

FACULTAD DE INGENIERÍA
UNIVERSIDAD DE LA REPÚBLICA

PROYECTO DE GRADO

WebQA
Respuesta automática a preguntas

Informe Final

11 de junio de 2007

Autores:
Daniel Castelo
Jorge Isi
Sebastián Martínez

Tutores:
Diego Garat
Guillermo Moncecchi

Resumen

Los sistemas de *búsqueda de respuestas* buscan responder de forma precisa preguntas concretas que un usuario realiza en lenguaje natural. Para esto deben analizar la pregunta, localizar y extraer la respuesta de un corpus de documentos y finalmente presentarla al usuario. Esta área de investigación está teniendo un crecimiento importante debido a la demanda de este tipo de sistemas y a la organización de conferencias para su investigación y evaluación, como son las TREC y las CLEF.

Uno de los escenarios en que estos sistemas tienen mayor importancia es cuando el corpus de documentos es la Web. En un corpus de documentos acotado se hace necesario aplicar técnicas de *Procesamiento de Lenguaje Natural* relativamente complejas para ubicar una respuesta, ya que generalmente se encuentra en un lugar específico del documento. Por otro lado, aprovechando la redundancia de información de la Web, con otras técnicas más sencillas se pueden obtener buenos resultados con menos procesamiento.

En el presente trabajo se desarrolló el sistema WebQA de *búsqueda de respuestas* para el idioma español utilizando la Web como repositorio de información. Las preguntas que responde son las fácticas, o sea, aquellas que esperan una única respuesta correcta y concreta. Para recuperar los documentos se recurre a los motores de búsqueda de Yahoo y Google. De éstos se analizan los fragmentos retornados por el buscador (*snippets*) y se ponderan según la probabilidad de contener la respuesta. A partir de este análisis, se aplican técnicas para extraer la respuesta concreta a la pregunta. Al usuario se le presentan los fragmentos ordenados, resaltando la respuesta extraída. Finalmente, se realiza una evaluación del sistema y una comparativa con otros presentados en los foros de investigación de las conferencias TREC y CLEF. El resultado de WebQA es muy bueno, demostrando que es posible responder correctamente preguntas fácticas de interés general, explotando la Web como fuente de información.

Palabras Clave: Búsqueda de repuestas en la Web, recuperación de información, procesamiento de lenguaje natural.

*Gustavo, hermano, donde quiera que estés, todo
te lo dedico a vos.*

Índice general

1. Introducción	1
1.1. Objetivo	2
1.2. Estructura del documento	3
2. Marco teórico	5
2.1. Reseña histórica	5
2.2. Módulos clásicos de los sistemas de BR	6
2.2.1. Análisis del tipo de la pregunta	8
2.2.2. Recuperación de documentos	8
2.2.3. Selección de pasajes relevantes	12
2.2.4. Extracción de Respuestas	12
2.2.5. Formulación de respuestas	13
3. El sistema WebQA	15
3.1. Análisis de la pregunta	17
3.2. Recuperación de documentos de la Web	19
3.2.1. Reformulador según patrones establecidos	20
3.2.2. Reformulador en base a permutación de dos primeras palabras	22
3.2.3. Reformulador en base a conjunción de palabras clave	23

3.3.	Filtrado y ponderación de fragmentos	24
3.4.	Extracción de la respuesta mediante análisis de ngramas . . .	28
3.4.1.	Recolección y ponderación de ngramas	29
3.4.2.	Solapamiento de NGramas	30
3.4.3.	Reponderación de fragmentos según ngrama respuesta	33
3.5.	Presentación de la respuesta	33
4.	Implementación	35
4.1.	Componentes de WebQA	36
4.2.	Implementación del flujo de componentes	41
4.3.	Implementación de patrones	43
4.3.1.	Patrones para el análisis de la pregunta	43
4.3.2.	Reformulador basado en patrones	47
4.4.	Inclusión de herramientas para análisis	48
4.5.	Motor de búsqueda	49
5.	Evaluación	51
5.1.	Forma de Evaluación	51
5.1.1.	En base a la respuesta extraída	52
5.1.2.	En base a los fragmentos presentados	52
5.2.	Conjunto de preguntas utilizado	53
5.3.	Resultado de la Evaluación	53
5.3.1.	En base a la respuesta extraída	54
5.3.2.	En base a los fragmentos presentados	55
5.3.3.	Análisis según tipo de pregunta	56
5.4.	Comparación de WebQA con otros sistemas	57
5.4.1.	Conjunto de preguntas de evaluación	57

5.4.2. Comparación de WebQA con sistemas participantes en las TREC	59
5.4.3. Comparación de WebQA con los sistemas presentados en las CLEF	60
5.4.4. Comparación de WebQA con el sistema de Alejandro del Castillo	62
5.5. Problemas detectados	62
5.6. Conclusión	63
6. Conclusiones y trabajos a futuro	65
6.1. Conclusiones	65
6.2. Trabajos a futuro	67
Glosario	69
Bibliografía	75
A. Apéndice Evaluación	77
A.1. Evaluación del impacto de la variación de Parámetros	77
A.2. Conjunto de prueba de 50 preguntas	84
A.3. Conjunto de prueba de 150 preguntas	86
A.4. Conjunto de Prueba de 200 preguntas CLEF 2003	92

Capítulo 1

Introducción

En los últimos tiempos, el uso de las computadoras como medio para respaldar, procesar y obtener información se ha perfeccionado. La Web se ha convertido en el mayor repositorio de información existente. Es por ello que es necesario aprovecharlo, mediante sistemas eficientes y precisos de búsqueda de información.

Actualmente los sistemas predominantes son los de *Recuperación de Información*. Tienen como objetivo obtener los documentos que son relevantes ante una consulta de un usuario. Los más conocidos son los que permiten localizar información a través de Internet (motores de búsqueda). Por ejemplo, Google e Yahoo son sistemas que califican dentro de esta categoría. En ellos se ingresa un conjunto de palabras para recuperar documentos de la Web que las contengan. El usuario luego debe buscar entre los resultados cuál es el documento que le interesa.

Procurando mejorar el rendimiento de los sistemas de *Recuperación de Información*, se puede destacar la aplicación de técnicas de *Procesamiento de Lenguaje Natural* (PLN) [12]. Esta última es una disciplina que estudia el lenguaje natural como objeto computacional. Se utiliza en variedad de aplicaciones, por ejemplo: sistemas de diálogo, de recuperación de información o de traducción automática.

Uno de los principales promotores de la investigación de los sistemas de *Recuperación de Información* son las conferencias TREC (*Text Retrieval Conference*) [21], en donde se lleva a cabo una evaluación y un seguimiento de las investigaciones desarrolladas en este campo. De aquí en adelante se referirá a los sistemas de recuperación de información como los sistemas de RI.

Por otro lado están los sistemas de *Extracción de Información*, que bus-

can información muy concreta en colecciones de documentos. Un ejemplo es, dado un documento, extraer determinada información de una persona (nombre, apellido, cédula, dirección, teléfono, etc.).

Si bien los sistemas de *Recuperación de Información y Extracción de Información* facilitan al usuario la búsqueda de información, presentan problemas cuando el usuario quiere obtener respuestas concretas a preguntas precisas formuladas en lenguaje natural. A partir de esto surgen los sistemas de *Búsqueda de Respuestas*. Estos tratan de resolver el problema de localizar, extraer y presentar al usuario, única y exclusivamente aquella información que desea conocer, evitándole el trabajo de tener que buscarla en los distintos documentos. Se definen como herramientas capaces de obtener respuestas concretas a necesidades de información muy precisas a partir de una pregunta y un análisis de documentos escritos en lenguaje natural [1].

Estos sistemas, según la fuente de información utilizada, podemos dividirlos en dos grandes grupos: los que buscan respuestas en un corpus de documentos y los que toman la Web como repositorio de información. En el primero se deben aplicar técnicas complejas de PLN para ubicar la respuesta, ya que generalmente ésta se encuentra en un lugar específico del documento. Sin embargo, la Web es mucho más grande que cualquier otro corpus y está en constante crecimiento, por lo que es muy probable que dada una pregunta contenga la respuesta, y repetida en distintos lugares. Por tal motivo no es tan necesario utilizar técnicas avanzadas de PLN, ya que con otras más sencillas también se obtienen buenos resultados y con menos procesamiento. Este proyecto se corresponde al segundo caso.

Las investigaciones de los sistemas de *búsqueda de respuestas* se vienen desarrollando rápidamente, principalmente gracias a la combinación de dos factores: la creciente demanda y la organización de una tarea para la evaluación de los mismos: las conferencias TREC. Estas investigaciones hasta el momento son básicamente para el inglés. Es muy poco lo que se ha hecho en español.

1.1. Objetivo

El principal objetivo de este trabajo es el desarrollo de un sistema de *búsqueda de respuestas* en español, que utilice la Web como fuente de información.

Como entrada debe recibir una pregunta fáctica formulada en lenguaje natural. Estas preguntas son las que requieren una única respuesta correcta y concreta, y además tienen un pronombre interrogativo claro (cuándo, dónde,

quién, etc.). Un ejemplo es “*¿En qué ciudad está la Torre Eiffel?*”, donde se espera como respuesta “*París*”. Otro tipo de preguntas más complejas no se atacan en este proyecto (por ejemplo: preguntas causales, de recopilación de información, las que requieren cierto análisis profesional, etc.).

El sistema debe aprovechar la redundancia de la Web. Esto es fundamental, no solo porque se evita la necesidad de utilizar recursos léxicos importantes, sino también porque permite obtener respuestas fiables, ya que la calidad y confiabilidad de los documentos de la Web puede ser muy variable (una respuesta extraída de un único documento de la Web no es tan confiable).

Para recuperar los documentos es necesario enviar consultas por medio de un motor de búsqueda (por ejemplo: Google, Yahoo, etc.). Estas se obtienen aplicando diversas técnicas a partir de la formulación de la pregunta. Por ejemplo, para la pregunta “*¿Cuándo se jugó el primer mundial de fútbol?*” una consulta posible es “*el primer mundial de fútbol se jugó en*”.

La respuesta puede presentarse de diferentes maneras: un ranking de los fragmentos recuperados de la Web ordenados por su probabilidad de contener la respuesta, una respuesta concreta, una respuesta formulada en lenguaje natural, etc.

Para el desarrollo del sistema inicialmente se debe realizar un estudio del problema y obtener una visión general de los sistemas de BR existentes. Luego centrarse en los que explotan la Web como repositorio de información y en las investigaciones realizadas para el idioma español, estudiando las técnicas empleadas.

