelectrónica
MUC-6	1995	Noticias	Negociación de disputas laborales
MUC-7	1997	Noticias	Accidentes de avión y lanzamientos de cohetes

2.5.2 FASTUS

SRI International desarrolló un sistema de EI llamado FASTUS (*Finite State Automata-based Text Understanding System)* para aplicarlo a tareas generales de extracción (Appelt et al., 1993). Es un sistema que sirve para extraer información de textos en inglés, japonés y otros idiomas. Funciona esencialmente como un conjunto de transductores de estado finito no determinista y en cascada. Se aplican varias etapas de procesamiento a la entrada, se reconocen los patrones especificados y se construyen las estructuras correspondientes. Las estructuras construidas en cada etapa constituyen la entrada a la etapa siguiente.

FASTUS se ideó originalmente en 1991 como un sistema de preprocesamiento. A mediados de 1992, viendo su alto rendimiento en las tareas de la conferencia MUC-4 (se obtuvo una cobertura del 44% y una precisión del 57%), se decidió desarrollarlo como un sistema completo.

La idea principal de FASTUS es separar el procesamiento en varias etapas. Las primeras reconocen objetos lingüísticos de menor tamaño y funcionan de manera correcta independientemente del dominio. Utilizan conocimiento puramente lingüístico para reconocer la estructura sintáctica de las oraciones, sin requerir prácticamente

ninguna modificación al cambiar de dominio. Las últimas etapas toman estos elementos lingüísticos como entrada y localizan patrones dependientes del dominio entre ellos.

La última versión de FASTUS hacía uso de cinco niveles de procesamiento:

1. *Palabras complejas*: Esto incluye el reconocimiento de nombres propios y palabras compuestas.
2. *Grupos básicos*: Las oraciones están divididas en grupos nominales, grupos verbales y partículas.
3. *Grupos complejos*: Se identifican grupos nominales complejos y grupos verbales complejos.
4. *Eventos de dominio*: La secuencia de grupos producida en el nivel 3 se analiza en busca de patrones para eventos de interés. Cuando se encuentran, se crean estructuras que codifican la información.
5. *Estructuras combinadas*: Algunas estructuras que aparecen en diferentes partes del texto se combinan si aportan información sobre la misma entidad o evento.

El sistema FASTUS fue uno de los sistemas de extracción más efectivos desarrollados en los 90. Poseía numerosas ventajas: además de ser bastante simple (conjunto de autómatas en cascada), era efectivo y su tiempo de ejecución era pequeño.

2.5.3 TREC

Las TREC (*Text Retrieval Conference*), patrocinadas por el NIST (*National Institute of Standards and Technology*) y el Departamento de Defensa de los Estados Unidos, son una serie de trabajos en curso que comenzaron en 1992 (Text Retrieval Conference, s.f.). No están directamente relacionadas con la EI, sino más bien con la RI, ya que su propósito principal era apoyar la investigación en esta área.

En particular, los trabajos TREC tienen las siguientes metas:

- Fomentar la investigación en RI basándose en grandes colecciones de test.
- Incrementar la comunicación entre la industria, el mundo académico y el gobierno creando un foro abierto para el intercambio de ideas de investigación.
- Acelerar la transferencia de tecnología desde los laboratorios de investigación hasta los productos comerciales mostrando mejoras sustanciales en las metodologías de RI.
- Incrementar la disponibilidad de técnicas apropiadas de evaluación para su uso por la industria y el mundo académico, incluyendo el desarrollo de nuevas técnicas de evaluación más aplicables a los sistemas actuales.

Un comité formado por representantes del gobierno, de la industria y del mundo académico es el encargado de supervisar TREC. Para cada TREC, el NIST proporciona un conjunto de documentos y preguntas de test. Los participantes ejecutan sus propios sistemas de recuperación sobre los datos.

En los últimos años aumentó tanto el número de sistemas participantes como el número de tareas. En TREC 2003 participaron 93 grupos representando a 22 países distintos. Las colecciones de test y el software de evaluación están disponibles para la comunidad investigadora, con el fin de que las organizaciones puedan evaluar sus propios sistemas. Los objetivos se alcanzaron con éxito, llegando a conseguir que la efectividad de los mecanismos de recuperación se multiplicara por dos en los seis primeros años de TREC.

2.5.4 ACE

El objetivo del programa ACE (*Automatic Content Extraction*) es desarrollar la capacidad de extraer significado a partir de fuentes multimedia (Doddington et al., 2004). Estas fuentes incluyen textos, audio e imágenes.

Este programa comenzó en 1999 con un estudio pensado para identificar aquellas tareas de extracción clave que sirvieran como objetivo de la investigación para el resto del programa. Se llegó a la conclusión de que dichas tareas eran la extracción de entidades, relaciones y eventos. En líneas generales, el programa ACE persigue los

mismos objetivos que su precursor, el programa MUC. Sin embargo, el programa ACE trata de que los objetivos de la investigación se centren en los objetos buscados en lugar de las palabras del texto.

El programa ACE, en un esfuerzo por estimular la investigación en EI, persigue cuatro retos distintos:

1. *Reconocimiento de entidades, no sólo de nombres*. En la tarea de rastreo y detección de entidades, todas las menciones de una entidad, ya sea con un nombre, una descripción o un pronombre, deben encontrarse y agruparse en clases de equivalencia basadas en referencias a la misma entidad. Por tanto, la resolución de correferencia es aquí fundamental.

2. *Reconocimiento de relaciones*. La tarea de detección de relaciones y caracterización requiere detectar relaciones entre entidades. Hay cinco tipos generales de relaciones: rol, parte, localización, localización relativa y relación social.

3. *Extracción de eventos*. Detección y caracterización de eventos.

4. *La extracción se mide no solamente en el texto, sino también en la voz y en Reconocimiento Óptico de Caracteres*. Moverse más allá de la búsqueda de nombres es un salto crucial para modalidades distintas al texto. La habilidad para relacionar dos cadenas en un entorno ruidoso puede verse degradada si se compara con la búsqueda de cadenas en entornos aislados. Además, la falta de caso y de puntuación, incluyendo la falta de marcadores de límites, supone un desafío para el análisis sintáctico completo de la voz.

Las tareas de ACE mejoran evaluación tras evaluación. De esta forma la especificación de cada tarea y los tipos de objetos involucrados varían año tras año. Desde el año 2005, el programa ACE está realizando muchos esfuerzos en la extracción de eventos de textos.

3. DISEÑO DEL SISTEMA

3.1. INTRODUCCIÓN

En el presente capítulo se describirá la arquitectura y diseño del sistema de EI implementado, explicando el funcionamiento de los bloques que formarán parte del mismo.

Como ya se comentó en la introducción, el objetivo que se persigue es el diseño de un sistema que sea capaz de extraer relaciones entre artículos de Wikipedia, apoyándose para ello en los enlaces contenidos en los mismos. Estas relaciones serán etiquetadas dentro de distintas categorías semánticas. Para llevar a cabo la búsqueda de relaciones, el diseño del sistema se ha basado en la búsqueda de patrones de extracción en el texto.

Para explicar más claramente lo que se pretende con el sistema, se ilustrará utilizando el siguiente fragmento de un artículo real de Wikipedia (Isaac Newton[5]):

Sir Isaac Newton fue un físico, filósofo, inventor, alquimista y matemático inglés, autor de los *Philosophiae naturalis principia mathematica*, más conocidos como los *Principia*, donde describió la ley de gravitación universal y estableció las bases de la Mecánica Clásica mediante las leyes que llevan su nombre. Entre sus otros descubrimientos científicos destacan los trabajos sobre la naturaleza de la luz y la óptica (que se presentan principalmente en el Opticks) y el desarrollo del cálculo matemático.
…
Nació el 25 de diciembre de 1642 (correspondiente al 4 de enero de 1643 del nuevo calendario) en Woolsthorpe, Lincolnshire, Inglaterra; fue hijo de dos campesinos puritanos, aunque nunca llegó a conocer a su padre, pues había muerto en octubre de 1642.
…
En 1663 conoció a Isaac Barrow, quien le dio clase como su primer profesor Lucasiano de matemática.
…
Mantuvo correspondencia con su amigo, el filósofo John Locke, en la que, además de contarle su mal estado, lo acusó en varias ocasiones de cosas que nunca hizo.
…
En 1684 Newton informó a su amigo Edmund Halley de que había resuelto el problema de la fuerza inversamente proporcional al cuadrado de la distancia.
…
Después de escribir los *Principia* abandonó Cambridge mudándose a Londres donde ocupó diferentes puestos públicos de prestigio siendo nombrado Preboste del Rey, magistrado de Charterhouse y director de la Casa de Moneda.
…

Figura 4. Fragmento de artículo de Wikipedia

[5] http://es.wikipedia.org/wiki/Isaac_Newton

A partir del texto aparecido en la figura anterior, el sistema debería ser capaz de obtener una salida similar a la siguiente:

```
FECHA_NACIMIENTO:
25 de diciembre de 1642
**********
LUGAR_NACIMIENTO:
Woolsthorpe
**********
CONOCIDOS:
Isaac Barrow
**********
AMISTADES:
John Locke
Edmund Halley
**********
PROFESION:
físico
filósofo
inventor
alquimista
matemático
**********
INVENCIONES_CREACIONES:
Philosophiae naturalis principia mathematica
ley de gravitación universal
Opticks
**********
LUGARES_RESIDENCIA:
Londres
**********
OTRAS_RELACIONES:
LINK: profesor Lucasiano
Relaciones encontradas:
   conoció a Isaac Barrow, quien le dio clase como su primer profesor Lucasiano de matemática
_____
LINK: Mecánica Clásica
Relaciones encontradas:
   estableció las bases de la Mecánica Clásica
_____
LINK: luz
Relaciones encontradas:
   trabajos sobre la naturaleza de la luz
_____
LINK: Casa de la Moneda
Relaciones encontradas:
   director de la Casa de la Moneda
…
…
```

Figura 5. Ejemplo de salida deseada

Como puede verse en el ejemplo anterior, se intentará etiquetar cada relación encontrada dentro de una categoría semántica apropiada. En los casos en los que no sea posible, se englobará dentro de una categoría genérica ("*OTRAS_RELACIONES*") y tratará de buscarse en el texto la expresión que mejor describa dicha relación.

En el capítulo anterior se estudiaron los componentes que conforman un sistema típico de EI y las relaciones entre ellos. Puede establecerse un paralelismo entre el esquema visto en el capítulo anterior (Figura 2) y la arquitectura propuesta para el sistema de EI diseñado (Figura 6).

Como ya se vio, el núcleo principal de un sistema de EI es el llamado módulo de ***extracción***. Este módulo será el encargado de producir la salida final del sistema (información estructurada). Para obtener la misma, el sistema se apoya en una gramática de extracción y una fuente de conocimiento externo, extrayendo una información específica de un volumen de texto que ha pasado por una fase de preprocesamiento.

La arquitectura general expuesta en el capítulo 2 utilizaba una fase de entrenamiento para generar la gramática (o patrones) de extracción. Sin embargo, para el sistema de EI que se ha implementado, se han diseñado unas reglas de extracción específicas adaptadas a los elementos concretos buscados en el texto. En este caso, la fase de aprendizaje se ha desarrollado de manera manual, adaptando y corrigiendo continuamente los patrones de extracción implementados para conseguir unos mejores resultados en una mayor cantidad de textos.

En la Figura 6 puede observarse el diagrama de bloques general del sistema de EI implementado. A lo largo de este capítulo se explicará detalladamente el funcionamiento de cada bloque individual y las decisiones tenidas en cuenta para su diseño.

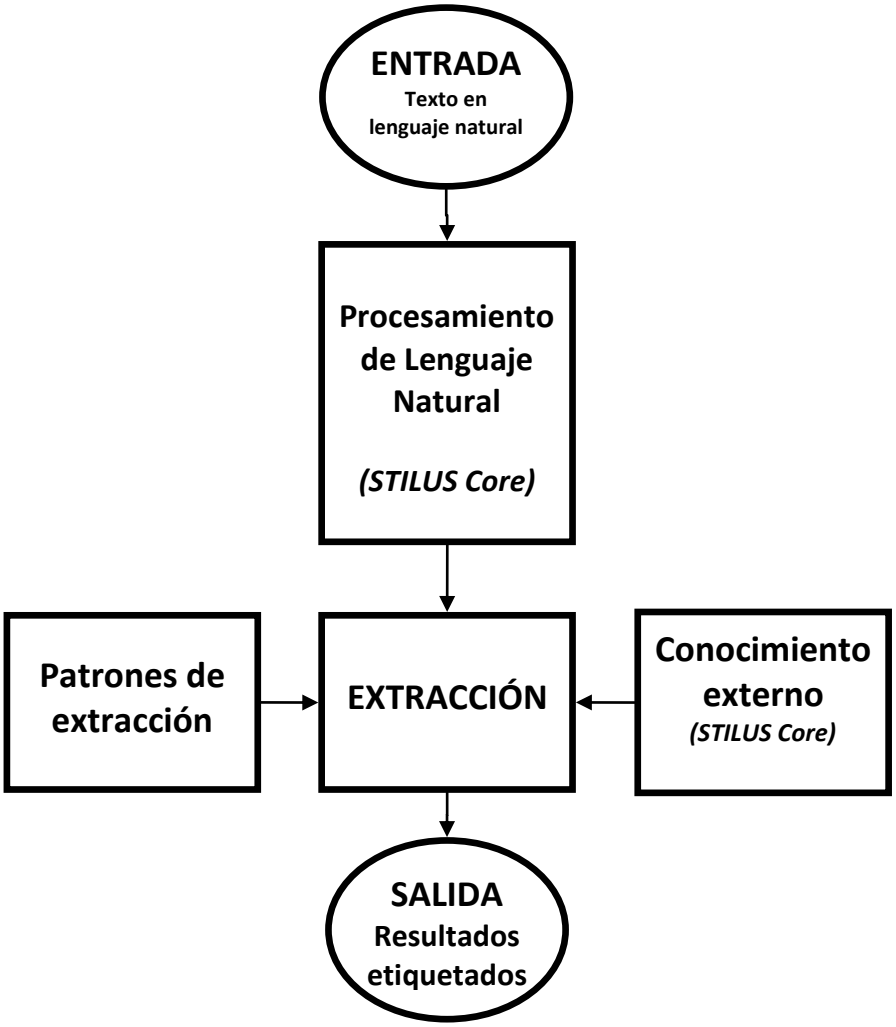


Figura 6. Arquitectura del sistema de EI diseñado

3.2. PREPARACIÓN PREVIA AL DISEÑO

Como punto de partida, los textos a analizar por el sistema serán siempre artículos extraídos de Wikipedia en castellano. Para facilitar el procesamiento y manejo de los mismos, existe la posibilidad de descargar una versión actualizada de todos los artículos en un único archivo XML, que puede encontrarse en la ubicación http://download.wikimedia.org/eswiki/.

La Wikipedia en formato XML contiene los artículos que componen la misma delimitados por una serie de etiquetas prefijadas. El formato de los mismos se ilustra en el siguiente ejemplo:

```
<page>
    <title>Juan de la Cierva y Codorníu</title>
      <id>1554</id>
      <revision>
          <id>11372360</id>
          <timestamp>2007-09-16T13:08:20Z</timestamp>
          <contributor>
            <ip>80.218.187.88</ip>
          </contributor>
          <text xml:space="preserve">'''Juan de la Cierva y Codorníu'''
([[Murcia]], [[España]], [[21 de septiembre]] de [[1895]] - [[Croydon]],
[[Inglaterra]], [[9 de diciembre]] de [[1936]]), [[Ingenieros|Ingeniero de
Caminos, Canales y Puertos]] y [[aviador]] [[España|español]], creador del
[[autogiro]].
    [[Imagen:Monumento_juan_de_la_cierva-
Murcia.JPG|right|thumb|350px|Monumento de Juan de la Cierva en Murcia]]

    Hijo del abogado, político y empresario [[Juan de la Cierva y
Peñafiel]], desde su infancia destacó su interés por las exhibiciones
aéreas que se realizaban como espectáculo y más tarde se trasladó a
[[Getafe]].

    [[Categoría: Aviadores de España|Cierva, Juan de la]]
    [[Categoría: Inventores de España|Cierva, Juan de la]]
    [[Categoría: Ingenieros de Caminos|Cierva, Juan de la]]
    [[Categoría: Murcianos|Cierva, Juan de la]]
    [[Categoría: Nacidos en 1895|Cierva, Juan de la]]
    [[Categoría: Fallecidos en 1936|Cierva, Juan de la]]
    [[Categoría: Muertes en accidentes aéreos|Cierva, Juan de la]]

    [[ca:Juan de la Cierva y Codorníu]]
    [[de:Juan de la Cierva]]
    [[en:Juan de la Cierva]]
    [[fi:Juan de la Cierva]]
    [[gl:Juan de la Cierva]]
    [[ja:???·?·?·????]]
    [[nl:Juan de la Cierva]]
    [[pl:Juan de la Cierva]]
    [[pt:Juan de la Cierva]]
    [[ru:??????, ???? ?? ??]]
    [[sv:Juan de la Cierva]]</text>
        </revision>
  </page>
```

Figura 7. Ejemplo de artículo de Wikipedia en formato XML

Como puede observarse, es necesario realizar una conversión previa del archivo XML antes de introducirlo en el sistema de EI. El texto de interés se encontrará siempre entre las etiquetas <*text*>. Además, dentro de este bloque aparecen ciertos códigos característicos de Wikipedia para denotar distintos tipos de formatos y funcionalidades. Entre los más frecuentes destacan los siguientes:

```
[[Enlace interno]]
[Enlace externo]
{{Plantilla}}
{{{Parámetro}}}
[[Categoría:]]
=Sección=
'''Negrita'''
''Cursiva''
```

Figura 8. Ejemplos de códigos utilizados en Wikipedia para dar formato al artículo

Todos estos códigos propios del formato del artículo deben ser eliminados, ya que es necesario contar con un texto limpio en lenguaje natural antes de proceder a la extracción de información del mismo. El objetivo será contar con un archivo de texto independiente por cada artículo de Wikipedia, lo que facilitará en gran medida el procesamiento posterior.

Además de llevar a cabo este paso previo de limpieza y normalización del texto, se aprovechará este módulo para extraer los *links* o enlaces internos que aparezcan en un artículo, almacenándolos en un segundo archivo. Se denomina *enlace interno* a un enlace a otro artículo de Wikipedia. El hecho de extraer estos enlaces dará la posibilidad posteriormente de establecer un grafo de relaciones de los mismos con el artículo principal a analizar.

El sistema de EI diseñado será totalmente aplicable a cualquier artículo en lenguaje natural que se utilice como entrada al mismo, no solamente a aquellos extraídos de Wikipedia. La ventaja de estos últimos es la de contar con una serie de enlaces dentro de los mismos con los que pueden buscarse relaciones de interés, aunque también sería perfectamente posible introducir manualmente un listado de conceptos de interés con los que buscar relaciones.

3.3. PROCESAMIENTO DE LENGUAJE NATURAL

El primer módulo del sistema de EI consistirá en un bloque de procesamiento de lenguaje natural (PLN). Aunque en un principio pueda parecer que la utilización de este módulo es innecesaria, y que podría pasarse directamente al diseño de la gramática y

patrones de extracción, posteriormente se comprobará que su uso disminuye mucho la complejidad del sistema.

La herramienta empleada para realizar este procesamiento ha sido **STILUS Core**[6]. Esta aplicación, desarrollada por la empresa DAEDALUS, proporciona una completa biblioteca software de herramientas para procesamiento lingüístico en castellano: filtrado, segmentación y etiquetado morfosintáctico de textos, análisis sintáctico, desambiguación morfológica, resúmenes, etc. (DAEDALUS, s.f.).

3.3.1 Opciones de ejecución de STILUS Core

La ejecución de STILUS Core utilizando un texto en lenguaje natural como entrada producirá un análisis de salida distinto dependiendo de las opciones que se fijen en el inicio. La ayuda de STILUS Core muestra las distintas opciones de ejecución:

```
/home/miracle/tools/stilus-core-es-2008/stilus-core-es:
  -d ruta     Directorio de datos (/home/miracle/tools/stilus-core-es-2008/)
  -t ruta     Directorio temporal (/tmp/)
  -m tipo     texto, html (defecto: texto)
  -utf8       codificación UTF8
  -ia         insensible a la acentuación
  -r          (sólo texto) El carácter \r separa líneas
  -br         (sólo html) La etiqueta BR separa párrafos
  -V version  Versión (sólo libros de estilo específicos)
Modos:
  -Mv         Versión de los recursos lingüísticos
  -Mam        Análisis morfológico
  -Mas        Análisis sintáctico
  -Mr         Resumidor de texto
  -Mgm        Generador morfológico
  -Mcv        Conjugador verbal
Opciones del análisis morfológico/sintáctico:
  -pd         Detectar palabras desconocidas
  -g          Desambiguar análisis
  -l          Imprimir descripción completa de etiquetas
  -i          Imprimir información semántica completa
  -n          Detectar entidades con nombre desconocidas (sin implementar)
  -s          Guardar las entidades con nombre en el diccionario personal (sin
implementar)
Opciones del resumidor de texto:
  -nf         Número de frases (3)
  -Pn         Puntuación de los nombres (10)
  -Pv         Puntuación de los verbos (5)
  -Pa         Puntuación de los adjetivos (0)
  -Pd         Puntuación de los adverbios (0)
  -Pg         Puntuación de las negritas (2)
  -mp         Mostrar las puntuaciones
```

Figura 9. Ayuda de STILUS Core

[6] http://www.daedalus.es/productos/stilus/stilus-core/

Para el sistema de EI a implementar, resultará de interés utilizar las siguientes opciones:

- ***-Mas: Análisis sintáctico.***

  La funcionalidad de análisis sintáctico permite detectar grupos de palabras que realizan una misma función en la oración. Así, se pueden detectar sintagmas nominales, verbales, preposicionales o adverbiales, así como su posible función dentro de cada frase.

  La función que desempeña un sintagma dentro de la oración (sujeto, verbo, objeto, etc.) resultará de gran interés en el proceso de extracción, ya que será determinante a la hora de obtener relaciones con otros conceptos.