1.2. Estructura del documento

El documento se divide de la siguiente manera:

El capítulo 2 presenta un marco teórico de los sistemas de búsqueda de respuestas: Se da una breve reseña histórica y estado actual de estos sistemas. También detalla los módulos clásicos, presentando diversas técnicas para cada módulo y ejemplos de sistemas existentes.

En el capítulo 3 se describe el sistema desarrollado WebQA, haciéndose hincapié en los algoritmos y técnicas utilizadas, explicando detalladamente cada etapa que compone el proceso de recibir una pregunta y retornar una respuesta al usuario.

En el capítulo 4 se describen puntos a destacar de la implementación del

sistema WebQA. Se mencionan los aspectos técnicos más importantes de la solución.

El capítulo 5 trata de la evaluación del sistema. De aquí se pueden determinar los resultados obtenidos a partir de las técnicas utilizadas (explicadas en el capítulo 3). Se presentan las medidas de evaluación tomadas y también se compara WebQA con otros sistemas similares.

El documento finaliza con el capítulo 6, el cual contiene las conclusiones finales y los posibles trabajos a futuro que se podrían realizar, de modo de solucionar problemas detectados y mejorar el rendimiento general del sistema.

Capítulo 2

Marco teórico

2.1. Reseña histórica

La primera especificación de lo que un sistema de BR debe cumplir fue llevada a cabo por Wendy Lehnert a final de la década del 70 [12]. Se identificaron las etapas de entender la pregunta del usuario, buscar la respuesta en una base de conocimiento y componerla para presentarla al usuario. Para estas etapas se concluyó que era necesario aplicar técnicas de PLN para entender la pregunta y formularla, y técnicas de búsqueda de conocimiento para buscar posibles respuestas, analizando además el caso de tener que inferir nuevo conocimiento a partir del existente.

El verdadero surgimiento de este tipo de sistemas se da en los 90 con varios sistemas desarrollados con distintas finalidades, dos de ellos para dominios restringidos fueron, un sistema de ayuda para consulta turística desarrollado para alemán llamado *LIALOG*(1991) y el otro *The Unix Consultant*(1998), un sistema de ayuda en lenguaje natural para Unix. En tanto para dominios no restringidos los sistemas que más se destacaron fueron, el sistema de *Wendlandt & Driscoll*(1991), creado para responder preguntas referentes a las misiones de la NASA y el sistema *MURAX* que contestaba preguntas realizadas por el usuario utilizando como fuente de conocimiento la enciclopedia Grolier. [12]

En gran medida la intensificación del estudio de BR en dominios no restringidos se debió a la creación de espacios de difusión e investigación como las TREC y las CLEF. Las Conferencias TREC son uno de los principales foros de investigación de sistemas de recuperación de información. Auspiciadas por el *Nacional Institute of Standard and Technology* (NIST) y por la *Defense Advanced Research Projects Agency* (DARPA), comenzaron en el

año 1992 y vienen realizándose hasta la fecha. Su objetivo es crear un espacio para comparar los sistemas de los distintos participantes y centralizar la información sobre la evolución de ellos a lo largo del tiempo; esto es posible dado que todos operan con las mismas colecciones y las mismas consultas además de presentar sus resultados de la misma forma. Estos sistemas utilizan técnicas diferentes, y es justamente lo que se trata de comparar. No fue hasta el año 1999 cuando se presentó una nueva sección orientada a los sistemas de BR, con la finalidad de fomentar la investigación en este campo y potenciar la mejora de los sistemas existentes. El rendimiento se evalúa sobre un número de preguntas de prueba, elaborada por la organización. Las respuestas a estas preguntas están en una colección de documentos dada. Cabe aclarar que con el paso de los años el nivel de complejidad y dificultad de los requerimientos de esta sección de las TREC a ido en aumento. [21]

El *Cross-Language Evaluation Forum* (CLEF) [4] creado en el 2000, es un foro cuyo fin es el de brindar la infraestructura para el testeo, ajuste y evaluación de los sistemas de recuperación de información y búsqueda de respuestas que soporten al menos un lenguaje europeo. Otro cometido muy importante de este foro es el de crear una comunidad de investigadores y desarrolladores que estudien y resuelvan los problemas que se presentan en estos sistemas.

La historia de sistemas construidos para dar soporte al español comienza con trabajos realizados por grupos españoles y mejicanos en los años 2003 y 2004, los cuales fueron presentados en las CLEF. Uno de los principales inconvenientes que tienen los sistemas de BR para el español en comparación con sistemas desarrollados para otros idiomas, es que solo el 2,4% de los documentos que se encuentran en la Web son para este idioma, mientras que para el inglés, por ejemplo, el porcentaje es del 68,4%. [20] Algunos ejemplos de sistemas son, el sistema de la Universidad de Alicante [22] que presenta una arquitectura general basada en las etapas clásicas de los sistemas de BR (análisis de la pregunta, extracción de pasajes relevantes y extracción de la respuesta), el sistema Miracle [7] que tiene como principal característica la utilización de técnicas de PLN de nivel semántico y por último el sistema del Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) de México [6] que se basa en el análisis del contexto utilizando etiquetadores de entidades para indexar las entidades nombradas en los documentos.

2.2. Módulos clásicos de los sistemas de BR

Visto desde un punto de vista general los módulos clásicos de un sistema de BR son las siguientes [12]

1. Análisis del tipo de la pregunta: se identifica el tipo de la pregunta y se obtienen las principales entidades y su contexto.
2. Recuperación de documentos: en base a los datos obtenidos en la etapa anterior se buscan los documentos que puedan llegar a tener la respuesta a la pregunta.
3. Selección de pasajes relevantes: dentro de los documentos obtenidos en la etapa de recuperación se seleccionan aquellos fragmentos que estén relacionados con la pregunta.
4. Extracción de Respuestas: se realiza un análisis de los fragmentos con el fin de localizar aquellos extractos que se considera que contienen la respuesta.
5. Formulación de respuestas: en base a la pregunta y a los extractos obtenidos se formula en lenguaje natural la respuesta para presentarle al usuario.

En la figura 2.1 se presenta una arquitectura general de los sistemas de BR, donde están incluidos estos módulos y su relación. Existen sistemas en los cuales la división de los módulos no es clara y en ciertos casos se solapan entre ellos, como por ejemplo el módulo de selección de pasajes con el módulo de extracción de respuestas.

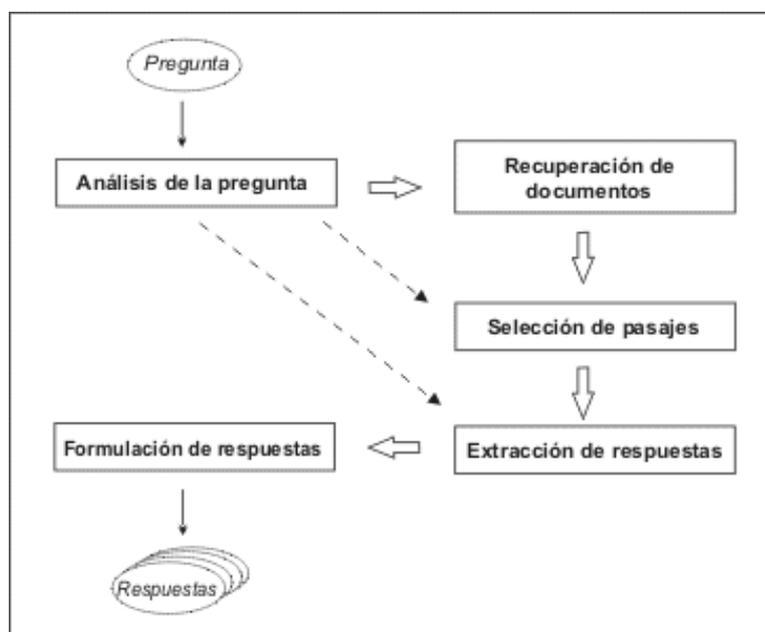


Figura 2.1: Arquitectura general de los sistemas de BR.

Cada sistema en particular utiliza diferentes técnicas para el tratamiento tanto de las preguntas como de los documentos donde se encuentra la respuesta. A continuación se analizará de qué se trata cada etapa y se presentarán tres ejemplos de este tipo de sistemas: AskMSR[2], Webclopedia[14] y AnswerBus[2]. Al final de esta sección se puede observar en la tabla 2.2 una comparación de estos tres sistemas.

2.2.1. Análisis del tipo de la pregunta

En esta etapa se recibe la pregunta formulada por el usuario y a partir de ella se identifica el tipo de pregunta, se extraen entidades y su contexto, pudiendo efectuar además, un etiquetado y análisis sintáctico [1]. Este punto es de gran importancia dado que de la cantidad y calidad de información extraída en este análisis dependerá en gran medida el resultado final del sistema.

Para identificar la pregunta depende de las clasificaciones que se manejen. Por ejemplo es posible identificar preguntas por su tipo “*Cuándo*”, “*Por qué*”, “*Cómo*”, “*Dónde*”. A su vez, dada la pregunta se debe determinar qué es lo que se quiere como respuesta: una fecha, un lugar, una persona. También se puede identificar más información dentro de la pregunta como el sujeto, predicado, sustantivo, verbo, adjetivos, etc.

El grado de complejidad en la búsqueda de la respuesta varía según el tipo de pregunta que sea [5]. Por ejemplo las preguntas del tipo “*Cuándo*” y “*Dónde*” se pueden mapear directamente al tipo de respuesta “*fecha*” y “*lugar*” respectivamente. En el caso de la pregunta de tipo “*Quién*”, la complejidad aumenta, ya que se puede estar buscando una respuesta que sea algo más que una persona. Por ejemplo con la pregunta “¿Quién es David Beckham?”, se esperaría que el sistema devuelva una descripción de esta persona. A diferencia de las preguntas anteriores, las de la forma “*Qué*”, “*Cómo*” y “*Por qué*”, son extremadamente ambiguas en el momento de determinar el concepto de la respuesta [5].

Para la comprensión de preguntas, la mayoría de los sistemas utilizan diccionarios, etiquetadores de categorías gramaticales (*POS-tagger*), analizadores sintácticos (*parser*), sistemas de clasificación de entidades (*Name-Entity tagger*) y clasificadores de preguntas (*Question-Type classifier*).

2.2.2. Recuperación de documentos

A partir del tipo de la pregunta y las principales entidades obtenidas en la etapa anterior, se comienza el proceso de recuperación de los documentos

en los cuales se puede encontrar la respuesta a la pregunta. Existen diferentes técnicas para realizar la búsqueda, muchas de ellas utilizadas en los sistemas de recuperación de información, la más utilizada para la búsqueda en dominios no restringidos se basa en la reformulación de la pregunta. Por ejemplo cuando se busca la respuesta a la pregunta “¿Cuándo nació Artigas?”, sabiendo que la respuesta podría tener la forma “Artigas nació en <FECHA>” o “En <FECHA>nació Artigas”, el sistema de BR puede buscar esas formulaciones en los documentos e instanciar <FECHA>donde corresponda. [12]

Dependiendo si el sistema se basa en obtener la respuesta de un grupo de documentos específicos, o de la Web, se utilizan técnicas más o menos complejas, esto se debe principalmente a los tiempos de respuesta y procesamiento que se requieren. Por ejemplo si se aprovecha la Web, dada una pregunta de la forma:

¿Dónde está p1 p2 ... pn?

se podrían generar las siguientes consultas, simplemente permutando las palabras de la pregunta:

P1 está p2 ... pn

P1 p2 está ... pn

...

El motor de búsqueda correspondiente buscará y retornará documentos que contengan estas frases. La idea aquí es que para las frases que no tienen sentido no se encuentre nada, y que para las frases bien formuladas en términos del lenguaje retorne los resultados correctos. Una suposición importante es que dado que la Web es un repositorio de documentos muy grande, seguramente en algún lugar se encontrará la respuesta formulada en una de estas variantes. Esta es una afirmación que no se puede hacer si la búsqueda se realiza en un documento específico, en donde la probabilidad de encontrar una frase que coincida exactamente con las reformulaciones anteriores es muy reducida.

Un ejemplo es el sistema de búsqueda de respuestas AskMSR [2], el cual utiliza este método tomando la Web como fuente de información. Dada una pregunta, genera una variedad de reformulaciones basadas en la reescritura de la pregunta inicial, con un peso asociado, para luego enviar las consultas al motor de búsqueda. Para esta reformulación no se utiliza ningún *parser* o *part-of-speech tagger*, simplemente se basa en la manipulación de las palabras de la pregunta inicial. Las reglas para la reescritura y asignación de pesos las crearon de forma manual, formulando ciertos patrones.

La formulación de patrones [16] para los distintos tipos de pregunta es una técnica que consiste en buscar en la pregunta la presencia de palabras específicas, etiquetas gramaticales y expresiones regulares, para luego poder definir a qué patrón corresponde y de esa forma obtener el formato de las posibles respuestas. Un ejemplo de patrón es el que se encuentra en el cuadro 2.1.

Patrón	Ejemplo
<i>¿Dónde VERBO cualquier seq. de caracteres?</i>	<i>¿Dónde nació Artigas?</i>
<i>cualquier seq. de caracteres VERBO en</i>	<i>Artigas nació en Uruguay.</i>
<i>en VERBO cualquier seq. de caracteres</i>	<i>En Uruguay nació Artigas.</i>

Cuadro 2.1: Ejemplo de patrón.

Una forma de crear estos patrones es utilizando métodos de aprendizaje automático. De esta manera se pueden aprovechar las características léxico-sintácticas, tanto de la pregunta como de los fragmentos donde se encuentra la respuesta, para entrenar a un clasificador y que éste aprenda de manera automática las reglas que determinen cuál es la respuesta correcta a la pregunta planteada. Esto evita el trabajo de generar reglas de manera manual al observar grandes conjuntos de instancias pregunta-respuesta.

Haciendo una comparación solamente a nivel de términos, es fácil encontrar casos en los que el sistema descarta documentos muy relevantes que contienen la respuesta por estar expresada en términos diferentes a los empleados en la pregunta. Se da principalmente cuando la búsqueda de respuestas es sobre un documento específico. Dentro de las soluciones posibles se enmarca la *expansión de la pregunta* [12]. Este proceso consiste en añadir al conjunto de términos originales de la pregunta, aquellos otros términos que pueden utilizarse para expresar las mismas ideas o conceptos. El objetivo es mejorar las preguntas iniciales formuladas por los usuarios para minimizar el número de documentos relevantes descartados. Existen diferentes métodos de selección de términos a incorporar a la pregunta, desde la búsqueda de sinónimos, hipónimos, hiperónimos hasta la realización de un análisis morfológico de las palabras. Un importante recurso léxico que es usado en muchos sistemas de búsqueda de respuestas es WordNet[25] y sus extensiones como EuroWordNet[10]. Este último incluye otros idiomas aparte del inglés como el español, francés, italiano, etc.

Citando un ejemplo, un sistema de búsqueda de respuestas en la Web que utiliza la expansión de preguntas es WebClopedia [14]. A diferencia del sistema AskMSR, utiliza técnicas más complejas de PLN para analizar la pregunta e identificar las respuestas. El principal problema de aplicar técnicas sofisticadas de PLN en la Web es lograr un buen grado de eficiencia, dado la cantidad de documentos que hay que manejar. WebClopedia utiliza

WordNet para expandir las preguntas con el objetivo de enviar consultas más completas al motor de búsqueda. Dependiendo de los resultados obtenidos, se puede expandir aún más o reducir las consultas.

Otra forma de enriquecer la pregunta es mediante la **realimentación** (*relevance feedback*) [12]. Esta técnica consiste en enriquecer la pregunta inicial realizada por el usuario del sistema, mediante la utilización de información relevante obtenida de documentos que se hayan recuperado utilizando exclusivamente la pregunta inicial. Esta información se añade a la pregunta, complementando la información que ésta contiene y facilitando la detección de nuevos documentos relevantes en búsquedas posteriores.

Otro ejemplo es AnswerBus [27], un sistema de búsqueda de respuestas, de dominio abierto, que se basa en búsqueda de documentos en la Web. Una de sus particularidades es que la pregunta se puede hacer en distintos idiomas (inglés, alemán, francés, español, italiano y portugués), pero la respuesta es siempre en inglés. Como parte de la etapa de análisis de la pregunta, si la pregunta no está en inglés, AnswerBus la envía al traductor de AltaVista BabelFish el cual la devuelve en inglés. Luego, ya en la etapa de recuperación de documentos, cinco motores de búsqueda son usados para recuperar los documentos de la Web: Google, Yahoo, WiseNut, AltaVista y Yahoo News. Dada una pregunta, según el tipo que sea, el sistema se encarga de seleccionar dos o tres motores de búsqueda que considera que se obtendrán mejores resultados. Otra de sus particularidades es que toma la performance como una prioridad. Es por eso que en esta etapa no se trata de encontrar la mejor consulta para enviar al motor de búsqueda, por ejemplo con la expansión de preguntas, sino que se busca una consulta lo suficientemente buena pero ponderando la calidad de los resultados y la velocidad con que son devueltos.

Tres técnicas son utilizadas y combinadas para formar consultas sencillas. La primera de ellas es la *eliminación de palabras funcionales*, entendiéndose por palabras funcionales, las preposiciones, los artículos, los pronombres y las conjunciones. La segunda es la *utilización de una tabla de frecuencia de palabras*, cuya idea básica es que las palabras más comunes del lenguaje son menos discriminatorias que las demás. AnswerBus por ejemplo, elimina de la consulta a aquellas palabras que se encuentran en su tabla de palabras frecuentes. Para finalizar, la tercer técnica es la de *modificación de palabras*, donde se convierten algunas palabras de la consulta original. Por ejemplo: “¿Cuándo fue el fin del período Jurásico?” , la palabra *fue el fin* se convierte a *finalizó*: “El período Jurásico finalizó”.

2.2.3. Selección de pasajes relevantes

Una vez que se tienen los documentos relacionados a la pregunta, se buscan los pasajes relevantes y en base al grado de relación que tengan con la pregunta se filtran o se ponderan. Esta selección se centra en la reducción de la cantidad de texto que se va a tener que procesar en la fase de extracción de respuestas. Una forma simple de obtener estos pasajes es en función de la aparición de los términos clave de la pregunta dentro de ellos. De todas formas, que aparezcan, no significa que tengan alguna relación con la pregunta, ya que pueden referirse a conceptos diferentes. Se puede dar el caso que aparezcan pero sin tener ninguna conexión entre ellos, por eso es bueno definir una forma de ponderación de los pasajes. Los que tengan mayor puntaje serán los tenidos en cuenta en primer lugar.

En el caso de búsquedas en la Web, dada la gran cantidad de documentos que pueden ser recuperados, se puede realizar un ranking previo a la selección de los pasajes. Esto es para reducir el número de documentos a procesar. El sistema de búsqueda de respuestas WebClopedia utiliza este método [14]. Primero pondera los documentos que son recuperados de la Web y luego, de los que obtuvieron mayor puntaje, selecciona y pondera los pasajes relevantes. *AnswerBus* utiliza un método similar [27], de los mejores documentos retornados por los distintos motores de búsqueda, primero realiza un análisis sintáctico y los divide en oraciones, y luego determina si esa oración es una respuesta candidata. Para esto, a cada una le asigna un puntaje en base a la cantidad de palabras de la consulta que figuran en la oración.

2.2.4. Extracción de Respuestas

Para la mayoría de los sistemas de búsqueda de respuestas, el proceso de extracción de las respuestas es la etapa final. Este proceso analiza los párrafos relevantes, obtenidos del proceso anterior, con la finalidad de localizar aquellos extractos reducidos de texto que el sistema considera que contienen la respuesta a la pregunta. Cada sistema en particular utiliza diversos métodos y técnicas en esta etapa, pero se pueden diferenciar ciertas sub etapas significativas dentro de ella:

- **Detección de las respuestas posibles.** Se revisa cada párrafo relevante con la intención de seleccionar aquellas estructuras sintácticas que pueden ser la respuesta a la pregunta. Se descartan aquellos conceptos que no pueden ser respuestas a la pregunta.
- **Valoración de las respuestas posibles.** Cada una de las respuestas

posibles detectadas en los párrafos relevantes se puntúa con la intención de valorar en que medida puede o no ser una respuesta correcta.

- **Ordenación de las respuestas según el valor asignado.**

WebClopedia [14] primero realiza un análisis sintáctico de la pregunta y le da una interpretación semántica, luego hace lo mismo con las respuestas candidatas para retornar la respuesta que mejor se corresponda a ella. Para esto utiliza una lista de patrones con tipos de preguntas y sus tipos de respuestas asociadas.

AskMSR [2], aplica un filtro a los n-gramas y les asigna un nuevo puntaje, acorde a cuan relacionado está con la pregunta. Un n-grama es una subsecuencia de n ítems de una secuencia dada, en este caso sería una subsecuencia de n palabras. Basado en el tipo de pregunta, el sistema determina que colección de filtros aplicar al conjunto de posibles respuestas. Estos filtros consisten en determinados patrones de expresiones regulares, la mayoría hechos a mano en base al conocimiento humano de los tipos de pregunta. Algunos, un poco más sofisticados, realizan un análisis semántico utilizando otras herramientas del tipo *part-of-speech taggers*. Estos pueden indicar por ejemplo que el string “Juan Pablo II” se refiere a una persona o “Montevideo” a una ciudad. Los filtros seleccionados son aplicados a cada n-grama candidato y usados para ajustar el puntaje que se les había asignado anteriormente y en algunos casos removerlos de la lista de candidatos.