- ***-l: Descripción completa de etiquetas*.**

  Resulta de utilidad contar con descripciones completas de las características gramaticales y sintácticas de una unidad concreta. Por ejemplo, puede ocurrir que cierta información que se quiera extrer siempre se encuentre dentro de un sintagma preposicional, por lo que resultaría conveniente contar con dicha información en el análisis.

  El siguiente ejemplo muestra dos salidas posibles de STILUS Core según se utilice o no esta opción:

```
en_Madrid    6      9      SEG [Diccionario general] GY----|en|en
~Sintagma preposicional

en_Madrid    6      9      SEG [Diccionario general] GY----|en|en
```

- ***-g: Desambiguar análisis.***

  En ocasiones una misma palabra o unidad sintáctica puede tener múltiples análisis posibles (p.ej.: *casa* → sustantivo femenino singular o presente del verbo *casar*). STILUS Core tratará de eliminar los análisis inválidos mostrando únicamente la opción correcta.

- ***-i: Información semántica completa***.

  Esta opción hace posible obtener información semántica para ciertas palabras o unidades sintácticas. Esta información puede incluir su temática, tipo de entidad, información geográfica, etc., lo que supondrá un interesante aporte para el proceso de EI.

3.3.2 Fases del análisis

Estableciendo una analogía con el sistema de PLN propuesto en el capítulo anterior (Figura 3), las fases del análisis llevado a cabo por STILUS Core pueden sintetizarse en las mostradas en la siguiente figura:

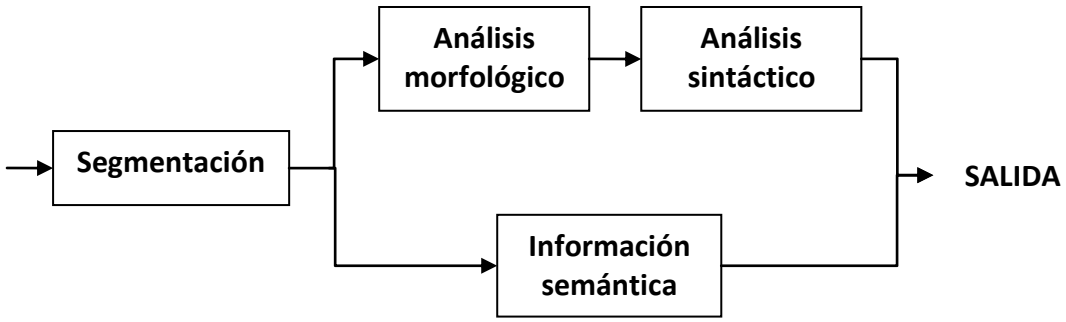


Figura 10. Fases del análisis de STILUS Core

Para comprender más fácilmente cada una de las fases de este análisis, se ilustrará con un ejemplo real de ejecución (Figura 11). La frase de entrada a la aplicación para este ejemplo ha sido "Juan *Sánchez nació en Madrid el 12 de abril de 1970*":

```
Juan_Sánchez    0       12      SEG [Diccionario general] GNMSS-
|Juan_Sánchez|Juan_Sánchez ~Sintagma nominal masculino singular sujeto
*{
Juan_Sánchez    0       12      SEG [Diccionario general] NPMS------
|Juan_Sánchez|Juan_Sánchez SemEntity=@inst@nofiction@PERSON@FULL_NAME@@@
~Nombre propio masculino singular
*}
nació   13      5       SEG [Diccionario general] GV-S--|nacer|nació
~Sintagma verbal singular
*{
nació   13      5       SEG [Diccionario general] VI-S3SIL-N5|nacer|nació
~Verbo léxico intransitivo 3ª persona singular pasado simple indicativo
*}
en_Madrid       19      9       SEG [Diccionario general] GY----|en|en
~Sintagma preposicional
*{
en      19      2       SEG [Diccionario general] Y-N8|en|en ~Preposición
Madrid  22      6       SEG [Diccionario general] GNUS--|Madrid|Madrid
~Sintagma nominal singular
*{
Madrid  22      6       SEG [Diccionario de Deportes] NPMS--N-
N6|Madrid|Madrid SemEntity=@inst@nofiction@ORGANIZATION@GAME_GROUP@@@
SemTheme=@SPORT@FOOTBALL SemRemission=@Real_Madrid_Club_de_Fútbol@@ ~Nombre
propio masculino singular    SEG [Diccionario general] NPMS--N-
N6|Madrid|Madrid
SemEntity=@inst@nofiction@LOCATION@GEO_POLITICAL_ENTITY@CITY@@
SemGeo=@@@Cundinamarca@@@Colombia@@@ ~Nombre propio masculino singular  SEG
[Diccionario general] NPFS--N-N6|Madrid|Madrid
SemEntity=@inst@nofiction@LOCATION@GEO_POLITICAL_ENTITY@CITY@@
SemGeo=@@@Cundinamarca@@@Colombia@@@ ~Nombre propio femenino singular
SEG [Diccionario general] NPMS--N-N6|Madrid|Madrid
SemEntity=@inst@nofiction@LOCATION@GEO_POLITICAL_ENTITY@CITY_PROVINCE@@
SemGeo=@@@Madrid@@Madrid@España@@@ ~Nombre propio masculino singular   SEG
[Diccionario general] NPFS--N-N6|Madrid|Madrid
SemEntity=@inst@nofiction@LOCATION@GEO_POLITICAL_ENTITY@CITY_PROVINCE@@
SemGeo=@@@Madrid@@Madrid@España@@@ ~Nombre propio femenino singular
*}
*}
el_12_de_abril_de_1970  29      22      SEG [Diccionario general] GNMSC-
|12_de_abril_de_1970|el_12_de_abril_de_1970 ~Sintagma nominal masculino
singular complemento circunstancial
*{
el      29      2       SEG [Diccionario general] TDMSN9|el|el ~Artículo
masculino singular
12_de_abril_de_1970     32      19      SEG [Diccionario general] NDMS--m-
|12_de_abril_de_1970|12_de_abril_de_1970 ~Nombre de fecha masculino singular
*}
*
```

Figura 11. Ejemplo de ejecución de STILUS Core

En el ejemplo anterior pueden observarse los distintos tipos de análisis realizados por STILUS Core:

Segmentación

El primer paso del procesamiento consiste en la separación del texto en unidades susceptibles de recibir análisis lingüístico. Éstas no siempre se corresponderán con palabras individuales separadas por espacios, sino que en ocasiones podrán abarcar grupos de dos o más palabras, como ocurre con el sujeto del ejemplo anterior (*Juan Sánchez*) o la fecha al final de la oración (*12 de abril de 1970*).

Información semántica

Como se ha comentado anteriormente, STILUS Core proporciona información semántica acerca de ciertas unidades o expresiones. Para ello hace uso internamente del módulo STILUS Sem, que obtiene rasgos semánticos tales como el tipo de entidad, la temática, remisión a otras entidades, tipo de relación o información semántica.

En el ejemplo mostrado en la Figura 11, puede verse la información semántica obtenida para *Madrid*.

Análisis morfológico

Para cada unidad analizada, la aplicación obtendrá la raíz o lema de la misma y sus rasgos morfosintácticos (como el género, el número, la persona, el tiempo verbal, el modo verbal, etc.). En el ejemplo puede verse este tipo de análisis realizado con el verbo *nacer*:

```
nació 13    5     SEG [Diccionario general] VI-S3SIL-N5|nacer|nació ~Verbo léxico intransitivo 3ª persona singular pasado simple indicativo
```

Análisis sintáctico

Una de las funciones más útiles de STILUS Core es la de realizar un análisis sintáctico superficial del texto. El objetivo es detectar grupos de palabras que realizan la misma función en la oración. Así, se pueden detectar sintagmas nominales, verbales, preposicionales o adverbiales, así como su posible función dentro de cada frase.

Gracias a esta funcionalidad, es posible construir a partir de cualquier texto un árbol sintáctico del mismo, en el que se refleje la pertenencia de ciertas unidades a

sintagmas de diverso tipo, y éstos a su vez estén incluidos en otros sintagmas o proposiciones de mayor tamaño. El siguiente ejemplo ilustra esta funcionalidad

```
El_árbol_que_he_visto     0     8     SEG [Diccionario general] GNMSS-
|árbol|El_árbol ~Sintagma nominal masculino singular sujeto
*{
El    0     2     SEG [Diccionario general] TDMSN9|el|el ~Artículo
masculino singular
árbol 3     5     SEG [Diccionario general] NCMS--N-N5|árbol|árbol
SemEntity=@inst@nofiction@NATURAL_OBJECT@LIVING_THING@FLORA@@
SemTheme=@NATURAL_SCIENCES@BOTANY ~Nombre común masculino singular
que_he_visto 9     17    SEG [Diccionario general] OSA-N----|que|que
~Proposición subordinada adjetiva complemento del nombre
*{
que   9     3     SEG [Diccionario general] GNUUS-|que|que ~Sintagma
nominal sujeto
*{
que   9     3     SEG [Diccionario general] RPMSUN8|que|que ~Relativo
pronominal masculino singular   SEG [Diccionario general] RPMPUN8|que|que
~Relativo pronominal masculino plural  SEG [Diccionario general]
RPFSUN8|que|que ~Relativo pronominal femenino singular    SEG [Diccionario
general] RPFPUN8|que|que ~Relativo pronominal femenino plural
*}
he_visto    13    8     SEG [Diccionario general] GV-S--|ver|he_visto
~Sintagma verbal singular
*{
he_visto    13    8     SEG [Diccionario general] VI-S1pTL-
N7|ver|he_visto ~Verbo léxico transitivo 1ª persona singular pretérito
perfecto indicativo
*}
*}
*}
da    22    2     SEG [Diccionario general] GV-S--|dar|da ~Sintagma
verbal singular
*{
da    22    2     SEG [Diccionario general] VI-S3PBL-N6|dar|da ~Verbo
léxico transitivo e intransitivo 3ª persona singular presente indicativo
*}
muchas_manzanas_rojas     25    21    SEG [Diccionario general] GNFPD-
|manzana|muchas_manzanas ~Sintagma nominal femenino plural objeto directo
*{
muchas 25    6     SEG [Diccionario general] QDFPU--N6|mucho|muchas
~Cuantificador determinante femenino plural
manzanas    32    8     SEG [Diccionario general] NCFP--N-
N4|manzana|manzanas ~Nombre común femenino plural
rojas 41    5     SEG [Diccionario general] APFP--NN5|rojo|rojas
~Adjetivo femenino plural postnominal
*}
.     46    1     SEG [Diccionario general] 1O--|.|. ~Signo de puntuación
individual
*
```

Figura 12. Ejemplo de árbol sintáctico realizado por STILUS Core

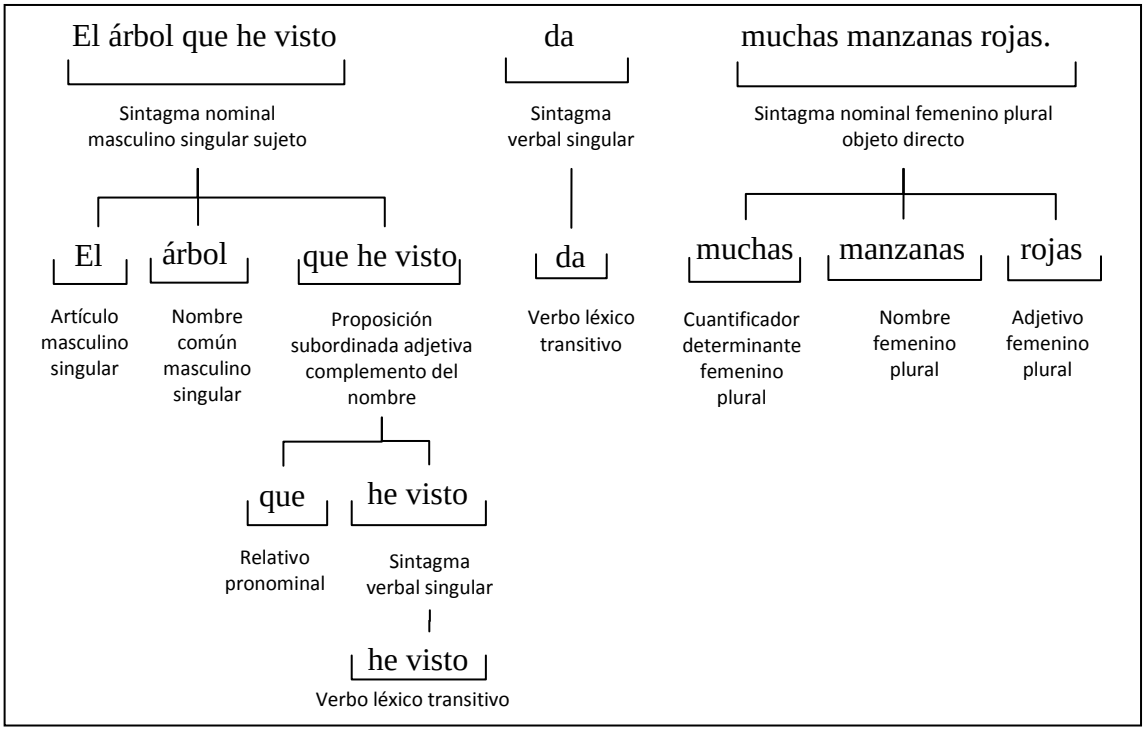


Figura 13. Ejemplo análisis realizado por STILUS Core mostrado en forma de árbol

Es importante recalcar que ninguna de las fases del análisis realizado por STILUS Core obtiene resultados correctos en un 100% de los casos. En ocasiones puede existir cierta ambigüedad en las oraciones a analizar y dar como resultado un análisis erróneo. Este aspecto deberá ser tenido en cuenta a la hora de evaluar el sistema de EI diseñado.

3.4. CONOCIMIENTO EXTERNO

El módulo de conocimiento externo aporta cierta información adicional para ayudar a determinar qué elementos de los textos de entrada son relevantes para el proceso de extracción y cómo se relacionan con ciertas clases semánticas (Moens, 2006).

Para el caso del sistema que se ha diseñado, existe una fuente principal de conocimiento que facilita el proceso de EI: la información semántica proporcionada por STILUS Core. La aplicación cuenta con una gran base de datos de información semántica que resultará de una gran utilidad. De la información proporcionada, hay una gran cantidad que puede ser utilizada en el proceso de extracción, como por ejemplo los nombres de persona, las localizaciones y las temáticas.

3.5. PATRONES DE EXTRACCIÓN

A la hora de diseñar los patrones de búsqueda en un sistema de EI, lo primero que debe tenerse en cuenta son las características de los textos con los que se está trabajando. Generalmente un sistema de EI se diseña para que funcione adecuadamente con textos de características similares (por ejemplo bases de datos de patentes, definiciones de diccionario, etc.) ya que su comportamiento será más fácilmente predecible.

Para el caso del sistema de EI que se ha diseñado, se utilizarán como textos de entrada artículos referidos a personajes (tanto históricos como de actualidad). En este tipo de textos hay ciertos datos que suelen aparecer con cierta frecuencia, y que pueden ser extraídos si se diseñan los patrones adecuados.

Una vez que se tiene claro el tipo de textos con los que se va a trabajar, debe decidirse qué aspectos de la información contenida en ellos quieren extraerse. Como se ha comentado anteriormente, el objetivo del sistema será tratar de etiquetar las relaciones del artículo analizado con otros artículos mediante los enlaces que aparecen en el mismo. Para etiquetar estas relaciones pueden definirse una serie de categorías, de forma que fuese posible obtener un grafo de relaciones con otros conceptos (Figura 14).

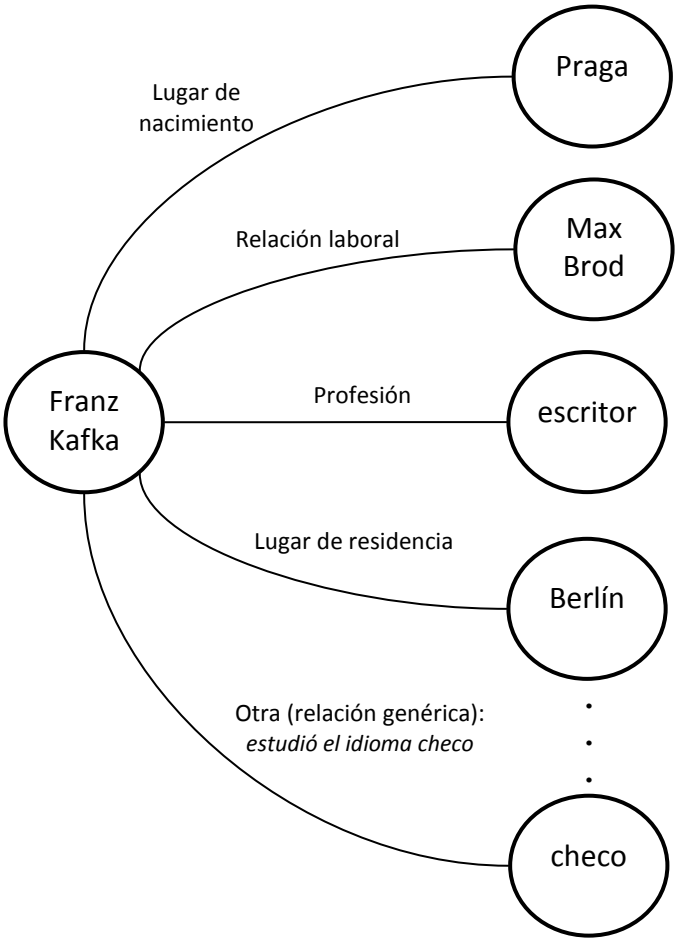


Figura 14. Ejemplo de grafo de relaciones etiquetadas extraídas por el sistema

Los nodos del anterior grafo son artículos de Wikipedia. Al estar trabajando con artículos de personajes, se tratará de buscar relaciones que aparezcan con frecuencia en este tipo de textos. Más concretamente, el sistema de EI que se ha diseñado tratará de etiquetar las relaciones encontradas asignándolas a uno de los siguientes tipos:

- Fecha de nacimiento
- Lugar de nacimiento
- Fecha de muerte
- Lugar de muerte
- Persona conocida
- Relación laboral
- Cónyuge
- Amistad

- Profesión
- Invención
- Lugar de residencia
- Lugar visitado
- Otra (relación genérica).

Como puede comprobarse, los patrones de extracción buscados se corresponden con características típicas que puedan aparecer en artículos sobre personajes. Si existiese la necesidad de adaptar el sistema para trabajar con otro tipo de textos, las relaciones a buscar variarían y por tanto los patrones de extracción también (pero el algoritmo presentado sería similar).

Aunque se explicará más detenidamente en el siguiente capítulo, la filosofía seguida a grandes rasgos para la búsqueda de patrones se ha basado en los siguientes aspectos:

- Se buscan en el texto palabras o expresiones que focalicen la información requerida. Por ejemplo, si se pretende localizar el lugar de nacimiento, se buscarían las apariciones del lema '*nacer*'.
- Se comprueba el sujeto de la oración para ver si correferencia al personaje del artículo. Para el ejemplo anterior, a partir de la oración '*Su mujer nació en Madrid*' no podría establecerse una relación de lugar de nacimiento, ya que el sujeto no se corresponde con el personaje del artículo.
- Se buscan sintagmas nominales con información semántica determinada. En ciertos casos también se comprueba si estos sintagmas están contenidos dentro de sintagmas preposicionales. Para el ejemplo del lugar de nacimiento, habría que tratar de encontrar localizaciones geográficas dentro de sintagmas preposicionales con la preposición '*en*'.
- Finalmente, se etiqueta el enlace dentro de la categoría correspondiente. Por ejemplo, '*LUGAR_NACIMIENTO*'.

Las relaciones que no puedan etiquetarse dentro de una categoría específica serán incluidas en una categoría genérica ("otra"). No obstante, el sistema tratará de extraer también el fragmento de oración que mejor describa dicha relación. Este tipo de

relación es la más complicada de implementar, por lo que la información extraída no siempre será del todo precisa. Sin embargo, a diferencia del resto de relaciones, puede ser aplicada a cualquier tipo de texto obteniendo un rendimiento similar.

En el capítulo 4 se describirá con mayor detalle la implementación y el diseño de las funciones necesarias para la creación de los patrones de extracción descritos.

3.6. PROCESO DE EXTRACCIÓN

Una vez que ya se han diseñado los patrones de extracción, el paso final a realizar para completar la arquitectura del sistema consiste en construir el bloque de extracción con el fin de obtener una salida etiquetada. Para ello, será necesario combinar los resultados proporcionados por los módulos anteriores.

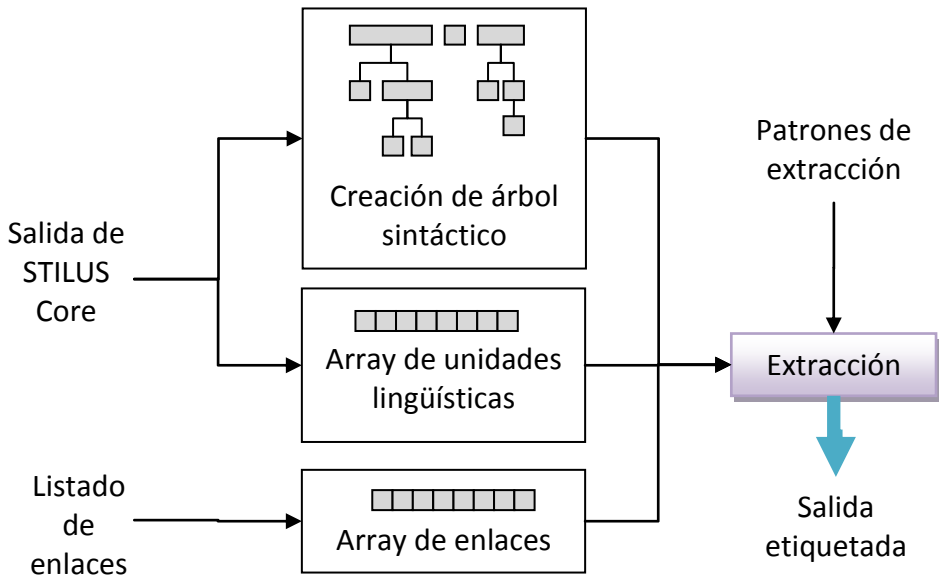


Figura 15. Diseño del módulo de extracción

Como se observa en la Figura 15, a partir de la salida de STILUS Core, el sistema construirá una estructura de datos en árbol y un array[7] de unidades sintácticas. El motivo de llevar a cabo este paso es que la salida dada por STILUS Core es en formato texto, lo que dificulta su manejo a lo largo del proceso de extracción. El hecho de poder trabajar con arrays y árboles facilitará el poder recorrer los nodos sintácticos de forma mucho más rápida y eficiente. El listado de enlaces que aparezcan en un artículo también será convertido en array para facilitar su manejo.