AnswerBus [27] lo primero que hace es descartar las oraciones que obtuvieron puntaje nulo en el paso anterior; sobre las otras aplica técnicas para refinar el puntaje inicial. Una de ellas consiste en relacionar palabras de la pregunta con palabras de las oraciones candidatas (por ejemplo la palabra distancia la relaciona con la palabras “millas”, “kilómetros”, etc.). Para determinadas preguntas también utiliza etiquetado de entidades para seleccionar oraciones que contengan el tipo de respuesta esperado. Otra técnica utilizada es la resolución de correferencias (por ejemplo si aparecen palabras como “él” resuelve que se refiere a una persona descrita en otra parte del documento) que le permite, por ejemplo, vincular oraciones con un sujeto referido en la pregunta.

2.2.5. Formulación de respuestas

En esta etapa, en general la mayoría de los sistemas obtienen la o las respuestas mejor valoradas en el proceso anterior y las retornan como respuestas a la pregunta. *WebClopedia* y *AnswerBus* lo hacen de esta manera. Sin embargo algunos sistemas profundizan aun más y en base al tipo de

la pregunta y la respuesta, forman en lenguaje natural la respuesta para presentarle al usuario.

El sistema *AskMSR* [2] hace una superposición de las respuestas similares construyendo n-gramas más largos a partir de n-gramas más cortos. De esta forma se obtienen respuestas mas largas que no se consiguen con el resultado de la etapa anterior y se restringe la cantidad de respuestas posibles pues, se fusionan varias respuestas candidatas en una respuesta mas larga.

	Análisis de la pregunta	Recuperación de documentos	Selección de pasajes relevantes	Extracción de respuestas	Formulación de respuestas
AskMSR	Asignación de uno de siete tipos de preguntas(who-question, what-question, etc.).	Permutación de palabras de la pregunta junto con el uso de patrones.	Recolección de n-gramas.	Reponderación de los n-gramas, basándose en la pregunta y utilización de filtros predefinidos.	Superposición de n-gramas.
WebClopedia	Uso de una topología de preguntas-respuestas.	Expansión de la pregunta.	Ponderación de documentos y pasajes.	Reponderación de pasajes basándose en la pregunta y la utilización de patrones	Pasajes con mayor puntaje.
AnswerBus	Uso de un diccionario para determinar tipo de pregunta y clasificación de la respuesta esperada (persona, fecha, etc.).	Expansión de la pregunta, modificación de palabras, eliminación de palabras funcionales, utilización de más de un buscador.	Ponderación de documentos y pasajes.	Reponderación de pasajes basándose en la relación entre las palabras de la pregunta y las del pasaje.	Pasajes con mayor puntaje.

Cuadro 2.2: Tabla comparativa de sistemas de BR.

Capítulo 3

El sistema WebQA

Como se mencionó en la sección 1.1, el principal objetivo de este proyecto es el desarrollo de un sistema de búsqueda de respuestas. El idioma establecido es el español y la fuente de información la Web.

Las preguntas que se responden son las fácticas. Éstas requieren una única respuesta correcta y concreta (cuándo, dónde, quién, etc.). Sin embargo, existen otro tipo de preguntas más complejas que no se abordan en este proyecto. Algunos ejemplos son: las de recopilación de información, (ej: “¿Con que países limita Uruguay?”), las que requieren de un cierto análisis profesional, (ej: “¿Cuál es el problema en el desarrollo de las economías sudamericanas?”), las que son extremadamente ambiguas donde pueden ser válidas más de una respuesta dependiendo del contexto y que es imposible determinar exactamente qué es lo que quiere el usuario (ej.: “¿Por qué se produjo la guerra de las Malvinas?”). Para responderlas se requeriría otro tipo de análisis: recuperación y síntesis de información obtenida de diferentes fuentes, métodos para la toma de decisiones, interacción del sistema con el usuario de modo de lograr una respuesta conjunta, etc [12].

La información en la Web, en muchos casos, puede no ser fiable. Sin embargo, se busca aprovechar la redundancia de la Web para obtener respuestas confiables (ya que están repetidas en distintos lugares). Además evita la necesidad de utilizar recursos léxicos importantes.

WebQA cumple con las etapas clásicas de este tipo de sistemas: análisis de la pregunta, recuperación de documentos, selección de pasajes relevantes y extracción de la respuesta (figura 3.1).

Con el *análisis de la pregunta*, a partir de la pregunta que formula el usuario, se busca obtener información importante de ésta (como por ejemplo el tipo de respuesta esperado), para luego ser utilizada en las etapas sucesivas

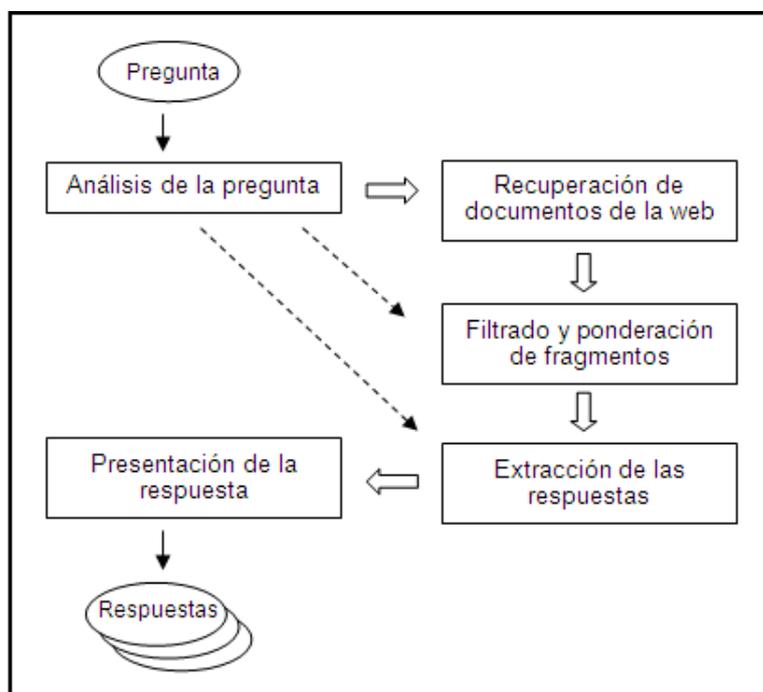


Figura 3.1: Etapas de WebQA.

con el objetivo final de retornar una respuesta correcta.

En la etapa de *recuperación de documentos*, se obtiene información relacionada a la pregunta de la Web. Para esto se reformula pregunta de modo de enviar consultas a un motor de búsqueda. Estas reformulaciones se basan exclusivamente en la manipulación de las palabras de la pregunta, y no se utiliza ningún recurso léxico para determinar sinónimos, homónimos, hiperónimos, etc. La suposición es que en la respuesta estarán todas o parte de las palabras de la pregunta.

En la *selección de pasajes relevantes* (en WebQA esta etapa es denominada “filtrado y ponderación de fragmentos”) se analizan los fragmentos que retorna el buscador como resúmenes de los documentos, con el fin de filtrar los que se consideren que no contienen la respuesta y ponderar el resto.

Tanto para el análisis de la pregunta como de los fragmentos, se utilizan recursos léxicos para el análisis morfológico de las palabras.

Para la última etapa de *extracción de la respuesta*, se aplica la técnica de análisis de ngramas, basada en el trabajo de Brill [2]. Con esta técnica se busca obtener una respuesta concreta, tomando como suposición que se repite en la mayoría de los fragmentos obtenidos de la etapa anterior.

Finalmente se retorna un conjunto de palabras de longitud n (*ngrama*).

Al usuario se le presentan como posibles respuestas la lista de fragmentos ordenados según su ponderación final, con la respuesta concreta marcada. Si el tipo de respuesta no fuera algo concreto, por ejemplo una descripción, se presentan los fragmentos ordenados, pero sin marcar la respuesta.

En las secciones siguientes se describe como se resolvió cada etapa y los algoritmos utilizados. También, a modo de ejemplo, se ilustra cada paso con la pregunta “¿Dónde está ubicado Cabo Polonio?”, hasta llegar a la respuesta retornada por el sistema.

3.1. Análisis de la pregunta

Dada una pregunta, se debe determinar de qué tipo es y qué se quiere como respuesta (por ejemplo: fecha, lugar, nombre, etc). Esta información es utilizada en etapas posteriores, para acotar los fragmentos a analizar y ubicar la respuesta exacta. También es necesario identificar las palabras que no son de ninguna utilidad al momento de buscar información en la Web, por no tener importancia por su frecuencia y por no aportar significado (por ejemplo: el, la, ellas, ellos, etc.).

Por esta razón, a cada palabra se le asigna un peso que indica cuan frecuente es. Para esto, se hace un análisis morfológico de la pregunta, determinando la categoría de cada palabra (nombres, verbos, adjetivos, etc.) El peso depende de la categoría, y está preestablecido. Por ejemplo las preposiciones, interjecciones, determinantes y signos de puntuación tendrán un peso significativamente menor que un nombre propio, o una fecha. Toda esta información será utilizada luego por los distintos reformuladores en la etapa de recuperación de documentos de la Web (punto 3.2).

El problema mayor que se debió enfrentar es que para el español existen muchísimas maneras de formular los distintos tipos de preguntas. Para el ejemplo que se irá analizando en cada etapa (“¿Dónde está ubicado Cabo Polonio?”), posibles formulaciones válidas serían “¿En qué lugar está ubicado Cabo Polonio?”, “¿En qué sitio está ubicado Cabo Polonio?”, “¿Cuál es el lugar dónde está ubicado Cabo Polonio?”, entre otras. Para resolverlo, se buscó un mecanismo que pueda reconocer la mayor cantidad posible de formulaciones de preguntas, su tipo y el tipo de respuesta que esperan, y que sea simple de incrementar y actualizar con nuevas formulaciones a ser reconocidas. La solución fue establecer una serie de patrones. Esta es una técnica que consiste en buscar en la pregunta la presencia de palabras específicas y etiquetas gramaticales, para luego poder definir a que patrón

corresponde y de esa forma obtener el tipo de la pregunta y el tipo de respuesta esperado ([16], [14]).

En el patrón se definen dos puntos importantes: primero una composición de palabras y etiquetas gramaticales que deberán coincidir con la formulación de la pregunta, y segundo la información que se obtendrá si esto último se cumple. Esta información inicialmente es el tipo de la pregunta y el tipo de respuesta esperado. En el cuadro 3.1 se muestran ejemplos.

Para las preguntas que empiezan con “Dónde”, se espera una ubicación como respuesta que generalmente es un lugar. Esto se puede especificar en un patrón. También existen otro tipo de formulaciones en las que también se espera una lugar como respuesta, por ejemplo las que empiezan con “En que (lugar | sitio | país | ciudad | ...)”, entre otras. A partir de aquí se puede definir otro patrón donde el tipo de pregunta sea “Dónde” y el tipo de respuesta esperado sea “lugar”.

Otros ejemplos son las preguntas que empiezan con “Cuándo” o “Cuántos”, donde también se desprende directamente el tipo de respuesta que esperan (“fecha” y “cantidad” respectivamente).

Las preguntas que empiezan con “Quién” no son tan triviales, ya que se podría esperar una persona o una descripción como respuesta (Ej.: “¿Quién fue el primer hombre en pisar la Luna?” y “¿Quién fue Neil Amstrong?” respectivamente). Observando distintas formulaciones de este tipo de preguntas, se pudo concluir que cuando se espera una descripción se cumple el patrón de que a la palabra “Quién” le sigue generalmente el verbo “es” (cualquiera de sus conjugaciones) y a continuación un nombre propio. En el resto de los casos generalmente se espera una persona como respuesta.

Formulación	Patrón		Ejemplos
	Tipo pregunta	Tipo respuesta	
Dónde	“Dónde”	“Lugar”	¿Dónde está ubicado Cabo Polonio?
Quién (es fue era) <Nombre Propio>	“Quién”	“Descripción”	¿Quién es Bill Gates?
Quién	“Quién”	“Persona”	¿Quién descubrió América?

Cuadro 3.1: Ejemplo de formulación de patrones.

Se podría manejar un criterio de especificidad en el tipo de respuesta esperado (por ejemplo, para lugar diferenciar entre país, ciudad, etc.). Esto

no se hizo en este proyecto, ya que, como se verá en los capítulos siguientes, no se utiliza ningún diccionario ni otro recurso léxico para determinar la especificidad de las palabras en el momento de la búsqueda de la respuesta. Su ubicación está basada en su repetición en los distintos resultados obtenidos de la Web.

Para las preguntas que no son fácticas, por ejemplo del estilo “¿Por qué . . .?”, se enmarca en el patrón que se espera como respuesta una descripción. En los casos que para una formulación no exista ningún patrón que se corresponda, se prefiere responder algo relacionado con la pregunta a no responder nada. Es por esta razón que se incluye el tipo de pregunta y respuesta “indefinido”, que se maneja de manera especial en etapas posteriores.

En total se definieron 25 patrones, de los cuales 10 corresponden a las preguntas del tipo *Cuándo*, 5 del tipo *Dónde*, 5 para el tipo *Cuántos*, 3 del tipo *Quién*, y por último 2 que no están incluidas en ninguno de estos casos, en los que se espera un nombre común como respuesta (por ejemplo: ¿Qué deporte predomina en Uruguay?). Esta lista de patrones es extensible, de modo de soportar otras formulaciones.

3.2. Recuperación de documentos de la Web

Luego del análisis de la pregunta, es necesario obtener documentos de la Web para analizar. Estos son extraídos por medio de un motor de búsqueda. Para ello, a partir de las palabras de la pregunta y del análisis efectuado en la etapa anterior, se definen un conjunto de reformulaciones, que serán las consultas a ser enviadas por el motor de búsqueda. Dado que el análisis de los documentos completos implica un alto tiempo de procesamiento, se obtienen los fragmentos retornados por el motor de búsqueda. Sobre ellos no se realiza ningún análisis para determinar si pueden o no contener la respuesta, esto se hará en etapas posteriores.

Existen muchísimas maneras de reformular una pregunta, pero hay reformulaciones que son mejores que otras. Es más probable que los resultados recuperados a partir de las reformulaciones más específicas contengan la respuesta. Sin embargo, generalmente se logra una cantidad de resultados muy reducida, pudiéndose dar el caso de no obtener ninguno. Ante esta situación es necesario aplicar una reformulación más general (por ejemplo conjunción de las palabras clave) de modo de conseguir un mayor número de resultados para analizar. Como contrapartida, muchos contienen estas palabras en un contexto diferente al de la pregunta, siendo irrelevantes a la hora de extraer la respuesta. Por ejemplo para la pregunta “¿Dónde está ubicado Cabo Polonio?”, se espera como respuesta algo del estilo “Cabo Polonio es-

tá ubicado en ...”. Una reformulación específica sería <“Cabo Polonio está ubicado”>, y otra más general <“está” + “ubicado” + “Cabo Polonio”>. Para la primera se buscan fragmentos que contienen la frase “Cabo Polonio está ubicado”, por lo que es muy probable que contengan la respuesta, en cambio para la segunda los fragmentos obtenidos tendrán que incluir las palabras “está”, “ubicado” y “Cabo Polonio” de manera aislada.

En el sistema WebQA se implementaron tres técnicas para formular las consultas:

1. Según *patrones* establecidos.
2. Permutación de las dos primeras palabras de la forma declarativa de la pregunta.
3. Conjunción de palabras clave de la pregunta.

A continuación se explicará en detalle cada reformulador implementado.

3.2.1. Reformulador según patrones establecidos

Dada una pregunta, en el momento de definir cuales documentos son interesantes para recuperar de la Web, por tener altas posibilidades de contener la respuesta, algunas interrogantes planteadas fueron: ¿Cómo debería estar formulada la respuesta? ¿Qué relación existe entre la manera en que está formulada la pregunta y la manera en que se formula la respuesta? ¿Cómo se puede aprovechar esto?

Analizando distintos ejemplos de preguntas y respuestas, se llegó a la conclusión de que la formulación de la respuesta se puede lograr a partir de la forma declarativa de la pregunta, combinándola con palabras que refieren a lo que se espera como respuesta (ej.: año, día, país, etc.). Esto permite definir una serie de patrones asociados a un conjunto de reformulaciones. Esta técnica también es utilizada por otros sistemas de búsqueda de respuestas, como por ejemplo AskMSR [3] y WebClopedia [14].

Se reformulará la pregunta que se viene siguiendo como ejemplo para que quede más claro el concepto: “¿En que departamento está ubicado Cabo Polonio?”. Aquí se podría esperar como respuesta “en el departamento ... está ubicado Cabo Polonio” o “Cabo Polonio está ubicado en el departamento ...”, entre otras. Por lo tanto posibles reformulaciones para recuperar documentos que contengan esas frases son:

- “departamento” + “está ubicado Cabo Polonio”

- “departamento” + “Cabo Polonio está ubicado”
- “departamento” + “está ubicado” + “Cabo Polonio”

La última reformulación es más general, y con ella se busca contemplar los casos en que la respuesta se encuentre dentro de un cierto contexto (Ej.: “Cabo Polonio <cualquier_conjunto_de_palabras>. Está ubicado en el departamento ...”).

Patrón	Ejemplo
“EN QUE”	¿En que departamento está ubicado Cabo Polonio?
X + Y V	“departamento” + “Cabo Polonio está ubicado”
X + V Y	“departamento” + “está ubicado Cabo Polonio”
X + V + Y	“departamento” + “está ubicado” + “Cabo Polonio”
“CUANTOS”	¿Cuántos días tiene un año bisiesto?
X V Y	“días tiene un año bisiesto”
X + Y V	“días” + “un año bisiesto tiene”
X + V Y	“días” + “tiene un año bisiesto”
X + V + Y	“días” + “tiene” + “un año bisiesto”

Cuadro 3.2: Ejemplos de patrones para reformulación de la pregunta, donde: X = secuencia de palabras anterior al verbo de la pregunta; V = verbo de la pregunta; Y = secuencia de palabras posterior al verbo de la pregunta.

En definitiva, existe una fuerte relación entre la formulación de la pregunta y la formulación de la respuesta, lo que permite establecer un conjunto de patrones. Si se observa en el ejemplo, es posible lograr las reformulaciones dividiendo y manipulando tres partes de la pregunta (“departamento”, “está ubicado”, y “Cabo Polonio”). Por un lado se tiene el verbo, a veces junto con un verbo auxiliar como es en este caso “está”, por otro el conjunto de palabras que está antes del verbo y finalmente las palabras que vienen después. Ésta fue la manera en que fueron definidos los patrones, y las reformulaciones que corresponden a cada uno de ellos. La correspondencia entre una pregunta y un patrón se determina con la coincidencia de sus palabras iniciales (en el ejemplo: “EN QUE”). En el cuadro 3.2 se muestran ejemplos.

3.2.2. Reformulador en base a permutación de dos primeras palabras

Esta técnica está basada en el trabajo de Luis Villaseñor “*Búsqueda de Respuestas basada en Redundancia*” [20]. Consiste en tomar la pregunta, quitando el pronombre interrogativo o palabras iniciales que determinan su tipo (Ej.: “quién”, “dónde”, “en qué año”, “a qué hora”, etc.), y realizarle ciertas permutaciones de las dos primeras palabras. Estas dos palabras generalmente se componen del verbo, y un pronombre o un verbo auxiliar, de modo que manipulándolas se puede lograr escribir la pregunta en distintas formas declarativas.

Las permutaciones realizadas son las siguientes:

1. Todas las palabras
2. Se quita la primera palabra
3. Se quita la primera y segunda palabra
4. Primera palabra al final
5. Primera y segunda palabra al final

Volviendo al ejemplo, para la pregunta “¿Dónde está ubicado Cabo Polonio?”, quitando el pronombre interrogativo queda “está ubicado Cabo Polonio”, por lo tanto las dos palabras a manipular son “está” y “ubicado”, que son un verbo auxiliar y verbo de la pregunta respectivamente. Finalmente se obtienen las siguientes reformulaciones:

1. “**está ubicado** Cabo Polonio”
2. “**ubicado** Cabo Polonio”
3. “Cabo Polonio”
4. “**ubicado** Cabo Polonio **está**”
5. “Cabo Polonio **está ubicado**”

Dado que en esta técnica no se utiliza ningún recurso lingüístico, es posible que alguna reformulación quede con poco o sin sentido gramatical (en este ejemplo, la cuarta). Esto no influye negativamente en el momento de extraer la respuesta en etapas posteriores, ya que al estar mal formulada

la consulta, no se encontrará en ningún documento que la contenga, por lo que el buscador no retornará resultados.

Vale la pena destacar que es posible recuperar extractos que tengan sinónimos verbales. En el ejemplo, con la tercera reformulación se podrían obtener resultados que incluyan un sinónimo de “está ubicado”: “Cabo Polonio” <“se encuentra”, “está situado”, etc.>.