Trabajando con los arrays y árboles anteriormente mencionados y aplicando los patrones de extracción diseñados se podrá obtener la salida deseada: relaciones del artículo principal con otros conceptos clasificadas por categorías.

En el siguiente capítulo se profundizará con mayor detalle en el diseño de este y otros módulos del sistema de EI.

[7] Se utiliza el anglicismo "array" por su uso habitual en programación, en vez del término castellano "matriz".

4. IMPLEMENTACIÓN DEL SISTEMA

4.1. INTRODUCCIÓN

En este capítulo se explicará con mayor detalle la implementación del sistema de EI. Concretamente, la explicación estará centrada en describir las funciones diseñadas para construir el módulo de extracción y la búsqueda de información en el texto de acuerdo a unos patrones específicos.

El sistema se ha desarrollado en el lenguaje PHP. Se comentarán en primer lugar las razones por las que se ha elegido este lenguaje para el desarrollo de clases y funciones, y posteriormente se abordará la explicación del código desarrollado.

4.2. PHP

PHP (acrónimo recursivo que significa "*PHP: Hypertext Preprocessor*") es un lenguaje de *scripting* de propósito general y de código abierto (The PHP Group, 2009). Se utiliza generalmente en desarrollo web, ya que permite escribir páginas generadas dinámicamente, aunque puede adaptarse a muchas otras aplicaciones. Su sintaxis se basa en C, Java y Perl (Achour et al., 2009).

Además de su uso en servidores web, otra de las áreas donde PHP es frecuentemente utilizado es en la creación de *scripts* en línea de comandos. Esta opción puede resultar de mucha utilidad para el diseño de un sistema de EI, debido a la potencia de PHP para trabajar con textos de gran tamaño. PHP cuenta con funciones y herramientas muy útiles para procesamiento de texto.

A la hora de trabajar con archivos de texto, una de las características más útiles de PHP es la posibilidad de utilizar **expresiones regulares**. PHP permite utilizar expresiones regulares POSIX extendidas o tipo Perl. Para la implementación del sistema de EI se han utilizado expresiones tipo Perl (PCRE), ya que son más potentes que las primeras y soportan un mayor número de características (Achour et al., 2009).

La última versión liberada de PHP es la 5.3.0 (versión del 30 de junio de 2009). Durante el desarrollo de este proyecto la versión utilizada ha sido la 5.2.6.

4.3. EXPRESIONES REGULARES

Una expresión regular es simplemente una cadena de caracteres que describe un patrón. Los patrones de búsqueda de texto se utilizan con frecuencia en muchas aplicaciones actuales (motores de búsqueda, listado de archivos en un directorio, etc.). En el caso del sistema de EI diseñado, se buscarán diferentes patrones tanto en el texto como en los análisis realizados por STILUS Core para encontrar la información que se adapte a las características buscadas.

Como se ha mencionado en el punto anterior, PHP permite el uso de expresiones regulares compatibles con Perl (PCRE), lo cual aporta un gran número de ventajas. Las expresiones regulares de Perl permiten una eficiencia y flexibilidad casi inalcanzable en la mayoría de lenguajes de programación. El dominio de los aspectos más básicos de este tipo de expresiones permitirá manipular texto con una sorprendente facilidad.

Resultaría imposible explicar el funcionamiento completo de este tipo de expresiones regulares y todas sus opciones posibles, ya que conllevaría un desarrollo demasiado extenso y no entra en el ámbito de este documento. Sin embargo, cuando se expliquen posteriormente los patrones de búsqueda utilizados en cada caso se describirá brevemente el manejo más básico de estas expresiones. Para tener un conocimiento más amplio al respecto se recomienda la consulta de tutoriales de expresiones regulares, como por ejemplo el realizado por Kvale (Kvale, 2000).

4.4. CLASES UTILIZADAS

A continuación se explicarán las clases y ficheros PHP utilizados para la implementación del sistema de EI, así como una descripción de su funcionamiento.

En la Figura 16 pueden observarse las relaciones entre los distintos ficheros, clases y funciones que componen la aplicación.

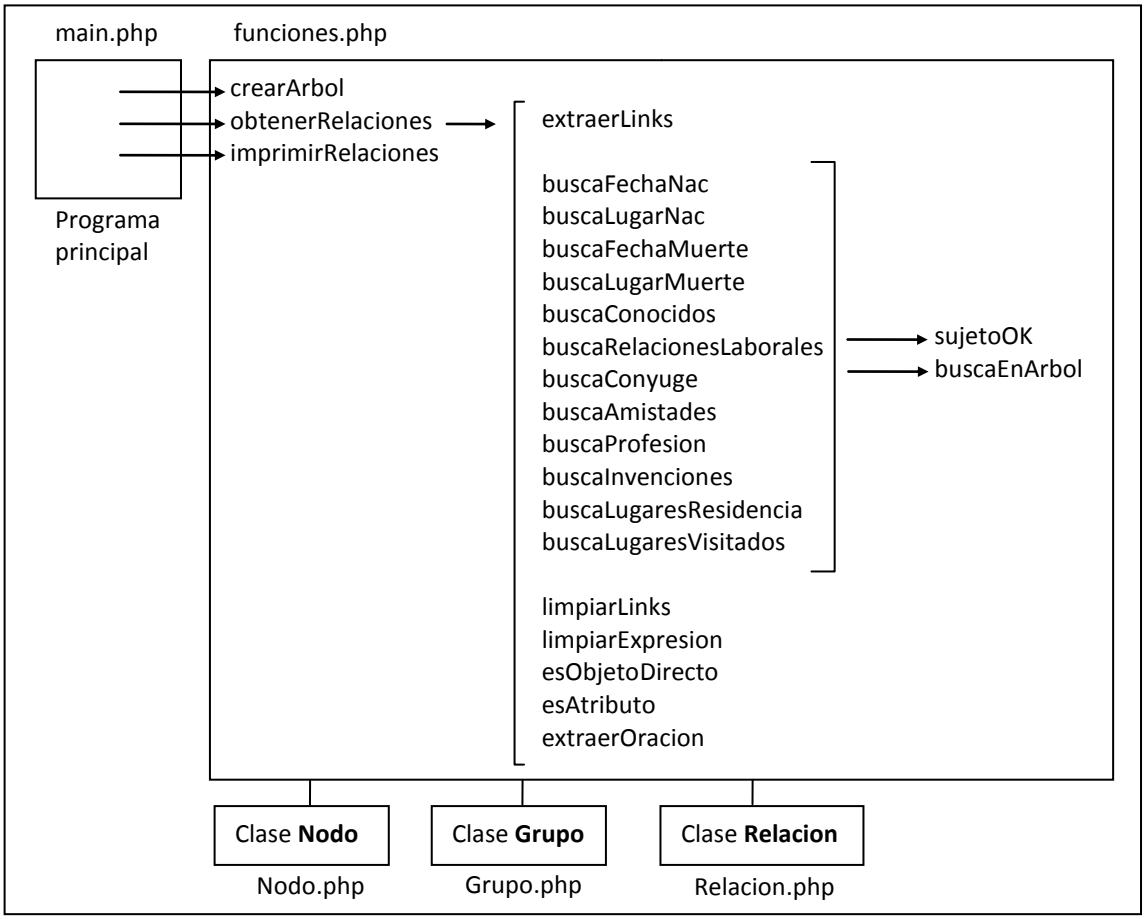


Figura 16. Esquema de ficheros, clases y funciones utilizadas

main.php

Es el fichero raíz de la aplicación. No contiene ninguna clase ni función, sino llamadas a otras funciones o programas necesarios para completar la tarea de EI. Será el único *script* PHP directamente ejecutado por el usuario. La ejecución del mismo se realizará de la siguiente forma:

```
php main.php nombre_del_artículo.txt
```

donde el único argumento de entrada será el archivo de texto que contenga el artículo del que se quiera extraer información.

Es importante recalcar que antes de la ejecución de este *script* principal, deben haberse extraído todos los artículos de Wikipedia en archivos de texto independientes, tal y como se mencionó en el apartado 3.2. Concretamente, se ha dividido en partes el fichero XML que contiene la Wikipedia completa, obteniendo dos ficheros por artículo. Ejemplo:

Miguel_de_Cervantes.txt → contiene el texto del artículo.
Miguel_de_Cervantes-links.txt → contiene los enlaces aparecidos en el artículo.

El *script* principal (main.php) procesará el texto del artículo introducido ejecutando para ello STILUS Core, con las opciones que se explicaron en el capítulo anterior (apartado 3.3.1). El resultado de este análisis será almacenado en un fichero, y pasado como argumento a la función *crearArbol*, encargada de crear una estructura en forma de árbol según el análisis sintáctico proporcionado, y otra en forma de array con todos los nodos del análisis. El funcionamiento de esta función se explicará posteriormente con mayor detalle.

Una vez creados el árbol y el array de unidades lingüísticas, se abrirá el archivo de texto que contiene el listado de links contenidos en el artículo, ya que serán necesarios para buscar relaciones con los mismos. Se llamará a la función *obtenerRelaciones* (encargada de extraer las mismas) y una vez devueltas serán finalmente mostradas por pantalla (función *imprimirRelaciones*).

Nodo.php

Este fichero contiene la clase **Nodo**. Cada objeto de esta clase representa una unidad lingüística analizada por STILUS Core.

muchas_manzanas_rojas	25	21	SEG [Diccionario general] GNFPD- |manzana|muchas_manzanas ~Sintagma nominal femenino plural objeto directo

Figura 17. Ejemplo de unidad lingüística analizada por STILUS Core

Cada línea dada por el análisis de STILUS Core sigue la estructura que puede verse en la Figura 17. Generalmente aparecen un mínimo de 4 campos separados por tabulaciones cuyo significado es el siguiente:

- Unidad lingüística analizada. Puede estar formada por una o más palabras o signos de puntuación. Las palabras aparecen separadas por el carácter '_'.
- Posición en el texto. Indica la posición de la unidad analizada tomando como referencia el inicio del texto. Está medido en caracteres.
- Tamaño en caracteres de la expresión analizada.
- Análisis sintáctico, morfológico y semántico.

La clase Nodo cuenta con cuatro atributos (*cadena*, *num1*, *num2* y *tipo*) que se utilizarán para almacenar cada uno de los campos del análisis que acaban de explicarse. Además contará con un quinto atributo (*hijo*) que almacenará siempre un objeto de la clase *Grupo* y que será utilizado en los casos en los que una unidad lingüística pueda subdividirse en varias.

Grupo.php

Este fichero contiene la clase **Grupo**. Los objetos pertenecientes a esta clase representan conjuntos de una o más unidades lingüísticas. Esta clase posee un atributo principal (*nodos*) que representa un array de elementos de la clase Nodo. Además contiene otros dos atributos (*grupoPadre* y *nodoPadre*) que se utilizarán para almacenar referencias al nodo y grupo de nivel inmediatamente superior.

La figura siguiente muestra un ejemplo de utilización de un objeto de esta clase. Como puede observarse, tanto en esta clase como en la clase Nodo, se utilizan ciertos atributos que actúan como referencias a otros objetos. Estas referencias resultarán muy útiles a la hora de crear la estructura sintáctica global del texto en forma de árbol, y sobre todo serán de vital importancia cuando sea necesario recorrer dicho árbol.

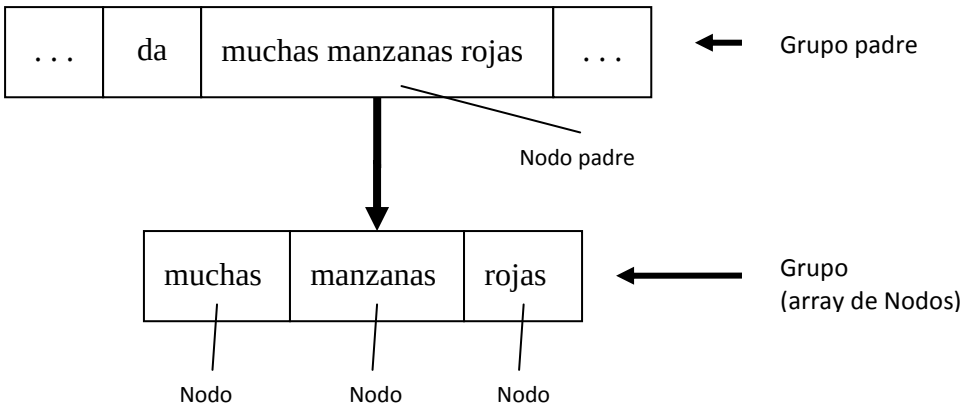


Figura 18. Ejemplo de objeto de la clase *Grupo*

Relacion.php

Este fichero contiene la clase **Relacion**. El objetivo de esta clase es contar con una estructura de datos para almacenar las relaciones que se vayan encontrando a lo largo del proceso de extracción.

Esa clase cuenta con tres atributos. El atributo *concepto* almacena el concepto o enlace con el cual se ha encontrado una relación. El atributo *relaciones* únicamente se utiliza con relaciones de tipo genérico, y guarda un listado de relaciones encontradas con el concepto de interés. Por último, el atributo *etiqueta* representa la categoría de la relación encontrada. En total habrá 13 posibles etiquetas para relaciones, según los distintos tipos de relación que se listaron en el capítulo anterior (apartado 3.5).

La siguiente figura dos ejemplos de posibles objetos de la clase Relacion para el artículo “Franz Kafka”[8]:

Concepto	Max Brod
Etiqueta	RELACIONES_LABORALES

Concepto	checo
Etiqueta	OTRAS_RELACIONES
Relaciones	era checo aprendió el idioma checo …

Figura 19. Ejemplos de objeto de la clase *Relación*

funciones.php

Este fichero no contiene ninguna clase, sin embargo es el más extenso, ya que contiene todas las funciones implementadas para llevar a cabo el proceso de EI.

Las funciones contenidas en este fichero PHP pueden dividirse en dos grupos:

- *Funciones auxiliares*: funciones que llevan a cabo tareas generalmente secundarias, pero necesarias para completar con éxito el proceso de extracción. Normalmente son llamadas por otras funciones.

- *Funciones de EI*: funciones principales de la aplicación, encargadas de recorrer y analizar el texto en busca de relaciones asignando éstas a categorías concretas.

La explicación de cada una de las funciones implementadas será llevada a cabo en los dos apartados siguientes.

[8] http://es.wikipedia.org/wiki/Franz_Kafka

4.5. FUNCIONES AUXILIARES

A continuación se describirá el funcionamiento de las funciones auxiliares desarrolladas para la implementación del sistema de EI.

Función *crearArbol*

Aunque se encuadre dentro del grupo de funciones auxiliares por no pertenecer propiamente al proceso de extracción, esta función es una de las más importantes del sistema y se ejecuta una única vez por artículo procesado.

El objetivo de esta función es construir una estructura de datos en forma de árbol a partir del análisis obtenido por STILUS Core. Este árbol estará formado por objetos pertenecientes a las clases Nodo y Grupo, que estarán conectados entre sí a base de referencias, lo que permitirá recorrer en cualquier momento el árbol creado.

Paralelamente a la creación del árbol, se construirá además un array de elementos de la clase Nodo, donde se almacenarán todas las unidades lingüísticas analizadas por STILUS Core según su orden de aparición en el análisis.

Para entender más fácilmente el funcionamiento de esta función, se ilustrará con un ejemplo. En la Figura 20 puede verse que la mayoría de las líneas del análisis constan de una unidad lingüística, dos valores numéricos y un análisis (estructura anteriormente explicada en la Figura 17). Sin embargo, también se observan líneas en las que únicamente aparece una llave de apertura (***{**) o de cierre (***}**). Una llave de apertura indica que la unidad que acaba de analizarse puede subdividirse en otras de menor tamaño. Éstas a su vez también pueden subdividirse, estableciendo una estructura en árbol. Una llave de cierre indicará que ha finalizado el análisis de las unidades contenidas en una de mayor tamaño.

```
El_árbol_que_he_visto     0     8     SEG   [Diccionario   general]   GNMSS-
|árbol|El_árbol ~Sintagma nominal masculino singular sujeto
*{
El     0     2     SEG    [Diccionario    general]    TDMSN9|el|el    ~Artículo
masculino singular
árbol  3     5     SEG    [Diccionario    general]    NCMS--N-N5|árbol|árbol
SemEntity=@inst@nofiction@NATURAL_OBJECT@LIVING_THING@FLORA@@
SemTheme=@NATURAL_SCIENCES@BOTANY ~Nombre común masculino singular
que_he_visto 9     17    SEG    [Diccionario    general]    OSA-N----|que|que
~Proposición subordinada adjetiva complemento del nombre
*{
que    9     3     SEG   [Diccionario   general]   GNUUS-|que|que   ~Sintagma
nominal sujeto
*{
que    9     3     SEG   [Diccionario   general]   RPMSUN8|que|que   ~Relativo
pronominal masculino singular    SEG  [Diccionario general]  RPMPUN8|que|que
~Relativo pronominal masculino plural  SEG       [Diccionario       general]
RPFSUN8|que|que ~Relativo pronominal femenino singular     SEG  [Diccionario
general] RPFPUN8|que|que ~Relativo pronominal femenino plural
*}
he_visto     13     8     SEG   [Diccionario   general]   GV-S--|ver|he_visto
~Sintagma verbal singular
*{
he_visto     13     8     SEG      [Diccionario      general]      VI-S1pTL-
N7|ver|he_visto  ~Verbo  léxico  transitivo  1ª  persona  singular  pretérito
perfecto indicativo
*}
*}
*}
```

Figura 20. Ejemplo de análisis de un sintagma realizado por STILUS Core

Dado que trabajar directamente con el análisis de STILUS Core resulta costoso y poco práctico, la opción más útil es contar con una estructura de datos que contenga la misma información pero organizada de manera más práctica. Tal y como se ha visto, según la salida dada por el análisis, la mejor opción será la estructura en árbol.

El funcionamiento de la función implementada es básicamente de la siguiente manera:

- Se crea un objeto de la clase Grupo que represente al árbol completo (almacenará todos los nodos de mayor entidad).

- Se crean dos referencias, que apuntarán al Nodo y al Grupo que se estén manipulando en cada momento.

- La salida de STILUS Core es leída línea a línea, y las acciones realizadas dependen del tipo de línea encontrada. Pueden darse tres situaciones:

    - Unidad analizada (expresión + número + número + análisis): se crea un nuevo objeto de la clase Nodo con los datos de la línea leída. Este Nodo es introducido en el Grupo que se esté manipulando y en el array general de elementos.
    - Apertura de llave *{: se crea un nuevo Grupo descendiente del último Nodo introducido. Las referencias son actualizadas debidamente.
    - Cierre de llave *}: se actualizan las referencias volviendo a apuntar al grupo y Nodo de nivel inmediatamente superior.

La función devolverá el Grupo raíz o de mayor entidad, que contendrá todas las unidades primarias del análisis, a partir de las cuales se podrá acceder al árbol completo simplemente moviéndose por los nodos descendientes. La siguiente figura ilustra un fragmento del árbol que se obtendría para el ejemplo anterior.

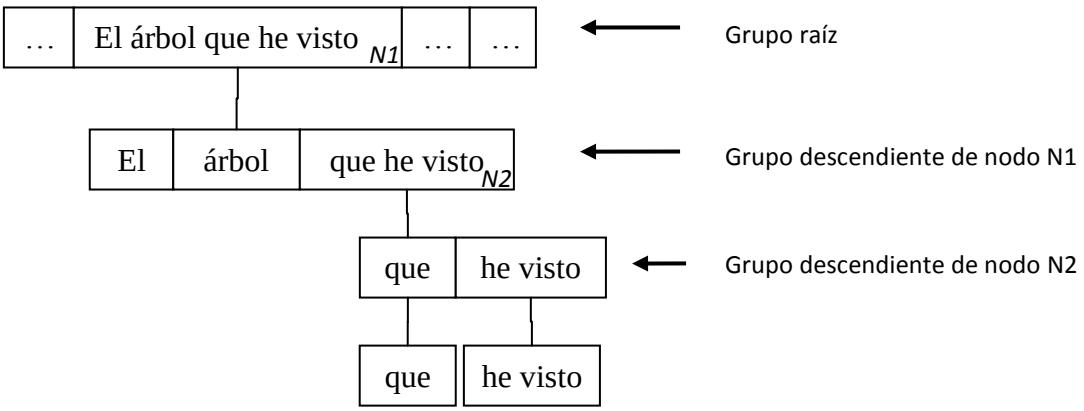


Figura 21. Función crearArbol

En esta función se pone de manifiesto por primera vez la utilidad de las expresiones regulares tipo Perl. Las líneas de texto correspondientes a unidades analizadas por STILUS Core se ajustan al siguiente patrón (en notación Perl):

```
/(\S+)\t(\d+)\t(\d+)\t(.+)/
```

cuyo significado es el siguiente:

Conjunto de uno o más caracteres distintos de espacio	TABULADOR	Conjunto de uno o más caracteres numéricos	TABULADOR	Conjunto de uno o más caracteres numéricos	TABULADOR	Conjunto de uno o más caracteres de cualquier tipo

La ventaja de utilizar este tipo de expresiones no es sólo la facilidad para encontrar cualquier tipo de patrón, sino además la posibilidad de extraer fragmentos del mismo que sean de interés. En este caso, si se combina la expresión regular con la función *preg_match* de PHP, podrán extraerse a un array de salida las zonas de la expresión que aparecen entre paréntesis (`(\S+)`,`(\d+)`,`(\d+)` y `(.+)`), que son precisamente los datos necesarios para crear un nuevo elemento de la clase Nodo.