3.2.3. Reformulador en base a conjunción de palabras clave

Con los dos reformuladores anteriores se obtienen documentos con gran probabilidad de contener la respuesta. El problema que presentan es que para ciertas preguntas, especialmente las largas, se obtienen pocos o ningún resultado. Por esta razón se decidió incluir un reformulador que genera reformulaciones más generales, para asegurar la obtención de resultados a analizar.

El reformulador genera dos consultas, que se nombrarán como máxima y mínima. La reformulación máxima se compone de la conjunción de todas las palabras de la pregunta, sin tomar en cuenta las muy frecuentes. La reformulación mínima se compone únicamente del verbo, nombres propios y fechas que ocurran en la pregunta. Existen casos donde la reformulación máxima y mínima coinciden, y por lo tanto se retorna una única consulta.

En el ejemplo se ilustrará el procedimiento:

Del resultado del análisis de la pregunta “¿Dónde está ubicado Cabo Polonio?” en la primera etapa, se puede obtener que su forma declarativa es “está ubicado Cabo Polonio” y el análisis morfológico se encuentra en el cuadro 3.3.

Token	está	ubicado	Cabo_Polonio
Etiqueta	Verbo	Verbo	Nombre Propio

Cuadro 3.3: Análisis morfológico de la frase “está ubicado Cabo Polonio”.

Como se puede apreciar, la forma declarativa de la pregunta se compone solamente de verbos y nombres propios, por lo que la reformulación máxima y mínima coinciden. Por lo tanto la única reformulación resultante es:

“está” + “ubicado” + “Cabo Polonio”

Un caso en que la reformulación máxima y mínima no coincide es por ejemplo para la pregunta “¿Cuándo se jugaron los juegos olímpicos de Atlanta?”. El verbo de la pregunta es “jugar”, y además está el nombre propio

“Atlanta”. Por lo tanto la reformulación máxima y mínima quedaría respectivamente de la siguiente manera: <“jugaron” + “juegos” + “olímpicos” + “Atlanta”>y <“jugaron” + “Atlanta”>.

Como se dijo anteriormente, este reformulador aumenta las posibilidades de encontrar resultados, pero como contrapartida, al ser las reformulaciones más generales, es muy probable que devuelva fragmentos irrelevantes.

3.3. Filtrado y ponderación de fragmentos

El objetivo de esta etapa es analizar los fragmentos obtenidos de la Web de modo de filtrarlos y ponderarlos. Para esto, al igual que en el análisis de la pregunta, antes que nada es necesario realizar un análisis morfológico por cada fragmento. Este análisis retorna el texto etiquetado identificando verbos, nombres propios, fechas, etc., denominados *tokens* (por ejemplo las palabras: “Cabo Polonio” son reconocidas como un token del tipo nombre propio).

Si un fragmento no tiene ningún token que sea del tipo de respuesta esperado, inmediatamente se puede determinar que no es una posible respuesta. En el filtrado se descartan estos fragmentos.

El resto de los fragmentos contendrán algún token que sea del tipo de respuesta esperado, lo que no significa que tengan la respuesta. La ponderación busca determinar en cuáles es más factible que esté la respuesta.

Por ejemplo, para la pregunta “¿Dónde está ubicada la ciudad de Montevideo?”, una buena respuesta sería “La ciudad de Montevideo está ubicada en Uruguay”. A partir de aquí es posible desprender algunas conclusiones: cuanto más tokens de la pregunta haya en un fragmento, probablemente estará más relacionado a lo que se quiere como respuesta. A su vez, cuanto más juntos estén entre sí, será mejor. Y mejor aun si están cerca de un token del tipo de respuesta esperado. Otra cosa importante es que con las reformulaciones más naturales de la pregunta habrá más posibilidades de que los fragmentos recuperados contengan la respuesta.

En este ejemplo, con el reformulador basado en la permutación de las dos primeras palabras (ver 3.2.2), se recuperan fragmentos con las reformulaciones “está ubicada la ciudad de Montevideo”, “la ciudad de Montevideo está ubicada”, entre otras. A partir de estas reformulaciones es muy probable que los fragmentos recuperados contengan la respuesta, pues son más naturales y más restrictivas. Sin embargo el reformulador basado en la conjunción de las tokens clave de la pregunta (ver 3.2.3), genera reformulaciones menos restrictivas, por ejemplo: <“está” + “ubicada” + “ciudad” + “Montevideo”>.

Con este reformulador se recuperan una mayor cantidad de fragmentos, pero no contienen las reformulaciones naturales de la pregunta. Por esta razón es importante tomar en cuenta con qué reformulación se obtuvo el fragmento.

A partir de estos supuestos, la ponderación de un fragmento se realiza básicamente a partir de la cantidad de tokens clave de la pregunta que éste tiene, considerando cuán cerca están entre ellos y cuán cerca están del token del tipo de respuesta esperado, influyendo también la reformulación con la cual se recuperó el fragmento.

Para resolverlo, siguiendo el trabajo de José Luis Vicedo González [12], se utiliza una ventana de análisis que se desplaza por los tokens del fragmento. Esta ventana tiene un cierto tamaño fijo que indica la cantidad de tokens que puede contener. El desplazamiento se realiza empezando por el primer token del fragmento y luego moviéndose de a un lugar, finalizando cuando la ventana incluye el último token. En cada desplazamiento se calcula la cantidad de tokens de la pregunta que se encuentran en la ventana (sin contar los pronombres interrogativos ni los tokens muy frecuentes por no tener importancia en la respuesta). Solamente se tomarán en cuenta las ventanas que tengan al menos un token del tipo de respuesta esperada y se llamará “mejor ventana” a la que contenga más tokens de la pregunta.

Finalmente, para el cálculo del puntaje de los fragmentos se aplica la fórmula del cuadro 3.4, que toma en cuenta lo siguiente: la cantidad de tokens de la pregunta que están en el fragmento, a partir de la mejor ventana de análisis cuán cerca están entre ellos y del tipo de respuesta esperado, y por último la reformulación con que fue obtenido el fragmento.

Sea:

- P = Puntaje final del fragmento
- F = Cantidad de tokens clave de la pregunta en el fragmento
- V = Cantidad de tokens clave de la pregunta en la mejor ventana
- $C1$ y $C2$ = Dos constantes
- R = Puntaje de la reformulación de la pregunta con la cual se obtuvo el fragmento.

La fórmula es: $P = (C1 * V + C2 * (F - V)) * R$

Cuadro 3.4: Fórmula para el cálculo del puntaje del fragmento

Como se puede apreciar en el cuadro 3.4, la fórmula suma dos resultados bien definidos:

1. $C1 * V$. La cantidad de tokens de la pregunta que están en la mejor ventana, multiplicado por una constante.

2. $C2^*(F-V)$. La cantidad de tokens de la pregunta que están en el fragmento pero quedaron fuera de la ventana, multiplicado por otra constante.

Del total de tokens clave de la pregunta en el fragmento, se tienen los que están incluidos en la mejor ventana y los que no (puntos 1 y 2 respectivamente). Las dos constantes ajustan estos valores, pudiéndose de esa forma definir y evaluar ambos casos con distintos niveles de prioridad. Dado que se desea priorizar que los tokens de la pregunta estén cerca entre sí en el fragmento, $C1$ será mayor que $C2$.

Continuando con el ejemplo:

Ante la pregunta “¿Dónde está ubicado Cabo Polonio?”, a partir de las reformulaciones establecidas en la etapa anterior, se obtienen entre otros los resultados siguientes (se marca con negrita los tokens que son del tipo de respuesta esperada, nombres propios en este caso, y subrayados los términos de la pregunta que se encuentran en el fragmento):

1. Cabo Polonio está ubicado en el sureste del **Departamento de Rocha** entre las playas **Valizas** y la **Pedrera** a unos 260 kilómetros de **Montevideo**. ...
2. Cabo Polonio es un balneario ubicado en el **Departamento de Rocha** , **Uruguay** Esta página fue modificada por última vez el 00 : 48 , 19 feb 2007. ...
3. El lugar esta ubicado en los **Alpes** italianos . Tiene la particularidad de que no llega luz ... **Uruguay** , más precisamente en las cercanías de Cabo Polonio. ...
4. Esta fortaleza ha sido escenario de múltiples invasiones y batallas por nuestra ... Este balneario ubicado a unos 10 km de **Castillos** por la Ruta 16 ...

El ejemplo se basará en las variables de la fórmula para el cálculo del puntaje del fragmento. Para éstas, supongamos la asignación de valores que se muestra en el cuadro 3.5. Los dos primeros fragmentos se obtienen con la reformulación R1, el tercero con la reformulación R2 y el último con la reformulación R3.

El cuadro 3.6 muestra los tokens de la pregunta que están en cada fragmento y los tokens del tipo de respuesta esperado detectados.

R1="Cabo Polonio está ubicado". Puntaje=1,5
 R2="Cabo Polonio" + "está" + "ubicado". Puntaje=1,2
 Constantes: C1=1 y C2=0,5
 Tamaño de la ventana: 8 tokens
 Tokens clave de la pregunta: ("Cabo Polonio", "está", "ubicado").
 Total de tokens calve: 3.

Cuadro 3.5: Ejemplo. Valores supuestos para las variables de la fórmula del cálculo del puntaje del fragmento.

Frag.	Tokens de la pregunta	Tokens del tipo de la respuesta esperada
1	{Cabo Polonio, está, ubicado}	{Departamento de Rocha, Valizas, Pedrera, Montevideo}
2	{Cabo Polonio, ubicado, Esta}	{Departamento de Rocha, Uruguay}
3	{está, ubicado, Cabo Polonio}	{Alpes, Uruguay}
4	{Esta, ubicado}	{Castillos}

Cuadro 3.6: Ejemplo. Tokens de la pregunta y del tipo de respuesta esperada detectados.

En el cuadro 3.7, se puede apreciar el análisis realizado para cada fragmento, asignando valores a las distintas variables. También se muestra la mejor ventana, o sea la ventana que tiene más tokens de la pregunta. En algunos casos puede haber más de una ventana con la misma cantidad máxima, situación que no afecta el puntaje final del fragmento, pero en el cuadro se muestra siempre la última ubicada.

Frag.	Mejor Ventana	F	V	R	Punt. final
1	<u>Cabo Polonio está ubicado</u> en el su- reste del Departamento de Rocha	3	3	1,5	$(3+3)*1,5 = 13,5$
2	<u>Cabo Polonio</u> es un balneario <u>ubicado</u> en el Departamento de Rocha	3	2	1	$(3+2)*1 = 5$
3	El lugar <u>esta ubicado</u> en los Alpes italianos	3	2	1	$(3+2)*1 = 5$
4	balneario <u>ubicado</u> a unos 10 km de Castillos	2	1	1	$(2+1)*1 = 3$

Cuadro 3.7: Ejemplo. Asignación de puntaje a cada fragmento.

En el cuadro 3.8 se muestran los fragmentos ordenados por su puntaje final. El que obtuvo el mayor puntaje fue el fragmento 1. Éste tiene los tres tokens clave de la pregunta ("Cabo Polonio", "está", "ubicado"), y fue

Frag.	Texto fragmento	Punt. final
1	Cabo Polonio está ubicado en el sureste del Departamento de Rocha entre las playas Valizas y la Pedrera a unos 260 kilómetros de Montevideo. ...	13,5
2	Cabo Polonio es un balneario ubicado en el Departamento de Rocha, Uruguay. ... Esta página fue modificada por última vez el 00 : 48 , 19 feb 2007. ...	5
3	El lugar está ubicado en los Alpes italianos . Tiene la particularidad de que no llega luz ... Uruguay, más precisamente en las cercanías de Cabo Polonio. ...	5
4	Esta fortaleza ha sido escenario de múltiples invasiones y batallas por nuestra ... Este balneario ubicado a unos 10 km de Castillos por la Ruta 16 ...	3

Cuadro 3.8: Ejemplo. Fragmentos ponderados y ordenados.

obtenido con la reformulación de mayor puntaje. Es importante observar que contiene la reformulación de la pregunta más natural de las cuatro, y a continuación le sigue la respuesta: “Cabo Polonio está ubicado en el sureste del Departamento de Rocha”.

Hay situaciones, en consultas de dos o más términos, en las que el motor de búsqueda retorna fragmentos que presentan enunciados separados por puntos suspensivos, extraídos de un mismo documento pero sin estar relacionados entre sí. En el ejemplo, se da en el fragmento 3, donde aparecen los tokens de la pregunta “está” y “ubicado” en el primer enunciado y “Cabo Polonio” en el segundo, sin tener ambos relación alguna. A su vez la mejor ventana quedó dentro del primer enunciado (“El lugar está ubicado en los Alpes italianos”), cuando es el segundo enunciado el que realmente tiene relación con lo que se está preguntando. De todas formas estas son situaciones particulares, que se pueden dar con las reformulaciones más generales, y que no afectan significativamente el orden final de los fragmentos.

3.4. Extracción de la respuesta mediante análisis de ngramas

En esta etapa se analizan los fragmentos con la finalidad de localizar aquellos extractos reducidos de texto que el sistema considera son la respuesta a la pregunta. Para esto se utiliza la técnica de análisis de ngramas. El objetivo de esta técnica es obtener una respuesta concreta, basándose en la suposición de que se repite en la mayoría de los fragmentos filtrados

y ponderados en la etapa anterior. Esta respuesta se retorna en forma de ngrama, o sea, un conjunto de palabras de longitud n . Este enfoque está basado en el trabajo realizado por Eric Brill [2], el sistema AskMSR, pero con algunas diferencias.

Básicamente se obtienen unigramas, bigramas y trigramas de los fragmentos y se les asigna un puntaje. Este puntaje es una valoración que toma en cuenta cuánto se repite en los distintos fragmentos y la ponderación de éstos. Todas las palabras que componen un ngrama tienen que ser del tipo de respuesta esperado.

El análisis de ngramas se aplica solo para las preguntas fácticas en las que se espera como respuesta una determinada palabra o un conjunto de palabras (ej: fecha, lugar, persona, etc.). Si la respuesta esperada fuera una descripción o esté indefinida, no se aplica este análisis.

La técnica se divide en tres etapas:

1. Recolección y ponderación de ngramas.
2. Solapamiento de ngramas.
3. Reponderación de fragmentos según el ngrama respuesta.

3.4.1. Recolección y ponderación de ngramas

En esta primera instancia, se analizan los fragmentos de modo de recolectar todos los unigramas, bigramas y trigramas candidatos a ser respuesta a la pregunta. La suposición fundamental es que, de todos los tokens del tipo de respuesta esperado, el que sea la respuesta a la pregunta será el que se repita más veces. A su vez se tienen en cuenta otros factores, como por ejemplo si el token está dentro de la ventana del fragmento y la ponderación del fragmento del cual fue obtenido. En los cuadros 3.9 y 3.10 se puede ver el pseudocódigo del algoritmo y la fórmula resultante para la ponderación de ngramas respectivamente.

La técnica recolecta hasta trigramas porque las respuestas buscadas son simples y concretas. De todas formas, luego con el solapamiento de ngramas, se podría llegar a formar un ngrama de longitud mayor que tres, aunque no es común que ocurra (en la evaluación del sistema no se dio el caso).

-
- a. Para cada fragmento:
 - b. Para cada unigrama, bigrama y trigramas del fragmento:
 1. Todas las palabras del ngrama tienen que ser del tipo de respuesta esperado, si no es así se descarta.
 2. A cada ngrama se le asigna un puntaje inicial parametrizable
 3. Según el largo del ngrama se le sumará un puntaje extra. Esto es para nivelar el puntaje de los ngramas más largos, por ser más difícil que se repitan.
 4. Por cada token del ngrama que pertenezca a la ventana del fragmento se suma un puntaje adicional. La idea aquí es priorizar las posibles respuestas que están cerca de las palabras de la pregunta.
 5. Para el puntaje final, se toma en cuenta el conocimiento adquirido en la etapa de filtrado y ponderación de fragmentos, multiplicando el puntaje del ngrama por el puntaje del fragmento al que pertenece.
 6. Finalmente, si no pertenece a la colección de ngramas analizados, se agrega junto con su puntaje. Si ya está en ésta, se suma el nuevo puntaje con el anterior.
 - c. Ordenar la colección de ngramas según los puntajes resultantes.
-

Cuadro 3.9: Algoritmo de recolección y ponderación de ngramas

Sean $P2$, $P3$, $P4$ y $P5$ los puntajes mencionados en el cuadro 3.9, puntos 2, 3, 4 y 5 respectivamente, y N el puntaje final de un ngrama, la fórmula resultante es la siguiente:

$$N = \sum_{i=1}^n ((P2 + \sum_{j=2}^m P3 + \sum_{k=1}^o P4) * P5)$$

donde:

- n = cantidad de repeticiones del ngrama
 - m = cantidad de palabras que componen el ngrama
 - o = cantidad de palabras del ngrama que están dentro de la ventana del fragmento
-

Cuadro 3.10: Fórmula para ponderación de ngramas

3.4.2. Solapamiento de NGramas

Luego del análisis anterior se tiene una lista ordenada de ngramas ponderados. El ngrama con más puntaje será la respuesta a la pregunta. A través del solapamiento de ngramas se busca retornar una respuesta más completa. Por ejemplo la respuesta “Rocha Uruguay” es mas completa que “Rocha”, o que “Uruguay”.

WebQA no maneja un criterio de especificidad explícitamente a la hora de generar la respuesta. Sin embargo el solapar ngramas puede traer como

resultado secundario que las respuestas sean más específicas. Por ejemplo, ante la pregunta anterior, si el ngrama de mayor puntaje fuera “Rocha”, luego del solapamiento podría quedar como respuesta “Rocha Uruguay”, la cual es más completa y también más específica.

El solapamiento de ngramas estuvo basado en el artículo de Brill sobre el sistema AskMSR [2]. La intención aquí es solapar el primer ngrama (el de mayor puntaje), el cual ya es la respuesta a la pregunta, con los que le siguen en puntaje, de forma de generar respuestas más largas y más completas. No se solaparán todos los ngramas obtenidos, sino solamente los de mayor puntaje. Esto es debido a que en el primer caso generalmente el resultado es un ngrama más largo, pero con palabras incluidas que no tienen relación alguna con la respuesta (ver apéndice evaluación de la variación de parámetros, A.1) En el cuadro 3.11 se puede ver el algoritmo.

Sean n_1, \dots, n_m los primeros m ngramas ordenados por su puntaje.

$i = 2$;

Mientras $(i < j)$ y $(puntaje(n_i) > (puntaje(n_1) * Pje)/100)$

$n_1 = solapar(n_1, n_i)$;

$i = i + 1$;

FinMientras

Donde:

- j , un parámetro de ajuste que determina hasta qué ngrama se intentará solapar con el primero, según el orden por puntaje.
 - $Pje.$, un parámetro de ajuste que indica que para qué se intenten solapar los ngramas n_1 con n_i , el puntaje de n_i , deberá ser mayor a $Pje\%$ del puntaje de n_1 .
 - “ $solapar(n_1, n_2)$ ”, realiza la concatenación de los ngramas n_1 y n_2 , si se cumple que las últimas palabras del primero coinciden con las primeras del segundo. Por ejemplo dados los ngramas AB y BCD, entonces $solapar('AB', 'BCD') = 'ABCD'$.
-

Cuadro 3.11: Algoritmo de solapamiento de ngramas

Como se puede ver en el cuadro 3.11, el algoritmo tiene dos condiciones basadas en los parámetros ‘ j ’ y ‘ Pje ’. El objetivo del porcentaje es solapar con el primer ngrama solamente los que tengan un puntaje lo suficientemente alto, por lo que también pueda estar contenido en la respuesta. El parámetro ‘ j ’, indica la cantidad de intentos máximos de solapamiento. Por ejemplo si este parámetro es igual a tres, entonces se intentará solapar el primer ngrama con el segundo y con el tercero (siempre que se cumpla la condición anterior). La razón de este tope es que se dan casos, especialmente para las preguntas que no hay mucha información en la Web, que la lista ordenada de ngramas tiene puntajes bajos y próximos entre sí. La condición del porcentaje se cumpliría en casi todos ellos y por lo tanto se intentaría solapar el primer ngrama con prácticamente todos los demás. El tope actúa de límite ante

este tipo de situaciones.

El ajuste de los valores de ambos parámetros se realizó empíricamente, por medio de pruebas sobre un conjunto de preguntas, y evaluando los resultados (por más detalles ver el apéndice evaluación de la variación de parámetros, A.1). Los valores fijados son: $j=3$ y $P_{je}=60\%$.

Se ilustrará el algoritmo continuando con la pregunta “¿Dónde está ubicado Cabo Polonio?”.

El cuadro 3.12, muestra los primeros ngramas, obtenidos en la etapa anterior, con sus respectivos puntajes.

NGRAMA	Puntaje
Rocha	63,0
Departamento Rocha	52,0
Rocha Uruguay	50,5
Pedrera	45,0
Departamento Rocha Uruguay	35,0
Paloma	33,0
Paloma Pedrera Servicios	33,0
Rocha Paloma Pedrera	29,0
...	...

Cuadro 3.12: Ngramas resultantes, con sus respectivos puntajes para la pregunta “¿Dónde está ubicado Cabo Polonio?”.