Función *extraerLinks*

Esta función toma como entrada el fichero de texto que contiene el listado de enlaces aparecidos en un artículo, con el fin de transformarlo en array. El fichero de enlaces tendrá un formato similar al de la siguiente figura:

```
Trinidad          http://es.wikipedia.org/wiki/Trinidad
Carlos II         http://es.wikipedia.org/wiki/Carlos_II
Cambridge         http://es.wikipedia.org/wiki/Cambridge
…
Imperial College  http://www.newtonproject.ic.ac.uk/catalogue
…
```

Figura 22. Estructura del fichero de enlaces

Únicamente se introducirán en el array aquellos enlaces que lleven a otros artículos de Wikipedia, ya que es con éstos con los que se intentará extraer relaciones. No se tendrán en cuenta enlaces a sitios externos. Esta condición será fácilmente conseguible mediante el uso de la siguiente expresión regular:

```
/(.+)\t(.+)(wikipedia\.org)(.+)/
```

Se comprobará también que un elemento no exista ya en el array antes de introducirlo, ya que un mismo enlace puede aparecer varias veces dentro del artículo.

Función *buscaEnArbol*

Esta función busca un nodo concreto dentro del árbol dada su posición en el texto y su tamaño en caracteres. Esto será de mucha utilidad cuando se necesite saber dentro de qué tipo de sintagma se encuentra un concepto específico. Para localizar dicho nodo se sigue el siguiente procedimiento:

- Se recorren uno a uno los nodos de mayor entidad, comprobando siempre que la variable *pos* (posición en el texto) sea menor que la buscada.
- Si se llega a un nodo cuyo indicador de posición es mayor al buscado, se vuelve al nodo anterior y se recorren sus nodos descendientes.
- El proceso se repite hasta localizar el nodo cuya posición y tamaño concuerden con los buscados.

La función no devuelve una copia del nodo encontrado, sino una referencia al mismo, para poder continuar trabajando en el árbol original.

Función *limpiarLinks*

Esta función se encarga de recorrer el array de enlaces (creado con la función *extraerLinks*) y eliminar del mismo los enlaces a los que ya se les ha localizado alguna relación en el texto.

La función *limpiarLinks* se ejecutará una vez que ya se hayan buscado en el texto los 12 patrones de extracción definidos (listados en el apartado 3.5) y únicamente sea posible clasificar los enlaces restantes dentro de relaciones genéricas. Para realizar la operación deseada, se recorre en el array de enlaces y se comprueba elemento por elemento si éste ya se halla en el array de relaciones (array que almacena las relaciones encontradas). Si aún no se encuentra en dicho array, se introduce en un nuevo array de enlaces que finalmente se devolverá como salida.

Función imprimirRelaciones

Ésta será siempre la última función en ejecutarse dentro del proceso de extracción, ya que se encarga de mostrar por pantalla los resultados obtenidos.

Una vez finalizado el proceso de extracción, las relaciones encontradas habrán sido almacenadas en un array de elementos de la clase Relación (ver Figura 19). Para mostrarlas por pantalla, se procederá de la siguiente forma:

- Se mostrará la etiqueta de la relación encontrada (*PROFESION*, *AMISTADES*, *INVENCIONES_CREACIONES*, etc.)
- Se mostrará el listado de conceptos que se encuadren dentro de la misma etiqueta.
- Para el caso de relaciones genéricas (etiqueta *OTRAS_RELACIONES*) se mostrará además la relación específica encontrada en el texto.

Función *limpiarExpresion*

Esta función toma una cadena de entrada e inserta delante de ciertos caracteres especiales el carácter de escape '\'. Esta funcionalidad será de mucha utilidad cada vez que se esté trabajando con expresiones regulares tipo Perl. En este tipo de expresiones existen ciertos caracteres que tienen un uso reservado, por lo que para incluirlos dentro de patrones de búsqueda es necesario colocar delante el carácter de escape.

El conjunto de caracteres espaciales en los que debe incluirse el carácter de escape son los siguientes:

```
{}[]()^$.|*+?\/
```

4.6. FUNCIONES DE EXTRACCIÓN DE INFORMACIÓN

En esta sección se explicará el desarrollo de las funciones que están directamente relacionadas con el proceso de extracción y de búsqueda de patrones en el texto. Dentro de este conjunto de funciones podrán observarse dos grupos claramente diferenciados:

- Un grupo de funciones encargado de localizar patrones específicos en el texto y etiquetar las coincidencias encontradas de acuerdo a diferentes categorías.

- Otro grupo de funciones de apoyo de utilidad para el proceso de extracción.

Además de los dos grupos mencionados, se ha desarrollado una función principal (*obtenerRelaciones*) que llamará al resto de funciones indicadas para completar el proceso de extracción.

Como se verá a continuación, las funciones destinadas a buscar patrones realizan esta tarea combinando tres clases de patrones diferentes (vistos anteriormente en la sección 2.2.1):

- ***Patrones léxicos***: según el tipo de información buscada, se incluirán en los patrones diferentes palabras de búsqueda. Por ejemplo, si se está buscando el lugar de nacimiento, debería aparecer en el patrón el verbo “nacer” en alguna de sus formas.

- ***Patrones sintácticos***: en muchas ocasiones la información buscada siempre aparecerá dentro de un mismo tipo de sintagma o cumplirá una misma función en la oración (objeto directo, sintagma preposicional, complemento circunstancial, etc.).

- ***Patrones semánticos***: la información buscada en cada caso suele pertenecer a una misma categoría semántica. Por ejemplo, si intenta buscar una amistad del personaje, siempre se tratará de un nombre de persona.

Normalmente será necesario utilizar simultáneamente las tres categorías de patrones mencionadas para cada elemento de información a buscar. En esta tarea será de una enorme utilidad el análisis ofrecido por STILUS Core.

Para el caso de los patrones léxicos, la primera dificultad que aparece son las distintas formas que puede adoptar un sintagma verbal (presente, pasado, participio,…) y sus variantes en persona y número. Un patrón de búsqueda aplicado directamente al texto necesitaría tener en cuenta todas estas formas para realizar con éxito la extracción. Sin embargo, STILUS Core ofrece la ventaja de trabajar a nivel de sintagma, pudiendo acceder directamente al lema del mismo (*nacer*, *conocer*, *trabajar*,…) y facilitando en gran medida el diseño del patrón de búsqueda.

A partir del análisis de STILUS Core también pueden tratar de localizarse patrones sintácticos y semánticos. Los primeros aparecerán frecuentemente gracias al análisis sintáctico superficial realizado por la herramienta. En cuanto a los segundos, en muchos casos el análisis también proporcionará la categoría semántica de la palabra, incorporando así una ayuda extra para el proceso de EI.

4.6.1 Función principal

En este apartado se explicará la implementación de la función ***obtenerRelaciones***, función principal del proceso de EI.

El objetivo de la función será construir un array de elementos de la clase Relación (ver apartado 4.4) en el que se almacenen todas las relaciones encontradas en un artículo con otros enlaces o conceptos. Este array será devuelto como salida de la función una vez ejecutada. La función recibe como entrada tanto el fichero de enlaces como el análisis realizado por STILUS Core (ya transformado en árbol y en array).

Los pasos seguidos hasta obtener el array de relaciones mencionado son los siguientes:

- Se extraen los enlaces del fichero de texto que contiene los mismos (función *extraerLinks*, apartado 4.5).

- Se buscan en el análisis de STILUS Core los 12 patrones de extracción distintos con los que se etiquetan todas las relaciones posibles. Para este paso se ejecutan las doce funciones de extracción diseñadas para tal fin (ver apartado 4.6.2).

- En este punto ya habrá un cierto número de enlaces etiquetados en el array de relaciones. Para los enlaces para los que aún no se ha encontrado una relación etiquetada se tratará de buscar una relación genérica. Se eliminan del array de enlaces los que ya han sido etiquetados (función *limpiarLinks*, apartado 4.5).

- Para cada enlace aún no etiquetado, se buscan todas sus apariciones dentro del árbol sintáctico. Las búsquedas se realizarán recorriendo las ramas de los grupos de interés hasta llegar a los nodos "hoja" donde aparezcan estos enlaces. De este modo podrá conocerse el tipo de sintagma en el que está contenido.

- Para cada aparición de un enlace dentro del árbol, se llevarán a cabo las siguientes acciones según la parte de la oración en la que aparezca:
    - Si el sintagma en el que está contenido cumple la función de objeto directo (función *esObjetoDirecto*, apartado 4.6.3), la relación con el

artículo principal estará formada por el sintagma verbal y dicho objeto directo.

- Si el sintagma en el que está contenido cumple la función de atributo (función *esAtributo,* apartado 4.6.3), la relación con el artículo principal estará formada por el sintagma verbal y dicho atributo.
- Si el sintagma en el que está contenido cumple cualquier otra función en la oración, se tomará como relación la proposición o parte de la oración en la que aparezca (función *extraeOracion*, apartado 4.6.3)

4.6.2 Funciones de búsqueda de patrones

En este apartado se explicará el desarrollo de las funciones de búsqueda de patrones. Se trata de doce funciones, cada una de ellas destinada a localizar un tipo de patrón en el texto. Las coincidencias encontradas serán etiquetadas dentro de la categoría correspondiente.

Función *buscaFechaNac*

Esta función tratará de localizar en el texto la fecha de nacimiento del personaje del artículo.

Ejemplo:
Nikola Tesla, nacido en Smiljan, Croacia (entonces Austria-Hungría), en el seno de una familia serbia, el 10 de julio de 1856.

Para ello se siguen los siguientes pasos:

- Se recorre el conjunto de nodos que forma el artículo, comprobando en cada uno si en el análisis de STILUS Core aparece el lema '*nacer*' en tercera persona del singular o en participo (serían válidos *nació el…*, *nacido el…*, etc.). Esta comprobación se realiza mediante una simple expresión regular:

```
/\|nacer\|(.*)(3ª persona singular|participio)/
```

- Una vez que se halla un sintagma verbal de las características anteriores, se comprueba si el sujeto de la oración es el personaje de interés (función *sujetoOK*, apartado 4.6.3).

- En caso afirmativo, se busca un nodo en la misma oración cuyo análisis contenga la etiqueta 'Nombre de fecha'. Si se localiza este nodo, se comprueba si alguno de los dos nodos anteriores es un artículo o la palabra '*día*' (*nació el 3 de…*, *nació el día…*). De ser así, se crea un nuevo objeto de la clase Relación que almacene como concepto la fecha hallada y como etiqueta "*FECHA_NACIMIENTO*", y se introduce en el array de relaciones. En este caso se sale de la función, ya que únicamente existirá una fecha de nacimiento por persona.

Función *buscaLugarNac*

Esta función tratará de localizar en el texto el lugar de nacimiento del personaje del artículo.

Ejemplo:
Nació el 29 de septiembre de 1547 en Alcalá de Henares.

Para ello se siguen los siguientes pasos:

- Se recorre el conjunto de nodos que forma el artículo, comprobando en cada uno si en el análisis de STILUS Core aparece el lema '*nacer*' en tercera persona del singular o en participo.

- Una vez que se halla un sintagma verbal de las características anteriores, se comprueba si el sujeto de la oración es el personaje de interés (función *sujetoOK*, apartado 4.6.3).

- En caso afirmativo, se busca un sintagma preposicional con la preposición '*en*' posterior al sintagma verbal. Si se encuentra dicho sintagma, se comprueba si en su interior aparece un nombre de ciudad. Esta comprobación podrá realizarse verificando si en la información semántica[9] de alguno de los nodos aparece el siguiente patrón:

  ```
  /@inst@nofiction@LOCATION@GEO_POLITICAL_ENTITY@CITY/
  ```

- Si se ha localizado un nombre de ciudad, se crea un nuevo objeto de la clase Relación que almacene como concepto la ciudad hallada y como etiqueta "*LUGAR_NACIMIENTO*", y se introduce en el array de relaciones. En este caso se sale de la función, ya que únicamente existirá un lugar de nacimiento por persona.

Función *buscaFechaMuerte*

Esta función tratará de localizar en el texto la fecha de fallecimiento del personaje del artículo.

Ejemplo:
Sufriendo un cólico nefrítico moriría -tras muchas horas de delirio- la noche del 31 de marzo de 1727.

Para ello se siguen los siguientes pasos:

- Se recorre el conjunto de nodos que forma el artículo, comprobando en cada uno si en el análisis de STILUS Core aparece alguno de los siguientes lemas: '*morir*', '*fallecer*', '*expirar*', '*perecer*', en tercera persona del

[9] En sus recursos, **STILUS Sem** incluye información semántica de distinta naturaleza. De forma esquemática, se puede concretar que las entradas léxicas pueden acompañarse de los siguientes rasgos semánticos:
<tipo de entidad> <temática> <remisión> <info geográfica> <relación>

singular, infinitivo, participo o gerundio. Esta comprobación se realiza mediante la siguiente expresión regular:

```
/(\|morir\||\|fallecer\||\|expirar\||\|perecer\|)(.*)(3ª
persona singular|infinitivo|participio|gerundio)/
```

- Una vez que se halla un sintagma verbal de las características anteriores, se procede de igual forma que en la función *buscaFechaNac*, aplicando la etiqueta *FECHA_MUERTE* si aparece alguna coincidencia.

Función *buscaLugarMuerte*

Esta función tratará de localizar en el texto el lugar de fallecimiento del personaje del artículo.

Ejemplo:
Falleció en Nueva York, Estados Unidos, el 7 de enero de 1943.

Para ello se siguen los siguientes pasos:

- Se recorre el conjunto de nodos que forma el artículo, comprobando en cada uno si en el análisis de STILUS Core aparece alguno de los siguientes lemas: '*morir*', '*fallecer*', '*expirar*', '*perecer*', en tercera persona del singular o en participo, infinitivo o gerundio.

- Una vez que se halla un sintagma verbal de las características anteriores, se procede de igual forma que en la función *buscaLugarNac*, aplicando la etiqueta *LUGAR_MUERTE* si aparece alguna coincidencia.

Función *buscaConocidos*

Esta función tratará de localizar en el texto personajes conocidos del personaje principal del artículo.

Ejemplo:
A principios de agosto, Picasso partió para Mougins y se reunió con Dora Maar.

Para ello se siguen los siguientes pasos:

- Se recorre el conjunto de nodos que forma el artículo, comprobando en cada uno si en el análisis de STILUS Core aparece un lema perteneciente a alguno de los siguientes grupos:
  - '*conocer'*, '*visitar*'
  - '*reunir*', '*hablar*', '*charlar*', '*conversar*', '*entrevistar*', '*debatir*', '*discutir*', '*relacionar*'.

  Para todos los casos se aceptarán formas en tercera persona del singular, infinitivo, participio o gerundio. También se aceptarán como coincidencias los sustantivos correspondientes a las anteriores formas verbales ('*reunión*', '*charla*', '*conversación*',…).

- Una vez que se halla un sintagma verbal o sustantivo de las características anteriores, se comprueba si el sujeto de la oración es el personaje de interés (función *sujetoOK*, apartado 4.6.3).

- En caso afirmativo, se busca un sintagma preposicional con la preposición '*a*' para el primer grupo de verbos o '*con*' para el segundo grupo de verbos. El análisis de este tipo de sintagmas se adapta a los siguientes patrones:

```
/(\|a\|a|\|al\|al) ~Sintagma preposicional/
/\|con\|con ~Sintagma preposicional/
```

- Si se encuentra dicho sintagma, se comprueba si en su interior hay un sintagma nominal. Deberá verificarse si dentro de este sintagma nominal

existe algún nodo cuya información semántica indique que es un nombre de persona. Esta comprobación podrá realizarse mediante el siguiente patrón:

```
/(PERSON@FULL_NAME|PERSON@FIRST_NAME|PERSON@LAST_NAME)/
```

- Si se ha localizado un nombre de persona, se crea un nuevo objeto de la clase Relación que almacene como concepto la persona hallada y como etiqueta "*CONOCIDOS*", y se introduce en el array de relaciones. Se continuará la búsqueda en el resto de nodos del artículo hasta hallar todas las relaciones posibles de esta categoría.

Función *buscaRelacionesLaborales*

Esta función tratará de localizar en el texto personajes que mantengan alguna relación laboral con el personaje principal del artículo.

Ejemplo:
Es un habitual de la música de películas de aventuras, sobre todo a partir de su colaboración, casi desde sus inicios, con Steven Spielberg (con Encuentros en la tercera fase o Tiburón) y con George Lucas.

Para ello se siguen los siguientes pasos:

- Se recorre el conjunto de nodos que forma el artículo, comprobando en cada uno si en el análisis de STILUS Core aparece alguno de los siguientes lemas: '*trabajar*', '*colaborar*', '*asociar*', '*iniciar*'. Para todos los casos se aceptarán formas en tercera persona del singular, infinitivo, participio o gerundio. También se aceptarán como coincidencias los sustantivos correspondientes a las anteriores formas verbales ('*trabajo*', '*colaboración*', '*asociación*',…).

- Una vez que se halla un sintagma verbal de las características anteriores, se procede de igual forma que en la función *buscaConocidos*, localizando un

sintagma preposicional con la preposición ‘con’ y un nombre de persona. En cada relación encontrada se aplicará la etiqueta “*RELACIONES_LABORALES”*.

Función *buscaConyuge*

Esta función tratará de localizar en el texto personajes que hayan sido cónyuges del personaje principal del artículo.

Ejemplo:
Casó en primeras nupcias con María de Portugal (1527-1545) el 15 de noviembre de 1543.

Para ello se siguen los siguientes pasos:

- Se recorre el conjunto de nodos que forma el artículo, comprobando en cada uno si en el análisis de STILUS Core aparece alguno de los siguientes lemas: ‘*casar*’, ‘*desposar*’, en tercera persona del singular, infinitivo, participio o gerundio, o bien los sustantivos ‘*boda*’, ‘*matrimonio*’, ‘*casamiento*’, ‘*nupcias*’.

- Una vez que se halla un sintagma de las características anteriores, se procede de igual forma que en la función *buscaConocidos*, localizando un sintagma preposicional con la preposición ‘*con*’ o ‘*a*’ y un nombre de persona. En cada relación encontrada se aplicará la etiqueta “*CONYUGE”*.

Función *buscaProfesion*

Esta función tratará de localizar en el texto la profesión o profesiones del personaje principal del artículo.

Ejemplo:
Friedrich Wilhelm Nietzsche fue un filósofo, poeta y filólogo alemán, considerado uno de los pensadores modernos más influyentes del siglo XIX.

Para ello se siguen los siguientes pasos:

- Se recorre el conjunto de nodos que forma el artículo, comprobando en cada uno si en el análisis de STILUS Core aparece el lema '*ser*' en tercera persona del singular, infinitivo, participio o gerundio.

- Una vez que se halla un sintagma verbal de las características anteriores, se recorren los nodos siguientes de la oración. Se analiza en el análisis semántico de los mismos aparece la etiqueta "vocación" o "subclase de persona", mediante el siguiente patrón:

```
/(@VOCATION@|@subc@nofiction@PERSON@@@@ SemTheme)/
```

- Para cada nodo en el que se encuentre el patrón anterior se creará un nuevo objeto de la clase Relación que almacene como concepto la profesión hallada y como etiqueta "*PROFESION*", y se introduce en el array de relaciones. Se continuará la búsqueda en el resto de nodos del artículo hasta hallar todas las relaciones posibles de esta categoría.

Función *buscaAmistades*

Esta función tratará de localizar en el texto personajes que hayan sido amigos del personaje principal del artículo.

Ejemplo:
Se hospedó primero en la Pensión Alhambra, donde su amigo desde su estancia en París, Ángel Barrios, le reservó habitaciones.

Para ello se siguen los siguientes pasos:

- Se recorre el conjunto de nodos que forma el artículo, comprobando en cada uno si en el análisis de STILUS Core aparece alguno de los siguientes lemas: '*amistad*', '*amigo*', tanto en forma de sustantivo como de adjetivo.

- Una vez que se halla un sintagma de las características anteriores, se procede de igual forma que en la función *buscaConocidos*, localizando un sintagma preposicional con la preposición '*con*' o '*de*' y un nombre de persona. En cada relación encontrada se aplicará la etiqueta "*AMISTADES"*.

Función *buscaInvenciones*

Esta función tratará de localizar en el texto invenciones, creaciones u obras del personaje principal del artículo.

Ejemplos:
Es muy probable que entre los años 1581 y 1583 Cervantes escribiera La Galatea, su primera obra literaria en volumen y trascendencia.

Con dos discos metálicos separados por un conductor húmedo, pero unidos con un circuito exterior logra, por primera vez, producir corriente eléctrica continua.

Para ello se siguen los siguientes pasos:

- Se recorre el conjunto de nodos que forma el artículo, comprobando en cada uno si en el análisis de STILUS Core aparece alguno de los siguientes lemas '*inventar*', '*crear*', '*escribir*', '*pintar*', '*esculpir*', '*fabricar*', '*producir*', '*dirigir*'. Para todos los casos se aceptarán formas en tercera persona del singular, infinitivo, participio o gerundio. También se aceptarán como coincidencias los sustantivos correspondientes a las anteriores formas verbales ('*inventor*', '*creador*', etc.).

- Una vez que se halla un sintagma verbal o sustantivo de las características anteriores, se comprueba si el sujeto de la oración es el personaje de interés (función *sujetoOK*, apartado 4.6.3).

- En caso afirmativo, si el lema encontrado fue un sintagma verbal, se busca un sintagma nominal a continuación del mismo. En caso de que el lema fuese un sustantivo, se busca primero un sintagma preposicional con la preposición '*de*', y se comprueba si dentro de él aparece un sintagma nominal.

- Si se ha localizado el sintagma, se crea un nuevo objeto de la clase Relación que almacene como concepto la invención hallada y como etiqueta "*INVENCIONES_CREACIONES*", y se introduce en el array de relaciones. Se continuará la búsqueda en el resto de nodos del artículo hasta hallar todas las relaciones posibles de esta categoría

Función *buscaLugaresResidencia*

Esta función tratará de localizar en el texto diferentes lugares de residencia del personaje principal del artículo.

Ejemplo:
Entre octubre de 1919 y abril de 1929 vivió en París.

Para ello se siguen los siguientes pasos:

- Se recorre el conjunto de nodos que forma el artículo, comprobando en cada uno si en el análisis de STILUS Core aparece alguno de los siguientes lemas: '*vivir*', '*residir*', '*asentar*', '*trasladar*', '*emigrar*', '*mudar*'. Para todos los casos se aceptarán formas en tercera persona del singular, infinitivo, participio o gerundio.

- Una vez que se halla un sintagma verbal o sustantivo de las características anteriores, se comprueba si el sujeto de la oración es el personaje de interés (función *sujetoOK*, apartado 4.6.3).

- En caso afirmativo, se busca un sintagma preposicional con la preposición '*a*' o '*en*'.

- Si se encuentra dicho sintagma, se comprueba si en su interior hay un sintagma nominal. Deberá verificarse si dentro de este sintagma nominal existe algún nodo cuya información semántica indique que es un nombre de ciudad o de país. Esta comprobación podrá realizarse mediante el siguiente patrón:

```
/inst@nofiction@LOCATION@GEO_POLITICAL_ENTITY@(CITY|COUNTRY)/
```

- Si se ha localizado una localización, se crea un nuevo objeto de la clase Relación que almacene como concepto el país o ciudad hallada y como etiqueta "*LUGARES_RESIDENCIA*", y se introduce en el array de relaciones. Se continuará la búsqueda en el resto de nodos del artículo hasta hallar todas las relaciones posibles de esta categoría

Función *buscaLugaresVisitados*

Esta función tratará de localizar en el texto diferentes lugares visitados por el personaje principal del artículo.

Ejemplo:
A inicios de setiembre llegó a Stuttgart.

Para ello se siguen los siguientes pasos:

- Se recorre el conjunto de nodos que forma el artículo, comprobando en cada uno si en el análisis de STILUS Core aparece alguno de los siguientes lemas: '*viajar*', '*estar*', '*llegar*', '*visitar*'. Para todos los casos se aceptarán formas en tercera persona del singular, infinitivo, participio o gerundio.

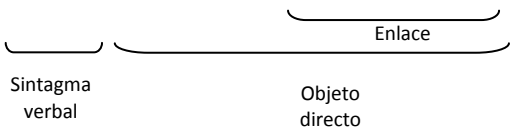
- Una vez que se halla un sintagma verbal de las características anteriores, se procede de igual forma que en la función *buscaLugaresResidencia*, localizando un sintagma preposicional con la preposición '*a*' o '*en*' y un nombre de país o ciudad. En cada relación encontrada se aplicará la etiqueta "*LUGARES_VISITADOS"*.

4.6.3 Funciones de apoyo

Función *esObjetoDirecto*

Esta función comprueba si un nodo del árbol pasado como parámetro se encuentra dentro de un objeto directo. En caso afirmativo, devolverá tanto el objeto directo encontrado como el sintagma verbal que lo precede.

Ejemplo:



Para realizar esta tarea se realizan los siguientes pasos:

- Se comprueba si en el nodo a analizar aparece la cadena "objeto directo" en el análisis de STILUS Core:

  - En caso afirmativo, se almacena el objeto directo y se localiza el sintagma verbal dentro de la oración (recorriendo los nodos que forman la misma hasta encontrarlo).
  - En caso negativo, se salta a su nodo ascendiente (de mayor tamaño) y se repite la comprobación (en el ejemplo anterior no se hallaría objeto directo en *lengua alemana* pero sí en *la cultura y la lengua alemana*).

- En caso de no encontrar objeto directo ni en el nodo analizado ni en sus ascendientes, la función devolverá *false*.

Función *esAtributo*

Esta función comprueba si un nodo del árbol pasado como parámetro se encuentra dentro de un atributo. En caso afirmativo, devolverá tanto el atributo encontrado como el sintagma verbal que lo precede.

Ejemplo:

Kennedy había sido la última persona en ser elegida ejerciendo como *senador*.

Enlace

Sintagma verbal

Atributo

El método seguido en este caso será similar al utilizado en la función anterior.

Función *extraeOracion*

Esta función se ejecutará en los casos en los que el enlace a etiquetar no se encuentre dentro de un atributo ni de un objeto directo. En estos casos se extraerá la oración o parte de la oración donde aparezca el enlace de interés.

Para realizar este proceso se llevan a cabo los siguientes pasos:

- Si el nodo pasado como argumento es descendiente de otros, se recorre el árbol en sentido ascendente hasta alcanzar el de mayor nivel.

- Se localiza el comienzo de la oración. Para ello se recorren los nodos en sentido inverso hasta llegar a un punto, punto y coma, salto de línea o comienzo de texto.

- Se va concatenando el contenido de los nodos hasta llegar al final de la oración. Para ello se realizan las mismas comprobaciones que en el paso anterior.

- Se devuelve la oración formada por los nodos concatenados.

Ejemplo:

En este ejemplo, se devolvería toda la oración como relación.

Función *sujetoOK*

Esta función será llamada frecuentemente desde funciones de búsqueda de patrones (apartado 4.6.2). Su objetivo será determinar si el sujeto de la oración que se esté analizando hace o no referencia al personaje del artículo principal.

Para comprender la utilidad de esta función, se ilustrará con el siguiente ejemplo:

- *Nació en la ciudad de Barcelona*.
- *Su hijo nació en Valencia*.

Si se observan las dos oraciones anteriores, puede verse que en ambas aparece un mismo patrón (verbo '*nacer*' + sintagma preposicional con nombre de ciudad). Si sólo se tiene en cuenta este patrón cualquiera de las ciudades podría ser etiquetada como lugar de nacimiento del personaje, pero sólo en el primer caso se estaría procediendo de manera correcta.

La función devolverá *true* si detecta que el sujeto de la misma hace referencia al personaje del artículo, y *false* en caso contrario. Para determinarlo se seguirán los siguientes pasos:

- Se recorren los nodos de la oración en busca del sujeto. Para ello se comprueba si la etiqueta de *sujeto* aparece en el análisis de STILUS Core.

- Si no se halla ningún sujeto, se entenderá que éste se encuentra omitido y hace referencia al personaje principal.

- Si se encuentra un sujeto, se analiza si éste coincide con el nombre del personaje principal (es suficiente con que coincida el apellido). En caso contrario, se concluirá que éste se corresponde con una persona diferente.

5. EVALUACIÓN DEL SISTEMA

5.1. INTRODUCCIÓN

A lo largo de este capítulo se llevará a cabo una evaluación del sistema de EI implementado. Para realizar la misma, se han seleccionado 50 artículos de Wikipedia pertenecientes a personajes de 10 categorías distintas (5 artículos por categoría). En el ANEXO A. LISTADO DE ARTÍCULOS EVALUADOS puede consultarse el listado completo de artículos utilizados. Las categorías seleccionadas han sido las siguientes:

- Personajes de actualidad
- Actores
- Escritores
- Pintores
- Físicos
- Personajes históricos
- Políticos
- Personajes de la realeza
- Compositores
- Inventores

Lógicamente la evaluación será de mayor fiabilidad si se realiza para un mayor de número de artículos, pero esto aumenta la complejidad y el tiempo destinado a la misma, debido a la necesidad de un experto humano para evaluar cada uno de los registros extraídos.

La salida del sistema será similar a la mostrada en el ejemplo de la Figura 5, aunque lógicamente los resultados no serán perfectos, existirán relaciones etiquetadas incorrectamente o no clasificadas en ninguno de los grupos de interés. En la siguiente figura se muestra un ejemplo de una salida real del sistema[10] para el artículo '*Isaac Newton*':

[10] No se muestra la salida completa debido a su gran extensión, ya que el artículo contiene un gran número de enlaces.

****
FECHA_NACIMIENTO:
25 de diciembre de 1642
****
FECHA_MUERTE:
31 de marzo de 1727
****
CONOCIDOS:
Isaac Barrow
****
RELACIONES_LABORALES:
John Locke
****
AMISTADES:
John Locke
Edmund Halley
Leibniz
****
PROFESION:
físico
filósofo
inventor
****
INVENCIONES_CREACIONES:
Cambridge
óptica
Didier
teología
****
LUGARES_RESIDENCIA:
Londres
****
LUGARES_VISITADOS:
Trinidad
Londres
****
OTRAS_RELACIONES:
LINK: Woolsthorpe
Relaciones encontradas:
Nació el 25 de diciembre de 1642 (correspondiente al 4 de enero de 1643 del nuevo calendario) en Woolsthorpe, Lincolnshire, Inglaterra; fue hijo de dos campesinos puritanos, aunque nunca llegó a conocer a su padre, pues había muerto en octubre de 1642

_____
LINK: caballero
Relaciones encontradas:
En 1705 fue nombrado caballero por la Reina Ana , como recompensa a los servicios prestados a Inglaterra
_____
LINK: Mecánica Clásica
Relaciones encontradas:
estableció las bases de la Mecánica Clásica mediante las leyes que llevan su nombre
…
…

Figura 23. Ejemplo de salida real del sistema

Analizando la salida mostrada en la figura anterior, puede verse que hay ciertos conceptos etiquetados incorrectamente o que se han quedado fuera de la categoría apropiada. Por ejemplo, no se ha etiquetado '*Woolsthorpe*' como lugar de nacimiento, posiblemente porque el módulo semántico no contiene información acerca de dicha localización. Además se etiqueta incorrectamente a '*Leibniz*' como amistad (cuando era rival de Newton) y hay conceptos que deberían haber sido etiquetados como profesión ('*alquimista*', '*matemático*') y no lo han sido. También hay errores de ambigüedad, como se ve al etiquetar incorrectamente '*Trinidad*' como localización (en el artículo se hace referencia a la "doctrina de la Trinidad"). En los siguientes apartados se analizarán con detalle los resultados por categoría para tratar de dar una explicación a estos y otros errores.

Como se ha visto en el capítulo anterior, cada una de las posibles 13 etiquetas (12 patrones más la relación '*otra*') que pueden establecerse para una relación tienen métodos diferentes para localizar y extraer la información de interés (apartado 4.6.2). Por este motivo, resultará más conveniente realizar una evaluación independiente para cada patrón de extracción buscado. De este modo, será más fácil determinar qué tipo de información es extraída con mayor eficiencia, y cuál entraña mayor dificultad.

Para llevar a cabo la evaluación, se han calculado una serie de medidas[11] para cada uno de los patrones buscados. La definición de cada una de estas medidas puede consultarse en el apartado 2.4.2.

En los siguientes apartados se detallarán y analizarán los resultados obtenidos en cada tipo de información buscada.

[11] Para el cálculo de la medida-F, se ha tomado $\beta = 1$, lo que supone valorar precisión y cobertura por igual.

5.2. RESULTADOS OBTENIDOS

5.2.1 Extracción de fechas de nacimiento

En la Tabla 2 se muestran los resultados obtenidos para la extracción de fechas de nacimiento:

Tabla 2. Medidas obtenidas en la extracción de fechas de nacimiento

Categoría	UG	OG	P	R	F1
Personajes de actualidad	0	0	0,6667	0,6667	0,6667
Actores en activo	0	0,5	0,5	1	0,6667
Escritores	0	0	1	1	1
Pintores	0	0	1	1	1
Físicos y científicos	0	0,2	0,8	1	0,8889
Personajes históricos	0	0	1	1	1
Políticos	0	0	0,75	0,75	0,75
Personajes de la realeza	0	0,25	0,75	1	0,8571
Compositores	0,25	0	1	0,75	0,8571
Inventores	0	0	1	1	1
RESULTADOS GLOBALES	0,0345	0,0968	0,8548	0,9138	0,8833

donde UG = subgeneración, OG = sobregeneración, P = precisión, R = cobertura y F1 = medida-F_1.

Como puede observarse (y como se verá también en las próximas evaluaciones), las medidas de evaluación se han calculado en primer lugar de manera independiente para cada categoría de personajes, y finalmente, se ha realizado el cálculo global sobre todos los resultados extraídos. Es importante recalcar que los resultados globales no son en ningún caso una media aritmética de los anteriores, sino un cálculo sobre el total, ya que habrá categorías en las que haya más extracciones que en otras.

En el caso que aquí ocupa, puede verse que tanto la medida de subgeneración como la de sobregeneración tienen valores razonablemente bajos (0,034 y 0,097), lo que indica que sólo un pequeño porcentaje de registros quedan sin ser extraídos y el número de respuestas espurias no es elevado.

Tanto la precisión como la cobertura alcanzan valores superiores a 0,85, lo que indica un alto porcentaje de respuestas correctas y de registros extraídos del total. Al ser ambos valores altos la medida-F_1 también alcanza un valor elevado, lo que indica que las fechas de nacimiento son extraídas con bastante fiabilidad.

- Ejemplo de respuesta incorrecta:
  Actualmente está comprometido con Helena Bonham Carter, y tienen dos hijos: un varón llamado Billy-Ray Burton y una niña, llamada Nell, nacida el 15 de diciembre de 2007.

  Para este artículo (Tim Burton[12]) se extrajo incorrectamente la fecha de nacimiento, ya que al no identificar correctamente el sujeto de la oración se interpretó que se estaba hablando del personaje principal.

- Ejemplo de respuesta perdida:
  Manuel María de los Dolores Falla y Matheu nació el 23 de noviembre de 1876.

  En este artículo (Manuel de Falla[13]) el sistema interpretó que el sujeto de la oración no hacía referencia al personaje del artículo por lo que no etiquetó la fecha aparecida como fecha de nacimiento. El problema en esta ocasión es que se muestra el nombre completo del personaje, que no coincide con el título del artículo. Sería necesario mejorar el algoritmo de identificación de sujetos para evitar estas situaciones.

[12] http://es.wikipedia.org/wiki/Tim_Burton
[13] http://es.wikipedia.org/wiki/Manuel_de_falla

5.2.2 Extracción de lugares de nacimiento

Tabla 3. Medidas obtenidas en la extracción de lugares de nacimiento

Categoría	UG	OG	P	R	F1
Personajes de actualidad	0	0	1	1	1
Actores en activo	0,25	0	1	0,75	0,8571
Escritores	0	0	1	1	1
Pintores	0	0	1	1	1
Físicos y científicos	0,25	0	1	0,75	0,8571
Personajes históricos	0	0	0,5	0,5	0,5
Políticos	0,5	0	1	0,5	0,6667
Personajes de la realeza	0	0	0,5	0,5	0,5
Compositores	0,2	0	0,875	0,7	0,7778
Inventores	0,6667	0	1	0,3333	0,5
RESULTADOS GLOBALES	0,1818	0	0,8889	0,7273	0,8

En esta ocasión se ha obtenido un valor de subgeneración de 0,181. Aunque no es un valor elevado, indica que en ciertas ocasiones hay registros válidos que no son extraídos. El valor de sobregeneración, sin embargo, es 0, por lo que en este caso el sistema no ha extraído ninguna respuesta cuando ésta no existía.

El valor de precisión en este caso es alto (0,89) por lo que las respuestas dadas son correctas en su mayoría. La cobertura tiene un valor algo menor (0,727) ya que, al igual que la subgeneración, se ve afectada por los registros no extraídos.

La mayoría de los casos en los que ha habido respuestas no extraídas se han debido a que el análisis semántico de STILUS Core no ha identificado las ciudades mencionadas como localizaciones. Aunque la base de datos semántica de STILUS Core es muy grande, lógicamente no contiene la totalidad de localizaciones geográficas existentes, aunque sí un gran número de ellas.

- Ejemplo de respuesta incorrecta:
  Nacida en Szczecin el 2 de mayo de 1729 y fallecida en San Petersburgo.

En este artículo (Catalina II de Rusia[14]) no se identificó '*Szczecin*' como localización geográfica, por lo que se etiquetó erróneamente '*San Petersburgo*' como lugar de nacimiento (cuando en realidad es el de fallecimiento).

- Ejemplo de respuesta perdida:
  Ewan McGregor nació en Crieff, Escocia (Reino Unido) en 1971.

  En este caso tampoco se identificó '*Crieff*' como localización, por lo que no fue etiquetado como lugar de nacimiento.

5.2.3 Extracción de fechas de muerte

Tabla 4. Medidas obtenidas en la extracción de fechas de muerte

Categoría	UG	OG	P	R	F1
Personajes de actualidad	-	-	-	-	-
Actores en activo	-	-	-	-	-
Escritores	0,25	0	1	0,75	0,8571
Pintores	0,2	0	1	0,8	0,8889
Físicos y científicos	0	0	0,8	0,8	0,8
Personajes históricos	0,5	0	0,75	0,375	0,5
Políticos	0	0,25	0,5	0,5	0,5
Personajes de la realeza	0	0	0,6	0,6	0,6
Compositores	0,2	0	0,625	0,5	0,5556
Inventores	0	0	1	1	1
RESULTADOS GLOBALES	0,1429	0,0333	0,7667	0,6571	0,7077

NOTA: Como puede observarse, no hay medidas calculadas para las dos primeras categorías de personajes. La razón es que en ambos casos se han elegido artículos de personajes aún no fallecidos.

[14] http://es.wikipedia.org/wiki/Catalina_II_de_Rusia

En la extracción de fechas de muerte se ha obtenido un valor de subgeneración de 0,143, por lo que al igual que en el caso anterior, un pequeño porcentaje de registros se han quedado sin extraer. La sobregeneración por el contrario es muy baja, por lo que apenas hay respuestas espurias.

La precisión y cobertura son menores que las obtenidas para la extracción de fechas de nacimiento. Esta diferencia es fundamentalmente debida a que existe un mayor número de formas de expresar la muerte de una persona, además de un gran número de causas de la misma (enfermedad, asesinato, suicidio, etc.) lo que conlleva una extracción de mayor dificultad. Para resolver este problema habría que tratar de ampliar los patrones de búsqueda utilizados.

- Ejemplo de respuesta incorrecta:
  El 28 de febrero, Cosme II, el protector de Galileo, muere súbitamente.

  Para este artículo (Galileo Galilei[15]) se ha extraído erróneamente la fecha de muerte, al no identificar correctamente el sujeto de la oración anterior.

- Ejemplo de respuesta perdida:
  El Presidente Kennedy fue asesinado el 22 de noviembre de 1963 en Dallas.

  En este caso no se etiquetó la fecha señalada como fecha de muerte, ya que la función implementada busca los lemas '*morir*', '*fallecer*', '*expirar*', etc., pero no '*asesinar*'.

[15] http://es.wikipedia.org/wiki/Galileo

5.2.4 Extracción de lugares de muerte

Tabla 5. Medidas obtenidas en la extracción de lugares de muerte

Categoría	UG	OG	P	R	F1
Personajes de actualidad	-	-	-	-	-
Actores en activo	-	-	-	-	-
Escritores	0	0	1	1	1
Pintores	0,5	0	1	0,5	0,6667
Físicos y científicos	0,75	0	1	0,25	0,4
Personajes históricos	0,5	0	1	0,5	0,6667
Políticos	1	1	0	0	0
Personajes de la realeza	0,3333	0,3333	0,3333	0,3333	0,3333
Compositores	0,6667	0	0	0	0
Inventores	0,5	0	1	0,5	0,6667
RESULTADOS GLOBALES	0,5238	0,1667	0,6667	0,381	0,4848

NOTA: Por el mismo motivo que en el apartado anterior, no hay medidas calculadas para las dos primeras categorías de personajes.

En esta ocasión se observa un porcentaje considerable de registros no extraídos, lo que implica una alta subgeneración y una baja cobertura. En este tipo de extracción, además de la dificultad de la identificación de entidades geográficas, se añaden los inconvenientes comentados en el apartado 5.2.3 (diferentes formas de expresar un fallecimiento), lo que hace que entrañe una mayor dificultad.

Se aprecia también un pequeño porcentaje de respuestas espurias (sobregeneración de 0,17). La precisión en este caso tiene un valor bastante superior al de la cobertura (0,67 frente a 0,38) por lo que de las respuestas dadas una mayor parte son correctas.

- Ejemplo de respuesta incorrecta:
  El primero en llegar a Venezuela fue José Palacios Sojo y Ortiz de Zárate, natural de Miranda de Ebro en 1647, que falleció en Caracas en 1703.

  En este artículo (Simón Bolívar[16]), se ha etiquetado incorrectamente el lugar de fallecimiento, debido al identificación errónea del sujeto de la oración anterior.

- Ejemplo de respuesta perdida:
  Empeoró seriamente su salud y murió, con apenas cuarenta años, en Boulogne-Sur-Seine el 11 de mayo de 1927.

  En este caso (Juan Gris[17]) no se ha etiquetado '*Boulogne-Sur-Seine*' como lugar de muerte al no ser reconocida por el módulo semántico como localización.

5.2.5 Extracción de personas conocidas

Tabla 6. Medidas obtenidas en la extracción de personas conocidas

Categoría	UG	OG	P	R	F1
Personajes de actualidad	0,3333	0,5	0,5	0,6667	0,5714
Actores en activo	0,5	0	1	0,5	0,6667
Escritores	0,6667	0	1	0,3333	0,5
Pintores	0,3333	0,0909	0,9091	0,6667	0,7692
Físicos y científicos	0,3333	0,2	0,8	0,6667	0,7273
Personajes históricos	0,5	0,2	0,8	0,5	0,6154
Políticos	0,5	0,5	0,5	0,5	0,5
Personajes de la realeza	0,5	0	1	0,5	0,6667
Compositores	0,4375	0	1	0,5625	0,72
Inventores	1	0	0	0	0
RESULTADOS GLOBALES	0,4355	0,1463	0,8537	0,5645	0,6796

NOTA: dentro de la última categoría de personajes (inventores) el sistema no fue capaz de dar ninguna respuesta.

[16] http://es.wikipedia.org/wiki/Simon_Bolivar
[17] http://es.wikipedia.org/wiki/Juan_Gris

En la tarea de extracción de personas conocidas, se observa un alto porcentaje de registros perdidos (subgeneración de 0,43), lo que influye negativamente en la cobertura obtenida (0,56). La sobregeneración obtenida una vez más es inferior a la subgeneración. En cuanto a la precisión obtenida, se ha conseguido un valor bastante superior al de la cobertura, por lo que el porcentaje de respuestas correctas es relativamente alto.

En esta ocasión se han detectado diversas dificultades que han impedido unos mejores valores de precisión y cobertura. En primer lugar hay que tener en cuenta que el número de expresiones posibles para nombrar a personas conocidas es bastante amplio, lo que hace difícil abarcar todas las situaciones posibles. Además, la funcionalidad de STILUS Core para detectar nombres de persona no es perfecta en el 100% de los casos. En ocasiones aparecen en los artículos nombres o apellidos poco comunes que no son considerados como tales por la herramienta.

- Ejemplo de respuesta espuria:
  Diplomática y militarmente su reinado se caracterizó por el éxito contra el Imperio Parto y una mejora de las relaciones con Grecia.

  En este caso (Nerón[18]) se ha etiquetado incorrectamente '*Grecia*', ya que ha sido incorrectamente identificada como nombre de persona.

- Ejemplo de respuesta perdida:
  Catalina mantiene relaciones con Sergéi Saltykov y Estanislao II Poniatowski.

  En este ejemplo (Catalina II de Rusia) existen dos relaciones que no son etiquetadas como personas conocidas, ya que el módulo semántico no los ha identificado como nombres de persona.

[18] http://es.wikipedia.org/wiki/Neron

5.2.6 Extracción de relaciones laborales

Tabla 7. Medidas obtenidas en la extracción de relaciones laborales

Categoría	UG	OG	P	R	F1
Personajes de actualidad	0,4545	0	1	0,5455	0,7059
Actores en activo	0,375	0	1	0,625	0,7692
Escritores	0	0	1	1	1
Pintores	0	0	1	1	1
Físicos y científicos	0,25	0,25	0,75	0,75	0,75
Personajes históricos	-	-	-	-	-
Políticos	0	0,3333	0,6667	1	0,8
Personajes de la realeza	0	1	0	1	0
Compositores	0,3333	0,3333	0,6667	0,6667	0,6667
Inventores	-	-	-	-	-
RESULTADOS GLOBALES	0,3226	0,16	0,84	0,6774	0,75

NOTA: En ciertas categorías de personaje no ha aparecido ninguna clave, por lo que ciertas medidas no han podido calcularse.

En esta ocasión también se ha obtenido una subgeneración elevada, aunque algo inferior a la obtenida para el caso anterior. La sobregeneración se mantiene en los mismos niveles conseguidos para la extracción de personas conocidas. Tanto la cobertura como la precisión alcanzan valores algo más altos que en el último caso, siendo nuevamente la precisión el mejor resultado obtenido (0,84).

Las dificultades de extraer este tipo de información son similares a las observadas en la extracción de personas conocidas: amplio número de patrones posibles y fallos en la detección de ciertos nombres de persona. Aun así se han conseguido resultados algo mejores que en este último caso.

- Ejemplo de respuesta espuria:
  Estos temas recibieron recientemente una especial atención con el reestreno de su más exitoso trabajo operístico, Pepita Jiménez, interpretaciones de conciertos y la grabación de Merlín, con Plácido Domingo.

  En este artículo (Isaac Albéniz[19]) se etiqueta erróneamente '*Plácido Domino*' como relación laboral. En la oración se menciona que uno de los trabajos de Albéniz fue grabado con Plácido Domingo, pero ambos personajes no tienen realmente una relación laboral.

- Ejemplo de respuesta perdida:
  Colaboró en los nuevos álbumes de Ringo Starr.

  En este ejemplo (Alanis Morissette[20]) aparece una relación laboral no extraída, al no identificar el sistema '*Ringo Starr*' como nombre de persona.

5.2.7 Extracción de cónyuges

Tabla 8. Medidas obtenidas en la extracción de cónyuges

Categoría	UG	OG	P	R	F1
Personajes de actualidad	1	0	0	0	0
Actores en activo	0,25	0,25	0,75	0,75	0,75
Escritores	-	-	-	-	-
Pintores	1	0	0	0	0
Físicos y científicos	1	0	0	0	0
Personajes históricos	0,5	0	1	0,5	0,6667
Políticos	0,5	0	1	0,5	0,6667
Personajes de la realeza	0,2857	0,5	0,5	0,7143	0,5882
Compositores	0	0,6667	0,1667	0,5	0,25
Inventores	-	-	-	-	-
RESULTADOS GLOBALES	0,5	0,4	0,575	0,4792	0,5227

NOTA: En algunas categorías de personaje no ha aparecido ninguna clave, por lo que ciertas medidas no han podido calcularse.

[19] http://es.wikipedia.org/wiki/Isaac_Albeniz
[20] http://es.wikipedia.org/wiki/Alanis_morissette

En este tipo de extracción se ha obtenido un porcentaje relativamente alto tanto de respuestas espurias como de registros no extraídos, lo que ha inducido altos valores de subgeneración y sobregeneración. La cobertura y precisión no han alcanzado valores especialmente altos.

El alto número de respuestas espurias es debido fundamentalmente a que en los artículos se mencionan en ocasiones matrimonios distintos a los contraídos por el personaje principal, que son erróneamente extraídos. Aunque el sistema trata de descartar estos "falsos positivos" comprobando el sujeto de la oración, éste no siempre es identificado correctamente por STILUS Core. En cuanto a los registros perdidos, son en parte provocados por nombres de persona no identificados, al igual que en casos anteriores.

- Ejemplo de respuesta espuria:
  Permaneció en una casona-palacio de Tordesillas con la única compañía de su última hija, Catalina (hasta que salió ésta para casarse con Juan III de Portugal).

  En este artículo (Juana I de Castilla[21]) se etiqueta erróneamente '*Juan III de Portugal*' como cónyuge, ya que el sistema no ha identificado que el sujeto de la última oración hace referencia a la hija de Juana.

- Ejemplo de respuesta perdida:
  El 2 de junio de 1919 se casó con una prima suya, Elsa Loewenthal.

  En este caso (Albert Einstein[22]) el módulo semántico no identificó el apellido '*Loewenthal*' como tal, por lo que la relación no fue etiquetada como cónyuge.

[21] http://es.wikipedia.org/wiki/Juana_I_de_Castilla
[22] http://es.wikipedia.org/wiki/Albert_Einstein

5.2.8 Extracción de amistades

Tabla 9. Medidas obtenidas en la extracción de amistades

Categoría	UG	OG	P	R	F1
Personajes de actualidad	0,375	0,5	0,6667	0,5	0,5714
Actores en activo	0,75	0,75	0,25	0,25	0,25
Escritores	0	0	1,3333	1	1,1429
Pintores	0,0625	0,125	0,875	0,875	0,875
Físicos y científicos	0,3333	0,2	0,8	0,6667	0,7273
Personajes históricos	0,4	0,1429	0,8571	0,6	0,7059
Políticos	1	1	0	0	0
Personajes de la realeza	0	0,2	0,8	1	0,8889
Compositores	0,45	0	1	0,55	0,7097
Inventores	0	0	1	1	1
RESULTADOS GLOBALES	0,3289	0,2167	0,8167	0,6447	0,7206

Al igual que en otras ocasiones, se ha obtenido un valor de precisión superior al de cobertura, ya que hay un cierto porcentaje de registros correctos que no son extraídos por el sistema (subgeneración de 0.32).

Nuevamente vuelve a influir en el valor de cobertura la potencia del módulo de análisis semántico para identificar nombres y apellidos de persona.

- Ejemplo de respuesta espuria:
  Su mejor amigo en la Upper Sixth era el propietario de un Ford Anglia color turquesa.

  En este artículo (J.K. Rowling[23]) se ha etiquetado erróneamente como amistad la relación con el artículo '*Ford Anglia*'. El sistema lo ha reconocido como nombre de persona (a pesar de ser un modelo de coche) contenido en un sintagma preposicional tras el lema '*amigo*'.

[23] http://es.wikipedia.org/wiki/JK_Rowling

- Ejemplo de respuesta perdida:

  El romance más famoso de Keaton fue con el director Woody Allen el cual duró la mayor parte de los años 70. A Allen lo describe como uno de sus amigos más cercanos.

  En este ejemplo (Diane Keaton[24]) no ha sido etiquetada la relación con '*Woody Allen*' como amistad. Aunque aparece el lema '*amigo*' en la segunda oración, sólo se nombra su apellido, por lo que el sistema no es capaz de discernir que está haciendo referencia a '*Woody Allen*'.

5.2.9 Extracción de profesiones

Tabla 10. Medidas obtenidas en la extracción de profesiones

Categoría	UG	OG	P	R	F1
Personajes de actualidad	0,25	0	1	0,75	0,8571
Actores en activo	0	0	1	1	1
Escritores	0,1429	0	1	0,8571	0,9231
Pintores	0	0,1667	0,8333	0,8333	0,8333
Físicos y científicos	0,2	0	1	0,8	0,8889
Personajes históricos	0,25	0	1	0,75	0,8571
Políticos	0,125	0	1	0,875	0,9333
Personajes de la realeza	1	0	0	0	0
Compositores	0,3333	0	1	0,6667	0,8
Inventores	0,2222	0	1	0,7778	0,875
RESULTADOS GLOBALES	0,2329	0,0179	0,9821	0,7534	0,8527

NOTA: Para la categoría "Personajes de la realeza" el sistema no ha proporcionado respuestas, por lo que algunas medidas no han sido calculadas.

[24] http://es.wikipedia.org/wiki/Diane_Keaton

En la extracción de profesiones se observa que en un cierto porcentaje de casos existen registros que no son extraídos (subgeneración de 0,23). Por el contrario, el valor de la sobregeneración es bastante bajo, por lo que los falsos positivos son poco frecuentes.

La precisión conseguida tiene un valor muy alto (0,98) por lo que en la mayoría de casos las respuestas obtenidas son correctas. La cobertura tiene un valor algo menor (0,75) ya que como se ha comentado antes hay un cierto porcentaje de registros no extraídos. Este porcentaje es debido mayormente al análisis semántico de STILUS Core, que en algunas ocasiones no identifica correctamente una profesión dentro de la categoría correcta (vocación, clase de persona, etc.).

- Ejemplo de respuesta espuria:
  Su padre, Salvador Dalí y Cusí, era notario de la ciudad.

  Para este artículo (Salvador Dalí[25]) se ha etiquetado como profesión el concepto '*notario*' que en realidad hace referencia al padre del personaje principal. En este caso ha habido una identificación errónea del sujeto.

- Ejemplo de respuestas perdidas:
  Sir Isaac Newton fue un científico, físico, filósofo, inventor, alquimista y matemático inglés.

  En este ejemplo podrían haberse etiquetado 6 conceptos como profesiones válidas, pero sólo se han etiquetado 3 de ellos. El módulo de análisis semántico ha considerado que los restantes no estaban dentro de la categoría semántica apropiada.

[25] http://es.wikipedia.org/wiki/Salvador_Dali

5.2.10 Extracción de invenciones

Tabla 11. Medidas obtenidas en la extracción de invenciones

Categoría	UG	OG	P	R	F1
Personajes de actualidad	0	0,2857	0,7143	1	0,8333
Actores en activo	0	0,625	0,375	1	0,5455
Escritores	0	0,4167	0,5833	1,4	0,8235
Pintores	0	0,5	0,4	0,8	0,5333
Físicos y científicos	0	0,4286	0,5714	1	0,7273
Personajes históricos	0	0,9167	0,0833	1	0,1538
Políticos	0	0,8889	0,0556	0,0417	0,0476
Personajes de la realeza	0	1	0	1	0
Compositores	0	0,6667	0,3333	1	0,5
Inventores	0,5455	0,2857	0,7143	0,4545	0,5556
RESULTADOS GLOBALES	0,1071	0,5859	0,401	0,7054	0,5097

NOTA: En la categoría 'Actores' no ha aparecido ninguna clave, por lo que ciertas medidas no han podido calcularse.

En este tipo de extracción, sin duda lo que más llama la atención es el alto número de respuestas espurias, lo que provoca unos valores de sobregeneración y precisión peores a los deseados. El porcentaje de registros válidos que quedan sin extraer es sin embargo pequeño, lo que hace que tanto la subgeneración como la cobertura alcancen valores bastante más aceptables.

Esta categoría ha sido en la que peores valores de precisión se han conseguido, lo que indica la dificultad de la misma. Tras estudiar las salidas del sistema, se ha observado que en ocasiones el análisis de STILUS Core identifica algunas formas del verbo '*creer*' como formas del verbo '*crear*', lo que provoca algunos falsos positivos. También hay problemas de ambigüedad con otras formas verbales.

- Ejemplo de respuesta espuria:
  El mismo día en que iba a llevarse a cabo dicho aviso se produjo la misteriosa explosión en esta zona de Rusia.

En este ejemplo (Nikola Tesla[26]), al aparecer el lema '*producir*' seguido de un sintagma nominal, se etiqueta erróneamente el concepto '*Rusia*' como creación.

- Ejemplo de respuesta perdida:
  El Tribunal Supremo de los Estados Unidos dictaminó que la patente relativa a la radio era legítimamente propiedad de Tesla, reconociéndolo de forma legal como inventor de ésta.

  En este caso (Nikola Tesla) no se ha etiquetado como invención la relación con el concepto '*radio*'. Al final de la oración aparece el lema '*inventor*' seguido de un sintagma preposicional. Sin embargo, este sintagma contiene el pronombre '*ésta*', que hace referencia a '*radio*', aunque el sistema no es capaz de relacionar el pronombre con el sustantivo.

5.2.11 Extracción de lugares de residencia

Tabla 12. Medidas obtenidas en la extracción de lugares de residencia

Categoría	UG	OG	P	R	F1
Personajes de actualidad	0,5	0	1	0,5	0,6667
Actores en activo	0,5	0	1	0,5	0,6667
Escritores	0	0,3333	0,6667	0,8571	0,75
Pintores	0,4286	0	1	0,5714	0,7273
Físicos y científicos	0,2	0,2	0,8	0,8	0,8
Personajes históricos	0,3333	0,3333	0,6667	0,6667	0,6667
Políticos	0,3	0	1	0,7	0,8235
Personajes de la realeza	0,3333	0	1	0,6667	0,8
Compositores	0,4	0,1429	0,8571	0,6	0,7059
Inventores	0,3333	0	1	0,6667	0,8
RESULTADOS GLOBALES	0,3226	0,1277	0,8723	0,6613	0,7523

[26] http://es.wikipedia.org/wiki/Nikola_Tesla

En esta ocasión se observa una cierta cantidad de registros no extraídos (subgeneración de 0,32) por lo que la cobertura tampoco alcanza un valor excesivamente alto. La sobregeneración es menor en este caso (0,13). Puede verse también que el valor de precisión obtenido es bastante superior al de la cobertura.

En este tipo de extracción juega un papel importante el análisis semántico de STILUS Core. En ocasiones hay registros no extraídos porque el análisis no los identifica como localizaciones. Esto ocurre generalmente con ciudades poco conocidas.

- Ejemplo de respuesta espuria:
  El 7 de octubre de 1571 participó en la batalla de Lepanto, donde participaba uno de los más famosos marinos de la época, el Marqués de Santa Cruz, que residía en La Mancha, en Viso del Marqués.

  En este caso (Miguel de Cervantes[27]) se ha etiquetado '*Viso del Marqués*' como lugar de residencia. Sin embargo, en la oración no se está haciendo referencia al personaje del artículo (error de detección de sujeto).

- Ejemplo de respuesta perdida:
  Meucci y su esposa emigraron a los Estados Unidos, y llegaron a Clifton, donde Meucci vivió el resto de su vida.

  En este ejemplo (Antonio Meucci[28]) no se ha etiquetado '*Clifton*' como lugar de residencia, ya que el módulo de análisis semántico no lo identificó como localización.

[27] http://es.wikipedia.org/wiki/Cervantes
[28] http://es.wikipedia.org/wiki/Antonio_Meucci

5.2.12 Extracción de lugares visitados

Tabla 13. Medidas obtenidas en la extracción de lugares visitados

Categoría	UG	OG	P	R	F1
Personajes de actualidad	0,2	0	1	0,8	0,8889
Actores en activo	0,5	0	1	0,5	0,6667
Escritores	0,2	0,1111	0,8889	0,8	0,8421
Pintores	0	0,0526	0,9474	0,9474	0,9474
Físicos y científicos	0	0,5	0,5	1	0,6667
Personajes históricos	0,3333	0,3636	0,6364	0,6667	0,6512
Políticos	0,1053	0,1707	0,8293	0,8947	0,8608
Personajes de la realeza	0,4444	0,1667	0,8333	0,5556	0,6667
Compositores	0,3333	0,0588	0,9412	0,6667	0,7805
Inventores	0	0	1	1	1
RESULTADOS GLOBALES	0,2	0,1769	0,8231	0,7926	0,8075

En este caso se ha obtenido una menor subgeneración y una mayor cobertura (0,79) que en el caso anterior. La precisión también ha alcanzado un valor aceptable, aunque algo inferior al del último análisis.

Se puede observar que un porcentaje de registros se ha sido extraído sin ser registros válidos (sobregeneración de 0,2). También influye en este resultado el análisis semántico, ya que existen casos en los que ciertos sintagmas son identificados como localizaciones sin serlo.

- Ejemplo de respuesta espuria:
  Colón empezó a idear su plan de llegar a Cipango (el moderno Japón).

  Para este artículo (Cristóbal Colón[29]) se etiquetó erróneamente '*Japón*' como lugar visitado, cuando el personaje principal sólo planeaba llegar a dicho destino.

[29] http://es.wikipedia.org/wiki/Cristobal_Colon

- Ejemplo de respuesta perdida:
  Entre los primeros países en los que fue recibido se encuentran Alemania, Reino Unido y Francia.

  En este ejemplo (Alfonso XIII[30]) aparecen tres localizaciones que podrían haber sido etiquetadas como lugares visitados, pero no lo fueron. El motivo es que el lema '*recibir*' no está incluido en la búsqueda para este tipo de patrón.

5.2.13 Extracción de relaciones genéricas

Tabla 14. Medidas obtenidas en la extracción de relaciones genéricas

Categoría	UG	OG	P	R	F1
Personajes de actualidad	0	0	0,845	0,845	0,845
Actores en activo	0	0	0,6842	0,6842	0,6842
Escritores	0	0	0,6657	0,6657	0,6657
Pintores	0	0	0,7597	0,7597	0,7597
Físicos y científicos	0	0	0,8828	0,8828	0,8828
Personajes históricos	0	0	0,9262	0,9262	0,9262
Políticos	0	0	0,8698	0,8698	0,8698
Personajes de la realeza	0	0	0,9083	0,9083	0,9083
Compositores	0	0	0,8847	0,8847	0,8847
Inventores	0	0	0,8704	0,8704	0,8704
RESULTADOS GLOBALES	0	0	0,844	0,844	0,844

En este último caso se evalúan las relaciones encontradas que no han podido ser clasificadas dentro de ninguno de los doce grupos anteriores. En esta situación, para todo enlace encontrado el sistema dará una respuesta (correcta o no) por lo que no habrá ni subgeneración (respuestas perdidas) ni sobregeneración (respuestas de más).

[30] http://es.wikipedia.org/wiki/Alfonso_XIII

Al haber el mismo número de respuestas que de claves los valores de precisión, cobertura y medida-F_1 coinciden. El valor alcanzado ha sido de 0,84. En los casos en los que ha habido respuestas incorrectas, éstas se han debido muchas veces a enlaces aparecidos dentro de listados, por lo que el sistema no ha sido capaz de relacionarlos con el personaje principal a través del texto.

- Ejemplos de respuestas incorrectas:

  LINK: Tierra
  Relaciones encontradas: invadiría la Tierra
  Esta relación, extraída del artículo 'Tim Burton' lógicamente no relaciona el artículo '*Tierra*' con el personaje del artículo, sino que está haciendo referencia a una película dirigida por él.

  LINK: pintura
  Relaciones encontradas: eran la pintura
  En este caso se ha identificado '*la pintura*' como atributo, pero no hace referencia al personaje del artículo.

  LINK: Breve Historia del Tiempo
  Relaciones encontradas: Breve Historia del Tiempo
  En este caso el sistema sólo ha podido extraer el propio concepto como relación consigo mismo. Este tipo de respuestas suelen ocurrir cuando del texto aparecen listados con enlaces (por ejemplo, listados de obras).

5.3. RESUMEN DE RESULTADOS

La Figura 24 muestra un resumen de los resultados obtenidos de precisión y cobertura para los distintos tipos de información buscada.

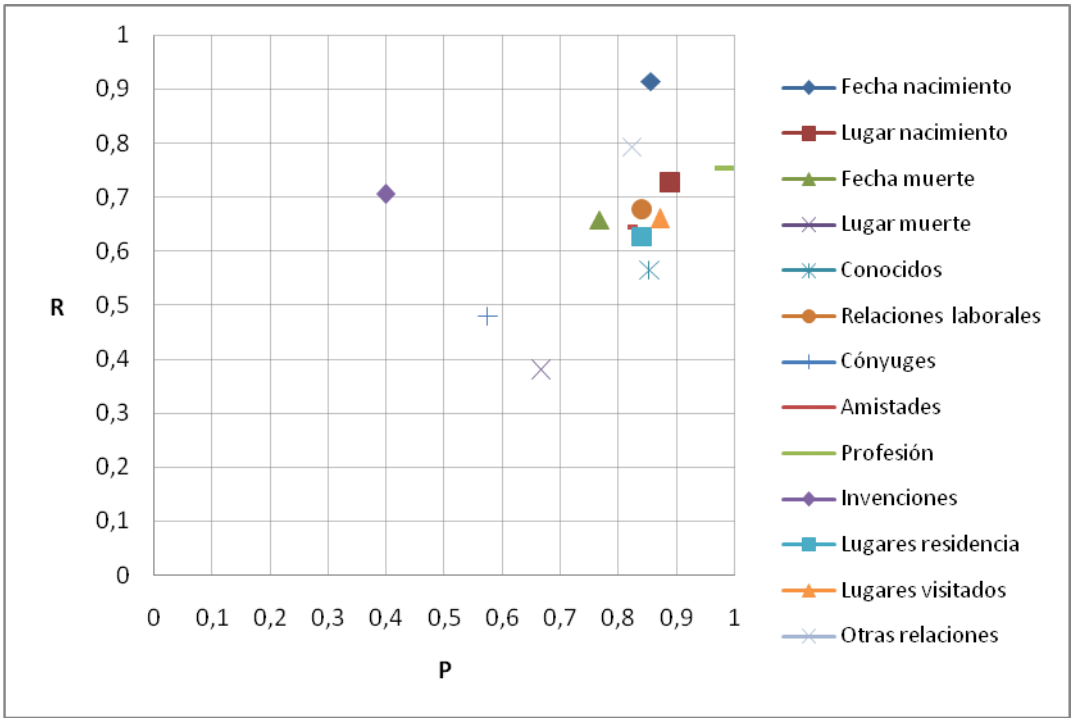


Figura 24. Resultados obtenidos de precisión y cobertura

Puede observarse en la gráfica anterior que el sistema diseñado obtiene generalmente mejores resultados de precisión que de cobertura, aunque para algún caso particular esto no sea así.

6. CONCLUSIONES Y TRABAJOS FUTUROS

6.1. CONCLUSIONES

En este proyecto se ha desarrollado un sistema de EI con el objetivo de extraer una serie de relaciones entre artículos de Wikipedia. Para llevar a cabo esta tarea, se ha necesitado el apoyo de herramientas de análisis morfológico, sintáctico y semántico.

La inmensa mayoría de sistemas de EI se desarrollan para funcionar razonablemente bien dentro de áreas muy restringidas. En el caso de este proyecto, se ha decido trabajar únicamente con artículos de personajes (tanto históricos como contemporáneos), ya que de ese modo resulta más sencillo acotar los diferentes tipos de relaciones de interés que pueden aparecer en los mismos.

Sin embargo, a pesar de haber llevado a cabo el diseño centrándose en artículos de personajes, el sistema no es en absoluto cerrado, sino que podría adaptarse fácilmente a cualquier otro tipo de artículo. Simplemente sería necesario rediseñar adecuadamente los patrones de búsqueda implementados adaptándolos a las relaciones que se necesiten extraer en cada caso. Además, como ya se comentó anteriormente, el sistema será compatible con cualquier artículo fuente en lenguaje natural, no únicamente con aquellos contenidos en Wikipedia.

Al desarrollar el sistema se ha supuesto que los artículos a analizar no contienen en ningún caso errores gramaticales ni ortográficos. Esta premisa podría resultar inadecuada en ciertos entornos, sin embargo se ha observado que este tipo de errores son muy infrecuentes dentro de artículos de Wikipedia. Además el sistema funciona correctamente para artículos de cualquier longitud, aunque en artículos más extensos será capaz de extraer un mayor número de relaciones de interés.

Una de las principales conclusiones a la que puede llegarse tras la implementación de este sistema es la inmensa utilidad que pueden tener las herramientas de procesamiento de lenguaje natural en aplicaciones de EI. La posibilidad

de contar con herramientas de análisis sintáctico, morfológico y semántico ha resultado de gran ayuda en diversos ámbitos del diseño.

Una de las funcionalidades de mayor utilidad es la segmentación del texto, que permite dividir las oraciones en unidades individuales de análisis. Esta división sería mucho más compleja de realizar sin contar con una herramienta de apoyo, ya que en muchos casos el espacio en blanco no determina la separación entre dos unidades lingüísticas.

Además de lo anterior, la información sintáctica, semántica y morfológica proporcionada también resulta de gran utilidad. El aprovechamiento adecuado de esta información y su combinación con el uso expresiones regulares, ha hecho posible que la búsqueda de patrones no se limite exclusivamente al texto en lenguaje natural, sino también a sus características léxicas, semánticas y sintácticas. Esto ha permitido poder trabajar a nivel de sintagma en lugar de a nivel de palabra, desarrollando unos patrones de búsqueda menos complejos y más eficientes. Además, se ha podido comprobar la gran versatilidad de las expresiones regulares tipo Perl para tareas de este tipo.

Cuando se va a desarrollar un sistema de estas características, siempre hay que tener en cuenta que se está trabajando con textos en lenguaje natural, en los que únicamente existe información no estructurada y una misma idea puede ser expresada de una infinidad de formas. Con estas premisas, es comprensible que nunca se conseguirá desarrollar un sistema que dé resultados correctos en el 100% de los casos. En cualquier caso, siempre se deben intentar obtener unos resultados con la mayor precisión y cobertura posibles. De no ser así, deberán sacarse las oportunas conclusiones para tratar de mejorar el sistema.

En el caso del sistema diseñado, se han obtenido valores diferentes de precisión y cobertura según el tipo de relación extraída en cada caso. En la mayoría de casos de ha obtenido un mejor valor de precisión, de lo que se deduce que generalmente es más complejo localizar todos los registros válidos que dar sólo respuestas correctas.

Para las relaciones en las que se han obtenido menores valores de precisión y cobertura, estos resultados son debidos a circunstancias diversas. Una de las posibles

fuentes de error son las expresiones regulares empleadas. También añade cierta dificultad la falta de verbos u otras palabras de apoyo. Generalmente es difícil cubrir todos los casos posibles en los que la información deba ser extraída. No obstante, siempre se pueden analizar los resultados y posteriormente adaptar y mejorar estas expresiones para que el sistema se comporte mejor en un mayor número de situaciones.

Como se ha visto a lo largo del proceso de evaluación, los análisis realizados por el módulo de PLN también pueden ser en algún caso una fuente de error. Como en cualquier herramienta encargada de procesar lenguaje natural, habrá muchos casos en los que se presenten situaciones de ambigüedad y se opte por la opción errónea. En la evaluación realizada se ha podido ver que en ocasiones ciertos errores en los análisis morfológicos, sintácticos y semánticos provocan una disminución de la precisión y la cobertura.

En resumen, puede concluirse que es posible desarrollar sistemas muy útiles de EI si se combinan correctamente las herramientas adecuadas de análisis y procesamiento. Siempre será recomendable tratar de centrarse en campos muy específicos para lograr extraer una información de mayor fiabilidad.

6.2. TRABAJOS FUTUROS

Una vez finalizado el desarrollo del proyecto y tras extraer una serie de conclusiones posteriores a la evaluación del mismo, pueden plantearse ciertas mejoras o líneas de trabajo para complementar las funcionalidades del mismo. A continuación se describen una serie de propuestas en este sentido:

- **Ampliación de patrones de búsqueda**
  En tareas de extracción de ciertas relaciones puede tratar de modificarse las expresiones regulares empleadas con el fin de abarcar más situaciones en las que aparezcan registros válidos. Con esto podría conseguirse un aumento de la cobertura y precisión en ciertos casos.

- **Empleo de bases de datos externas**

  Además de la base de datos semántica ofrecida por STILUS Core, sería de mucha utilidad ampliar el conocimiento externo disponible contando con otra serie de bases de datos de contenido más específico. Una opción sería la posibilidad de consultar una base de datos que almacenase una gran cantidad de nombres y apellidos de persona. De esta forma podría minimizarse la problemática de nombres y apellidos no reconocidos por el sistema. Otra propuesta muy similar sería el uso de una base de datos de localizaciones geográficas. Esta medida supondría en muchos casos un aumento de la precisión del sistema, y sobre todo de la cobertura del mismo.

- **Extracción de información adicional sobre personajes**

  En el sistema desarrollado se han tratado de localizar 12 tipos de relaciones posibles. Lógicamente podría ampliarse la funcionalidad y tratar de extraerse muchos otros tipos de relaciones (relaciones familiares, asociaciones, partidos políticos, equipos donde juega/ha jugado, etc.), aunque en algunos casos éstas entrañarán una dificultad adicional.

- **Adaptación del sistema a entornos más específicos**

  Una posibilidad a tener en cuenta sería trabajar en entornos más restringidos. Por ejemplo, podría tratar de adaptarse el sistema a una categoría específica de personajes (inventores, escritores, etc.). De este modo, la información de interés en todos los casos sería más similar y podrían diseñarse patrones que se adaptasen mejor a todos ellos.

- **Resolución de correferencia**

  Una de las características más útiles en el diseño de un sistema de EI es la resolución de correferencia, ya que permite determinar si ciertos nombres o pronombres aparecidos en el texto hacen referencia a un mismo concepto. Si se integra en el sistema una herramienta con esta funcionalidad enriquecería el proceso de extracción proporcionando resultados que se perderían en una situación normal.

Ejemplo:

Entabló amistad con Max Brod. Posteriormente trabajó con él.

En este ejemplo el sistema diseñado obtendría una relación de amistad con el artículo ‘Max Brod’. Sin embargo también existe una relación laboral que se pierde al no haber una herramienta de resolución de correferencia.

- **Cambio o extensión de dominio a otros tipos**

  Si existiese la necesidad de utilizar el sistema de EI en otros tipos de artículo (empresas u organismos, lugares geográficos, etc.) podría adaptarse el diseño para conseguirlo. Simplemente sería necesario definir los tipos de relación a localizar y adaptar los patrones de búsqueda implementados.

ANEXO A. LISTADO DE ARTÍCULOS EVALUADOS

Tabla 15. Listado de artículos evaluados

Categoría	Personaje	Enlace en Wikipedia
Actualidad	Stephen Hawking	http://es.wikipedia.org/wiki/Stephen_Hawking
	J. K. Rowling	http://es.wikipedia.org/wiki/J._K._Rowling
	Alanis Morissette	http://es.wikipedia.org/wiki/Alanis_Morissette
	Tim Burton	http://es.wikipedia.org/wiki/Tim_Burton
	John Williams (compositor)	http://es.wikipedia.org/wiki/John_Williams_(compositor)
Actores	Brad Pitt	http://es.wikipedia.org/wiki/Brad_Pitt
	Tom Hanks	http://es.wikipedia.org/wiki/Tom_Hanks
	Blanca Portillo	http://es.wikipedia.org/wiki/Blanca_Portillo
	Diane Keaton	http://es.wikipedia.org/wiki/Diane_Keaton
	Ewan McGregor	http://es.wikipedia.org/wiki/Ewan_McGregor
Escritores	Miguel de Cervantes	http://es.wikipedia.org/wiki/Miguel_de_Cervantes
	Camilo José Cela	http://es.wikipedia.org/wiki/Camilo_José_Cela
	Franz Kafka	http://es.wikipedia.org/wiki/Franz_Kafka
	Corín Tellado	http://es.wikipedia.org/wiki/Corín_Tellado
	Tirso de Molina	http://es.wikipedia.org/wiki/Tirso_de_Molina
Pintores	Juan Gris	http://es.wikipedia.org/wiki/Juan_Gris
	Pablo Picasso	http://es.wikipedia.org/wiki/Pablo_Picasso
	Remedios Varo	http://es.wikipedia.org/wiki/Remedios_Varo
	Salvador Dalí	http://es.wikipedia.org/wiki/Salvador_Dalí
	Diego Velázquez	http://es.wikipedia.org/wiki/Diego_Velázquez
Físicos	Isaac Newton	http://es.wikipedia.org/wiki/Isaac_Newton
	Marie Curie	http://es.wikipedia.org/wiki/Marie_Curie
	Galileo Galilei	http://es.wikipedia.org/wiki/Galileo_Galilei
	Lise Meitner	http://es.wikipedia.org/wiki/Lise_Meitner
	Albert Einstein	http://es.wikipedia.org/wiki/Albert_Einstein
Históricos	Cristóbal Colón	http://es.wikipedia.org/wiki/Cristóbal_Colón
	Friedrich Nietzsche	http://es.wikipedia.org/wiki/Friedrich_Nietzsche
	Hernán Cortés	http://es.wikipedia.org/wiki/Hernán_Cortés
	Nerón	http://es.wikipedia.org/wiki/Nerón
	Adolf Hitler	http://es.wikipedia.org/wiki/Adolf_Hitler
Políticos	Manuel Azaña	http://es.wikipedia.org/wiki/Manuel_Azaña
	Salvador Allende	http://es.wikipedia.org/wiki/Salvador_Allende
	Simón Bolívar	http://es.wikipedia.org/wiki/Simón_Bolívar
	Winston Churchill	http://es.wikipedia.org/wiki/Winston_Churchill
	John F. Kennedy	http://es.wikipedia.org/wiki/John_Kennedy
Realeza	Juana I de Castilla	http://es.wikipedia.org/wiki/Juana_I_de_Castilla
	María Antonieta de Austria	http://es.wikipedia.org/wiki/María_Antonieta_de_Austria
	Alfonso XIII de España	http://es.wikipedia.org/wiki/Alfonso_XIII_de_España
	Catalina II de Rusia	http://es.wikipedia.org/wiki/Catalina_II_de_Rusia

	Felipe II de España	http://es.wikipedia.org/wiki/Felipe_II_de_España
Compositores	Manuel de Falla	http://es.wikipedia.org/wiki/Manuel_de_Falla
	Fréderic Chopin	http://es.wikipedia.org/wiki/Chopin
	Wolfgang Amadeus Mozart	http://es.wikipedia.org/wiki/Wolfgang_Amadeus_Mozart
	Isaac Albéniz	http://es.wikipedia.org/wiki/Isaac_Albéniz
	Robert Schumann	http://es.wikipedia.org/wiki/Robert_Schumann
Inventores	Nikola Tesla	http://es.wikipedia.org/wiki/Nikola_Tesla
	Emile Berliner	http://es.wikipedia.org/wiki/Emile_Berliner
	Arquímedes	http://es.wikipedia.org/wiki/Arquímedes
	Antonio Meucci	http://es.wikipedia.org/wiki/Antonio_Meucci
	Alessandro Volta	http://es.wikipedia.org/wiki/Alessandro_Volta

ANEXO B. DETALLE DE RESULTADOS POR ARTÍCULO

Tabla 16. Detalle de resultados por artículo

		Fecha Nacimiento							Lugar Nacimiento						
Categoría	Personaje	CLA	RES	COR	PAR	INC	ESP	PER	CLA	RES	COR	PAR	INC	ESP	PER
Actualidad	Stephen Hawking														
Actualidad	J. K. Rowling	1	1	1											
Actualidad	Alanis Morissette	1	1	1					1	1	1				
Actualidad	Tim Burton	1	1			1									
Actualidad	John Williams (compositor)								1	1	1				
Actores	Brad Pitt								1	1	1				
Actores	Tom Hanks								1	1	1				
Actores	Blanca Portillo	1	1	1					1	1	1				
Actores	Diane Keaton														
Actores	Ewan McGregor		1				1		1						1
Escritores	Miguel de Cervantes	1	1	1					1	1	1				
Escritores	Camilo José Cela	1	1	1					1	1	1				
Escritores	Franz Kafka	1	1	1					1	1	1				
Escritores	Corín Tellado														
Escritores	Tirso de Molina	1	1	1					1	1	1				
Pintores	Juan Gris	1	1	1					1	1	1				
Pintores	Pablo Picasso	1	1	1					1	1	1				
Pintores	Remedios Varo	1	1	1											
Pintores	Salvador Dalí	1	1	1					1	1	1				
Pintores	Diego Velázquez								1	1	1				
Físicos	Isaac Newton	1	1	1					1						1
Físicos	Marie Curie	1	1	1					1	1	1				
Físicos	Galileo Galilei	1	1	1					1	1	1				
Físicos	Lise Meitner	1	1	1					1	1	1				
Físicos	Albert Einstein		1				1								
Históricos	Cristóbal Colón														
Históricos	Friedrich Nietzsche	1	1	1					1	1		1			
Históricos	Hernán Cortés														
Históricos	Nerón	1	1	1					1	1		1			
Históricos	Adolf Hitler								1	1		1			
Políticos	Manuel Azaña														
Políticos	Salvador Allende														
Políticos	Simón Bolívar	1	1		1				1	1	1				
Políticos	Winston Churchill														
Políticos	John F kennedy	1	1	1					1						1
Realeza	Juana I de Castilla	1	1	1					1	1	1				
Realeza	María Antonieta de Austria	1	1	1											
Realeza	Alfonso XIII de España		1				1								
Realeza	Catalina II de Rusia	1	1	1					1	1			1		
Realeza	Felipe II de España														
Compositores	Manuel de Falla	1						1	1						1
Compositores	Fréderic Chopin	1	1	1					1	1		1			
Compositores	Wolfgang Amadeus Mozart	1	1	1					1	1	1				
Compositores	Isaac Albéniz	1	1	1					1	1	1				
Compositores	Robert Schumann								1	1	1				
Inventores	Nikola Tesla	1	1	1					1						1
Inventores	Emile Berliner														
Inventores	Arquímedes														
Inventores	Antonio Meucci	1	1	1					1	1	1				
Inventores	Alessandro Volta								1						1

		Fecha Muerte							Lugar Muerte						
Categoría	**Personaje**	**CLA**	**RES**	**COR**	**PAR**	**INC**	**ESP**	**PER**	**CLA**	**RES**	**COR**	**PAR**	**INC**	**ESP**	**PER**
Actualidad	Stephen Hawking														
Actualidad	J. K. Rowling														
Actualidad	Alanis Morissette														
Actualidad	Tim Burton														
Actualidad	John Williams (compositor)														
Actores	Brad Pitt														
Actores	Tom Hanks														
Actores	Blanca Portillo														
Actores	Diane Keaton														
Actores	Ewan McGregor														
Escritores	Miguel de Cervantes	1	1	1					1	1	1				
Escritores	Camilo José Cela	1	1	1											
Escritores	Franz Kafka	1	1	1											
Escritores	Corín Tellado														
Escritores	Tirso de Molina	1						1	1	1	1				
Pintores	Juan Gris	1	1	1					1						1
Pintores	Pablo Picasso	1	1	1					1						1
Pintores	Remedios Varo	1	1	1					1	1	1				
Pintores	Salvador Dalí	1						1							
Pintores	Diego Velázquez	1	1	1					1	1	1				
Físicos	Isaac Newton	1	1	1											
Físicos	Marie Curie	1	1	1					1						1
Físicos	Galileo Galilei	1	1			1			1						1
Físicos	Lise Meitner	1	1	1					1	1	1				
Físicos	Albert Einstein	1	1	1					1						1
Históricos	Cristóbal Colón														
Históricos	Friedrich Nietzsche	1	1	1											
Históricos	Hernán Cortés	1	1		1				1	1	1				
Históricos	Nerón	1						1							
Históricos	Adolf Hitler	1						1	1						1
Políticos	Manuel Azaña	1	1			1									
Políticos	Salvador Allende		1				1								
Políticos	Simón Bolívar	1	1	1						1				1	
Políticos	Winston Churchill	1	1	1											
Políticos	John F kennedy	1						1	1						1
Realeza	Juana I de Castilla	1	1			1			1	1			1		
Realeza	María Antonieta de Austria	1	1			1									
Realeza	Alfonso XIII de España	1	1	1					1	1	1				
Realeza	Catalina II de Rusia	1	1	1					1						1
Realeza	Felipe II de España	1	1	1						1				1	
Compositores	Manuel de Falla	1	1	1											
Compositores	Fréderic Chopin	1	1			1									
Compositores	Wolfgang Amadeus Mozart	1	1		1				1	1			1		
Compositores	Isaac Albéniz	1	1	1					1						1
Compositores	Robert Schumann	1						1	1						1
Inventores	Nikola Tesla	1	1	1					1	1	1				
Inventores	Emile Berliner	1	1	1											
Inventores	Arquímedes														
Inventores	Antonio Meucci														
Inventores	Alessandro Volta	1	1	1					1						1

		Conocidos							Relaciones laborales						
Categoría	**Personaje**	**CLA**	**RES**	**COR**	**PAR**	**INC**	**ESP**	**PER**	**CLA**	**RES**	**COR**	**PAR**	**INC**	**ESP**	**PER**
Actualidad	Stephen Hawking								1	1	1				
Actualidad	J. K. Rowling		1				1								
Actualidad	Alanis Morissette	1	1				1	1	3						3
Actualidad	Tim Burton								4	3	3				1
Actualidad	John Williams (compositor)	2	2	2					3	2	2				1
Actores	Brad Pitt	2	1	1				1	3	1	1				2
Actores	Tom Hanks														
Actores	Blanca Portillo								1						1
Actores	Diane Keaton	2	1	1				1	3	3	3				
Actores	Ewan McGregor								1	1	1				
Escritores	Miguel de Cervantes	1						1							
Escritores	Camilo José Cela	1	1	1											
Escritores	Franz Kafka	1						1	1	1	1				
Escritores	Corín Tellado														
Escritores	Tirso de Molina														
Pintores	Juan Gris	2	2	2											
Pintores	Pablo Picasso	5	4	4				1	1	1	1				
Pintores	Remedios Varo	2	2	1			1	1							
Pintores	Salvador Dalí	3	2	2				1	1	1	1				
Pintores	Diego Velázquez	3	1	1				2							
Físicos	Isaac Newton	2	1	1				1		1				1	
Físicos	Marie Curie	1	1	1					1						1
Físicos	Galileo Galilei	1	1				1	1							
Físicos	Lise Meitner	1	1	1											
Físicos	Albert Einstein	1	1	1					3	3	3				
Históricos	Cristóbal Colón	1	1	1											
Históricos	Friedrich Nietzsche	5	3	3				2							
Históricos	Hernán Cortés														
Históricos	Nerón	1	1				1	1							
Históricos	Adolf Hitler	1						1							
Políticos	Manuel Azaña	1	1				1	1	1	1	1				
Políticos	Salvador Allende									1				1	
Políticos	Simón Bolívar														
Políticos	Winston Churchill	1	1	1											
Políticos	John F kennedy								1	1	1				
Realeza	Juana I de Castilla														
Realeza	María Antonieta de Austria														
Realeza	Alfonso XIII de España	2	2	2						1				1	
Realeza	Catalina II de Rusia	2						2							
Realeza	Felipe II de España														
Compositores	Manuel de Falla	7	5	5				2	2	2	2				
Compositores	Fréderic Chopin	4	1	1				3							
Compositores	Wolfgang Amadeus Mozart	3	1	1				2							
Compositores	Isaac Albéniz								1	1				1	1
Compositores	Robert Schumann	2	2	2											
Inventores	Nikola Tesla	1						1							
Inventores	Emile Berliner														
Inventores	Arquímedes														
Inventores	Antonio Meucci														
Inventores	Alessandro Volta														

		Cónyuge							Amistades						
Categoría	**Personaje**	**CLA**	**RES**	**COR**	**PAR**	**INC**	**ESP**	**PER**	**CLA**	**RES**	**COR**	**PAR**	**INC**	**ESP**	**PER**
Actualidad	Stephen Hawking														
Actualidad	J. K. Rowling								1	1				1	1
Actualidad	Alanis Morissette														
Actualidad	Tim Burton								4	3	2			1	1
Actualidad	John Williams (compositor)	1						1	3	2	2				1
Actores	Brad Pitt	2	3	2			1		1	2	1			1	
Actores	Tom Hanks	1	1	1					1						1
Actores	Blanca Portillo									2				2	
Actores	Diane Keaton								1						1
Actores	Ewan McGregor	1						1	1						1
Escritores	Miguel de Cervantes								1		1				
Escritores	Camilo José Cela								2	2	2				
Escritores	Franz Kafka								1	1	1				
Escritores	Corín Tellado														
Escritores	Tirso de Molina														
Pintores	Juan Gris														
Pintores	Pablo Picasso	1						1	10	9	8			1	1
Pintores	Remedios Varo														
Pintores	Salvador Dalí								6	7	6			1	
Pintores	Diego Velázquez	1						1							
Físicos	Isaac Newton								2	3	2			1	
Físicos	Marie Curie	1						1							
Físicos	Galileo Galilei								3	2	2				1
Físicos	Lise Meitner														
Físicos	Albert Einstein	2						2	1						1
Históricos	Cristóbal Colón								1						1
Históricos	Friedrich Nietzsche								5	3	3				2
Históricos	Hernán Cortés								1	1	1				
Históricos	Nerón	1	1	1					2	2	1			1	1
Históricos	Adolf Hitler	1						1	1	1	1				
Políticos	Manuel Azaña	1						1	2						2
Políticos	Salvador Allende	1	1	1											
Políticos	Simón Bolívar								1	2				2	1
Políticos	Winston Churchill	1						1							
Políticos	John F kennedy	1	1	1											
Realeza	Juana I de Castilla	1	3				3	1	1	2	1			1	
Realeza	María Antonieta de Austria	1	1	1											
Realeza	Alfonso XIII de España	1	2				2	1	1	1	1				
Realeza	Catalina II de Rusia								2	2	2				
Realeza	Felipe II de España	4	4	4											
Compositores	Manuel de Falla								8	6	6				2
Compositores	Fréderic Chopin		1				1		8	4	4				4
Compositores	Wolfgang Amadeus Mozart		1				1		1						1
Compositores	Isaac Albéniz														
Compositores	Robert Schumann	1	1		1				3	1	1				2
Inventores	Nikola Tesla														
Inventores	Emile Berliner														
Inventores	Arquímedes														
Inventores	Antonio Meucci														
Inventores	Alessandro Volta								1	1	1				

		Profesión							Invenciones						
Categoría	**Personaje**	**CLA**	**RES**	**COR**	**PAR**	**INC**	**ESP**	**PER**	**CLA**	**RES**	**COR**	**PAR**	**INC**	**ESP**	**PER**
Actualidad	Stephen Hawking	2	2	2					1	1	1				
Actualidad	J. K. Rowling	1	1	1					1	2	1			1	
Actualidad	Alanis Morissette	2	1	1				1	3	3	3				
Actualidad	Tim Burton	3	2	2				1	5	8	5			3	
Actualidad	John Williams (compositor)														
Actores	Brad Pitt	2	2	2					2	3	2			1	
Actores	Tom Hanks	1	1	1											
Actores	Blanca Portillo	1	1	1											
Actores	Diane Keaton	1	1	1					1	4	1			3	
Actores	Ewan McGregor	1	1	1						1				1	
Escritores	Miguel de Cervantes	3	3	3					5	7	5			2	
Escritores	Camilo José Cela	1						1		1				1	
Escritores	Franz Kafka	1	1	1					1	1	1				
Escritores	Corín Tellado								1	1	1				
Escritores	Tirso de Molina	2	2	2						2				2	
Pintores	Juan Gris	1	1	1											
Pintores	Pablo Picasso	2	2	2					3	7	1	2		4	
Pintores	Remedios Varo	1	1	1											
Pintores	Salvador Dalí	1	1				1			1				1	
Pintores	Diego Velázquez	1	1	1					2	2	2				
Físicos	Isaac Newton	5	3	3				2	3	4	3			1	
Físicos	Marie Curie	1	1	1											
Físicos	Galileo Galilei	3	3	3					1	3	1			2	
Físicos	Lise Meitner														
Físicos	Albert Einstein	1	1	1											
Históricos	Cristóbal Colón	1	1	1						3				3	
Históricos	Friedrich Nietzsche	3	3	3						2				2	
Históricos	Hernán Cortés	1						1							
Históricos	Nerón	1						1		4				4	
Históricos	Adolf Hitler	2	2	2					1	3	1			2	
Políticos	Manuel Azaña	2	2	2						1				1	
Políticos	Salvador Allende	2	2	2					1	2		1		1	
Políticos	Simón Bolívar	2	2	2											
Políticos	Winston Churchill	1	1	1						4				4	
Políticos	John F kennedy	1						1		2				2	
Realeza	Juana I de Castilla	1						1							
Realeza	María Antonieta de Austria	1						1							
Realeza	Alfonso XIII de España	1						1		1				1	
Realeza	Catalina II de Rusia	1						1		3				3	
Realeza	Felipe II de España	1						1		1				1	
Compositores	Manuel de Falla	1	1	1						1				1	
Compositores	Fréderic Chopin	1	1	1					1	8	1			7	
Compositores	Wolfgang Amadeus Mozart	2	1	1				1	1	2	1			1	
Compositores	Isaac Albéniz	1	1	1					3	3	3				
Compositores	Robert Schumann	1						1		1				1	
Inventores	Nikola Tesla	2	2	2					1	2				2	1
Inventores	Emile Berliner	1						1	3						3
Inventores	Arquímedes	5	4	4				1	2						2
Inventores	Antonio Meucci								2	2	2				
Inventores	Alessandro Volta	1	1	1					3	3	3				

		Lugares residencia							Lugares visitados						
Categoría	**Personaje**	**CLA**	**RES**	**COR**	**PAR**	**INC**	**ESP**	**PER**	**CLA**	**RES**	**COR**	**PAR**	**INC**	**ESP**	**PER**
Actualidad	Stephen Hawking														
Actualidad	J. K. Rowling	3	1	1				2	3	3	3				
Actualidad	Alanis Morissette								2	1	1				1
Actualidad	Tim Burton	1	1	1											
Actualidad	John Williams (compositor)														
Actores	Brad Pitt	1						1	1	1	1				
Actores	Tom Hanks	2	1	1				1							
Actores	Blanca Portillo														
Actores	Diane Keaton														
Actores	Ewan McGregor	1	1	1					1						1
Escritores	Miguel de Cervantes	3	3	2			1		7	8	7			1	
Escritores	Camilo José Cela								1						1
Escritores	Franz Kafka	1	1	1											
Escritores	Corín Tellado	1	1	1											
Escritores	Tirso de Molina	2	4	2			2		2	1	1				1
Pintores	Juan Gris	1	1	1											
Pintores	Pablo Picasso	4	3	3				1	8	8	7			1	
Pintores	Remedios Varo	1						1	1	1	1				
Pintores	Salvador Dalí	1						1	3	3	3				
Pintores	Diego Velázquez								7	7	7				
Físicos	Isaac Newton	1	1	1					1	2	1			1	
Físicos	Marie Curie	2	1	1				1	3	4	3			1	
Físicos	Galileo Galilei	1	1	1											
Físicos	Lise Meitner														
Físicos	Albert Einstein	1	2	1			1								
Históricos	Cristóbal Colón	1	1	1					8	8	5			3	3
Históricos	Friedrich Nietzsche	1						1	3	1	1				2
Históricos	Hernán Cortés	1	2	1			1		5	3	3				2
Históricos	Nerón								2	3	2			1	
Históricos	Adolf Hitler								3	7	3			4	
Políticos	Manuel Azaña	4	3	3				1	3	2	2				1
Políticos	Salvador Allende	1	1	1					1	1	1				
Políticos	Simón Bolívar	2	2	2					17	21	17			4	
Políticos	Winston Churchill	1						1	7	5	5				2
Políticos	John F kennedy	2	1	1				1	10	12	9			3	1
Realeza	Juana I de Castilla	2	2	2					2	2	2				
Realeza	María Antonieta de Austria								1	1	1				
Realeza	Alfonso XIII de España	2	2	2					2						2
Realeza	Catalina II de Rusia								1	1	1				
Realeza	Felipe II de España	2						2	3	2	1			1	2
Compositores	Manuel de Falla	4	2	2				2	4						4
Compositores	Fréderic Chopin	2	2	2					14	14	13			1	1
Compositores	Wolfgang Amadeus Mozart	1	1				1	1	4	2	2				2
Compositores	Isaac Albéniz	3	2	2				1	1	1	1				
Compositores	Robert Schumann								1						1
Inventores	Nikola Tesla	2	2	2					1	1	1				
Inventores	Emile Berliner	3	2	2				1							
Inventores	Arquímedes								1	1	1				
Inventores	Antonio Meucci	1						1							
Inventores	Alessandro Volta								1	1	1				

		Otros						
Categoría	**Personaje**	**CLA**	**RES**	**COR**	**PAR**	**INC**	**ESP**	**PER**
Actualidad	Stephen Hawking	94	94	82	0	12		
Actualidad	J. K. Rowling	93	93	83	4	6		
Actualidad	Alanis Morissette	67	67	56	2	9		
Actualidad	Tim Burton	48	48	30	0	18		
Actualidad	John Williams (compositor)	56	56	48	1	7		
Actores	Brad Pitt	46	46	22	0	24		
Actores	Tom Hanks	37	37	28	0	9		
Actores	Blanca Portillo	26	26	17	0	9		
Actores	Diane Keaton	35	35	29	0	6		
Actores	Ewan McGregor	27	27	21	0	6		
Escritores	Miguel de Cervantes	39	39	32	0	7		
Escritores	Camilo José Cela	28	28	19	1	8		
Escritores	Franz Kafka	31	31	20	0	11		
Escritores	Corín Tellado	35	35	34	0	1		
Escritores	Tirso de Molina	36	36	7	0	29		
Pintores	Juan Gris	34	34	31	0	3		
Pintores	Pablo Picasso	30	30	16	0	14		
Pintores	Remedios Varo	38	38	36	0	2		
Pintores	Salvador Dalí	26	26	7	0	18		
Pintores	Diego Velázquez	53	53	47	1	5		
Físicos	Isaac Newton	68	68	65	0	3		
Físicos	Marie Curie	57	57	49	0	8		
Físicos	Galileo Galilei	58	58	48	1	9		
Físicos	Lise Meitner	18	18	17	0	1		
Físicos	Albert Einstein	72	72	60	3	9		
Históricos	Cristóbal Colón	53	53	47	2	4		
Históricos	Friedrich Nietzsche	61	61	55	1	5		
Históricos	Hernán Cortés	52	52	49	1	2		
Históricos	Nerón	68	68	64	0	4		
Históricos	Adolf Hitler	37	37	34	0	3		
Políticos	Manuel Azaña	68	68	56	2	10		
Políticos	Salvador Allende	56	56	50	1	5		
Políticos	Simón Bolívar	57	57	51	0	6		
Políticos	Winston Churchill	71	71	67	0	4		
Políticos	John F kennedy	59	59	45	0	14		
Realeza	Juana I de Castilla	37	37	34	1	2		
Realeza	María Antonieta de Austria	53	53	47	0	6		
Realeza	Alfonso XIII de España	44	44	33	1	10		
Realeza	Catalina II de Rusia	40	40	39	0	1		
Realeza	Felipe II de España	55	55	54	0	1		
Compositores	Manuel de Falla	63	63	60	1	2		
Compositores	Fréderic Chopin	71	71	65	0	6		
Compositores	Wolfgang Amadeus Mozart	69	69	62	1	6		
Compositores	Isaac Albéniz	45	45	42	0	2		
Compositores	Robert Schumann	60	60	42	1	17		
Inventores	Nikola Tesla	46	46	36	2	8		
Inventores	Emile Berliner	16	16	13	0	3		
Inventores	Arquímedes	35	35	34	0	1		
Inventores	Antonio Meucci	28	28	27	0	1		
Inventores	Alessandro Volta	37	37	29	2	6		

Abreviaturas utilizadas en Anexo B:

CLA	Claves
RES	Respuestas
COR	Respuestas correctas
PAR	Respuestas parciales
INC	Respuestas incorrectas
ESP	Respuestas espurias
PER	Respuestas perdidas

BIBLIOGRAFÍA Y REFERENCIAS

Achour, M., Betz, F., Dovgal, A., Lopes, N., Magnusson, H., Richter, G., Seguy, D. y Vrana, J. (2009)

PHP Documentation

http://www.php.net/docs.php [Visitado el 13/08/2009]

Appelt, D., Hobbs, J. R., Bear, J., Israel, D. y Tyson, M. (1993).

FASTUS: Extracting Information from Natural-Language Texts.

Artificial Intelligence Center. SRI International. Menlo Park, California.

http://www.ai.sri.com/natural-language/projects/fastus-schabes.html

[Visitado el 14/12/2007]

Appelt, D. E. y Israel, D.J. (1999).

Introducion to Information Extraction Technology.

A tutorial prepared for IJCAI-99. Artificial Intelligence Center.

http://www.ai.sri.com/~appelt/ie-tutorial/ [Visitado el 1/11/2007]

Bagga, A. y Baldwin, B. (1998).

Algorithms for scoring coreference chains.

Proceedings of the Linguistic Coreference Workshop at the First International Conference on Language Resources and Evaluation (LREC'98) (pp. 563-566).

LREC.

Bishop, C. M. (1995).

Neural Networks for Pattern Recognition.

Oxford: Oxford University Press.

Blum, A. y Mitchell, T. (1998).

Combining labeled with unlabeled data with co-training.

Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT) (pp. 92-100).

San Francisco, CA: Morgan Kaufmann.

Cáceres, A.E. (2008).

La métrica de Levenshtein

Universidad Juárez Autónoma de Tabasco, DACB

http://www.dacb.ujat.mx/publicaciones/revista_dacb/Acervo/v7n2OL/v7n2a4.pdf

[Visitado el 17/09/2009]

Cherkassky, V. S., Friedman, J. y Wechsler, H. (1996).

From Statistics to Neural Networks: Theory and Pattern Recognition application. Springer.

Chinchor, N. (1992).

MUC-4 Evaluation metrics.

Proceedings of the 4th Message Understanding Conference (MUC-4) (pp. 22-50).

San Mateo, CA: Morgan Kaufmann.

Collins, M. (2002).

Ranking algorithms for named-entity extraction: Boosting and the voted perceptron.

Proceedings of the 40th Annual Meeting of the Association of Computational Linguistics (pp. 489-496).

San Franciso: Morgan Kaufmann.

Cunningham, H. (1999).

Information Extraction – a User Guide (Second edition).

Institute for Language, Speech and Hearing (ILASH).

http://www.dcs.shef.ac.uk/~hamish/IE/ [Visitado el 24/11/2007]

DAEDALUS - Data, Decisions and Language, S. A. (s.f.)

STILUS Core

http://www.daedalus.es/productos/stilus/stilus-core/ [Visitado el 08/08/2009]

Dasarathy, B. V. (1991).

Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques.

IEEE Computer Society.

DeJong, G. (1982).

An overview of the FRUMP system.

Computer Science Department. Yale University.

http://dli.iiit.ac.in/ijcai/IJCAI-77-VOL1/PDF/003B.pdf [Visitado el 20/01/2008]

Doddington, G., Mitchell, A., Przybocki, M. y Ramshaw, L. (2004).

The Automatic Content Extraction (ACE) Program. Tasks, Data, and Evaluation.

http://papers.ldc.upenn.edu/LREC2004/ACE.pdf [Visitado el 14/12/2007]

Duda, R.O., Hart, P. E. y Stork, D. G. (2001).

Unsupervised Learning and Clustering.

En *Pattern Classification* (2nd edition, p. 571).

Wiley, New York.

González Fernández, I. (2006).

Evaluación de la recuperación de documentos.

http://evaluacion-recuperacion.iespana.es/ [Visitado el 14/12/2007]

Hayes, P. J. y Weinstein, S.P. (1991).

CONSTRUE/TIS: A system for content-based indexing of a database of news stories.

2nd Annual Conference on Innovative Applications of Artifical Intelligence (pp. 49-64).

Menlo Park, CA: AAAI Press.

Jacobs, P. S. y Rau, L.F. (1990).

"SCISOR": Extracting information from on-line news.

Communications of the ACM, 33 (11), 88-97.

Kaufman, L. y Rousseeuw, P. J. (1990).
Finding Groups in Data: An Introduction to Cluster Analysis.
New York: John Wiley and Sons.

Kohonen, T. (2001).
Self-Organizing Maps.
Third, extended edition. Springer.

Kvale, M. (2000).
Perl regular expressions tutorial
http://perldoc.perl.org/perlretut.html [Visitado el 14/08/2009]

Mani, I., Schiffmann, B. y Zhang, J. (2003)
Interferring temporal ordering of events in news.
Proceedings of the Human Language Technology Conference (pp. 55-57).
Edmonton, CA.

Manning, C. (2007).
Statistical natural language processing and corpus-based computational linguistics: An annotated list of resources.
Stanford University.
http://www-nlp.stanford.edu/links/statnlp.html [Visitado el 17/09/2009]

McCallum, A., Nigam, K., Rennie, J. y Seymore, K. (1999).
A machine learning approach to building domain specific search engines.
Proceedings of the 16th International Joint Conference on Artificial Intelligence (pp. 662-667).
San Mateo, CA: Morgan Kaufmann.

McCarthy, J. y Lehnert, W.G. (1995).
Using decision trees for coreference resolution.
Proceedings of the 14th International Joint Conference on Artificial Intelligence (pp. 1050-1055).
San Mateo, CA: Morgan Kaufmann.

Minsky, M. (1975).

A framework for representing knowledge.

The Psychology of Computer Vision (pp. 211-277).

New York. McGraw-Hill.

Mitchell, T. (1977).

Version spaces: A candidate elimination approach to rule learning.

Proceedings of the 5th International Joint Conference on Artificial Intelligence (pp. 305-310).

Cambridge, MA: William Kaufmann.

Mitchell, T. (1997).

Machine Learning.

MacGraw Hill.

Moens, M. F. (2006).

Information Extraction: Algorithms and Prospects in a Retrieval Context.

Katholieke Universiteit Leuven, Belgium. Published by Springer.

Moreda, P. (2008).

Los roles semánticos en la tecnología del lenguaje humano: anotación y aplicación.

Universidad de Alicante.

http://www.cervantesvirtual.com/servlet/SirveObras/12140522329062617654435/032073.pdf [Visitado el 17/09/2009]

National Institute of Standards and Technology. (2001).

MUC Evaluations.

http://www.itl.nist.gov/iaui/894.02/related_projects/muc/ [Visitado el 22/09/2009]

Ng, V. y Cardie, C. (2002).

Improving machine learning approaches to coreference resolution.

Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (pp. 104-111).

San Francisco: Morgan Kaufmann.

Palmer, D. D. (2000).

Tokenisation and sentence segmentation.

Handbook of Natural Language Processing (pp. 11-35).

New York, NY: Marcel Dekker.

Pazienza, M.T. (1997).

Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology.

International Summer School, SCIE-97.

Rabiner, L. L. (1989).

A tutorial on hidden Markov models and selected applications.

Proceedings of the IEEE 77 (pp. 257-285).

Los Alamitos, CA: The IEEE Computer Society.

Rosenblatt, F. (1958).

The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain.

Psychological Review, v65, No. 6 (pp. 386-408).

Schank, R. C. (1972).

Conceptual dependency: A theory of natural language understanding.

Cognitive Psychology, 3 (4), 532-631.

Shen et al. (2004).

Multi-criteria-based active learning for named entity recognition.

Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (pp. 590-597).

East Stroudsburg, PA: ACL.

Szabó, Z. G. (2004).

Compositionality.

The Standard Encyclopedia of Philosophy (Edición de Otoño 2004).

Text Retrieval Conference (TREC). (s.f.).

http://trec.nist.gov/ [Visitado el 29/10/2007]

The PHP Group. (2009).

PHP: Hypertext Preprocessor (sitio oficial).

http://www.php.net/ [Visitado el 23/09/2009]

Theodoridis, S. y Koutroumbas, K. (2003).

Pattern Recognition.

Amsterdam, Países Bajos: Academic Press.

Vilain, M. et al. (1995).

A Model-Theoretic Coreference Scoring Scheme.

Proceedings of the 6th Message Understanding Conference (MUC-6) (pp. 45-52).

Vilares, J. (2008)

Lenguajes naturales. Extracción de información.

Facultad de Informática. Universidad de La Coruña.

http://www.grupolys.org/docencia/ln/2008-09/ln_extraccion_de_informacion.pdf

[Visitado el 22/08/2009]

Voorhees, E. (2001).

Introduction to Information Extraction.

http://www-nlpir.nist.gov/related_projects/muc/index.html

[Visitado el 14/10/2007]

Wikimedia Foundation (2009)

Wikipedia:About

http://en.wikipedia.org/wiki/Wikipedia:About

[Visitado el 17/09/2009]

Young, S. R. y Hayes, P.J. (1985).

Automatic classification and summarization of banking telexes.

The Second Conference on Artificial Intelligence Applications: The Engineering of Knowledge Based Systems (pp. 402-408).

Washington, DC: IEEE Computer Society Press.

Zabalegui, J.A (2007).

Recuperación y Organización de la información.

http://extraccioninformacion.iespana.es/ [Visitado el 29/10/2007]

Zhao, S. (2004).

Information Extraction from Multiple Syntactic Sources.

Department of Computer Science. New York University.