El unigrama “Rocha” obtuvo el mayor puntaje, por lo tanto, sin realizar el solapamiento, sería la respuesta. Luego se aplica el solapamiento, de modo de retornar una respuesta más completa.

Tomando los parámetros por defecto, se solapa el primer ngrama con los dos siguientes en puntaje si éstos últimos son superiores al 60% del puntaje del primero. Por lo tanto, los ngramas a solapar son: “Rocha”, “Departamento Rocha” y “Rocha Uruguay”.

1. “Rocha” + “Departamento Rocha” = “Departamento Rocha”
2. “Departamento Rocha” + “Rocha Uruguay” = “Departamento Rocha Uruguay”.

Finalmente la respuesta a la pregunta es “Departamento Rocha Uruguay”, siendo una respuesta más completa, y también más específica que “Rocha”.

3.4.3. Reponderación de fragmentos según ngrama respuesta

Luego del análisis de ngramas se vuelve a realizar otro ranking de los fragmentos en base a si contienen (total o parcialmente) la respuesta obtenida. Esto es, los fragmentos que contengan el ngrama completo quedarán en primer lugar. A estos le seguirán los que los contengan parcialmente y por último estarán los fragmentos en los que no figura el ngrama. En cada una de estas tres franjas se mantiene el orden según su puntaje obtenido en la etapa de filtrado y ponderación de fragmentos (pto. 3.3).

Para realizar esto se asigna un nuevo puntaje a los fragmentos según la siguiente formula:

*Nuevo puntaje = Puntaje actual * cantidad de tokens del ngrama en el fragmento / cantidad total de tokens del ngrama*

Así por ejemplo el fragmento:

*Cabo Polonio es un balneario ubicado en el **Departamento de Rocha, Uruguay.** . . .*

Quedará ubicado antes que:

*Cabo Polonio está ubicado en el sureste del **Departamento de Rocha** entre las playas Valizas y la Pedrera a unos 260 kilómetros de Montevideo.*

Puede ocurrir que un fragmento que tiene un puntaje alto descienda significativamente en el ranking por no tener la respuesta obtenida del análisis de ngramas. Este caso sería una indicación de que el fragmento analizado aisladamente tiene una alta probabilidad de contener la respuesta, pero basándose en la mayoría de los fragmentos no sería la correcta.

Resumiendo, un fragmento conserva su ponderación si contiene el ngrama respuesta completo, o la pierde en mayor o menor proporción según tenga el ngrama parcialmente o no lo contenga.

3.5. Presentación de la respuesta

Al usuario se le presentan como posibles respuestas los fragmentos resultantes ponderados y ordenados. En los casos en que se aplica la técnica de ngramas se marcan en negrita en el texto las palabras del ngrama respuesta. A su vez, si entre dos palabras a ser marcadas hay palabras auxiliares también se marcan.

De esta forma, para la pregunta “¿Dónde está ubicado Cabo Polonio?”, los dos primeros fragmentos que se presentarán como respuesta, con ésta marcada en negrita, son los siguientes:

1. *Cabo Polonio es un balneario ubicado en el **Departamento de Rocha, Uruguay**. . . .*
2. *Cabo Polonio está ubicado en el sureste del **Departamento de Rocha** entre las playas Valizas y la Pedrera a unos 260 kilómetros de Montevideo.*

La Pregunta: Donde está ubicado Cabo Polonio?

Posibles respuestas:

Imagen de Bar sobre Playa Sur de Cabo Polonio (enero 2005) Cabo Polonio es un balneario ubicado en el Departamento de Rocha , Uruguay Ver Cas Fragmento	
Fotos de Cabo Polonio . Departamento de Rocha . Uruguay . Ver Cas Fragmento	
Cabo Polonio está ubicado en la zona sureste de el departamento de Rocha entre las playas Valizas y la Pedrera a unos 260 kilómetros de Montevideo Ver Cas Fragmento	
Mayoral . com . uy , es un sitio dedicado el a turismo en Rocha , Uruguay . Visitando nos encontrara una completa guia de alojamiento , fotos , informacion en general , ... Ver Cas Fragmento	
Ubicado en la zona sureste de el Dpto. de Rocha , entre los balnearios Valizas y La Pedrera , Cabo Polonio se encuentra a 260 km. de Montevideo por ruta 9 , ... Ver Cas Fragmento	
Los barrí os de Punta de el Este , las zonas mas destacadas y las mejores casas y departamentos (apartamentos) de la ciudad de Punta de el Este en Uruguay . Ver Cas Fragmento	

Figura 3.2: Presentación de la respuesta del sistema WebQA.

En la figura 3.2 se puede observar la salida del sistema WebQA para la pregunta “¿Dónde está ubicado Cabo Polonio?”, mostrando los fragmentos ponderados y ordenados, con la respuesta marcada en negrita.

Capítulo 4

Implementación

Como se mencionó en el capítulo 2, los sistemas de BR están compuestos por varios módulos que cumplen una función determinada; de esta estructura es que surge la arquitectura de WebQA. El cometido principal que se busca además de que el sistema sea performante es el que de forma sencilla se puedan reemplazar módulos y modificar la secuencia de éstos, con el fin de mejorar la performance o simplemente realizar pruebas sobre la eficiencia de alguno de ellos. Es por ello que se decide utilizar como plataforma de desarrollo, la desarrollada por IBM llamada UIMA (Unstructured Information Management Architecture) [15].

UIMA es una arquitectura que se basa en componentes enfocados para el desarrollo en Java [18], cuyo fin es el de analizar, organizar y obtener información relevante para el usuario a partir de información no estructurada. Como ventaja de esta plataforma se tiene que la manipulación de módulos y flujos es muy sencilla, simplemente alcanza con modificar algunos XML de configuración para realizar cambios. En contrapartida la desventaja que posee es que al ser Java el lenguaje de programación, es necesario que todos los módulos que se utilicen sean desarrollados en Java, lo cual no es del todo conveniente debido a que la mayoría de las aplicaciones para procesamiento de lenguaje natural no son desarrolladas en este lenguaje.

UIMA incluye un conjunto de APIS y herramientas para crear componentes de análisis. Su principal componente son los *Analysis Engine*, más comúnmente llamados por sus iniciales AE, estos son organizados en cierto orden con el fin de analizar documentos. Uno o varios de estos AE son los que van a cumplir el rol de módulo dentro del sistema. Dentro de los AE es donde se encuentran los algoritmos que realizan el análisis del documento y registran los datos obtenidos (por ejemplo, detectar nombres de personas). Estos algoritmos no están dentro del AE propiamente dicho sino que quien

los contiene es el llamado *Annotator*. Existen dos tipos de AE: simples o agregados. Los simples incluyen un único *Annotator*. Los AE agregados son la composición de uno o más AE simples o agregados. El diagrama de la figura 4.1 ilustra lo explicado.

Los *Annotators* representan y comparten sus resultados mediante la utilización de *Common Analysis Structures* (CAS). Un CAS es un contenedor encargado de almacenar y manipular objetos fuertemente tipados mediante la utilización de propiedades, estos objetos suelen llamarse anotaciones. Cada *Annotator* procesa y modifica el CAS, generando o modificando anotaciones realizadas por otros *Annotators*. El CAS ingresa al sistema conteniendo únicamente el texto a ser analizado, en el caso de WebQA, la pregunta. Luego a medida que este es pasado por cada uno de los AE que componen el flujo del sistema, se le agregan anotaciones como por ejemplo, el tipo de la pregunta, los tokens que posee la pregunta, etc. Una vez que se completa el flujo el CAS resultante del mismo contiene la respuesta a la pregunta en una o varias anotaciones, más todas las anotaciones que se fueron realizando a lo largo del flujo.

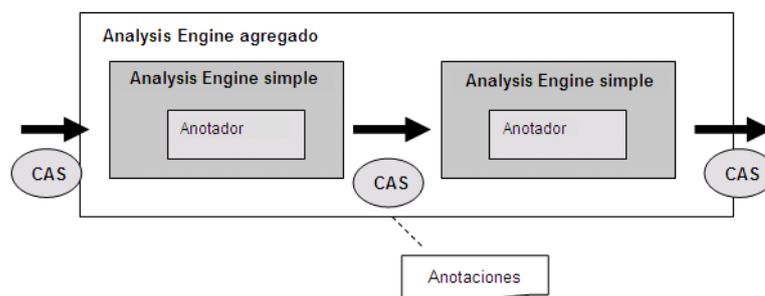


Figura 4.1: Actores principales de la arquitectura UIMA.

4.1. Componentes de WebQA

En WebQA se crearon distintos AE para la implementación de las etapas mencionadas en el capítulo 3. En la figura 4.2 se puede observar un diagrama general de componentes del sistema WebQA. En éste, se muestran los AE con las respectivas anotaciones que generan en el CAS. Estos AE son agregados y se componen de los AE simples que implementan los respectivos algoritmos. En la figura 4.3 se puede ver la composición de los AE agregados.

El análisis de la pregunta genera la anotación Pregunta que es la primera del CAS, la cual está constituida por distintos tipos de datos que se corresponden con la información obtenida a partir de este análisis. El análisis, en

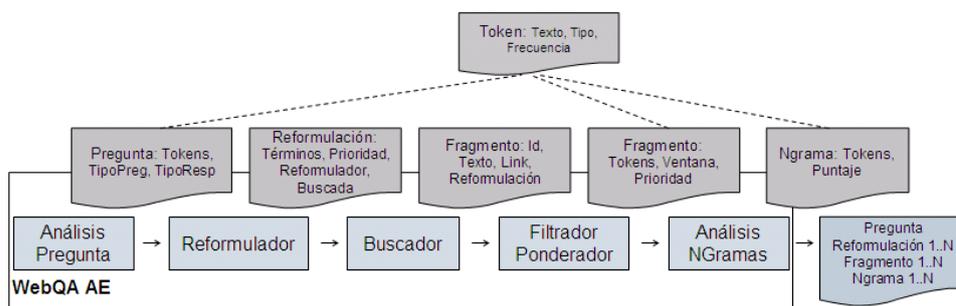


Figura 4.2: Diagrama general de componentes de WebQA, con sus respectivas anotaciones.

el cual se etiqueta cada palabra de la pregunta, se guarda en el tipo de datos Token. Un token está compuesto por un texto que es la palabra analizada, un tipo que es la etiqueta asignada por el tagger, y la frecuencia que se utiliza para determinar las palabras poco frecuentes (*stop words*). También se anota el tipo de la pregunta y el tipo de la respuesta esperada.

Por cada reformulación que retorne el reformulador, se genera una anotación Reformulación. En ésta se guarda la consulta a analizar. También se guarda el nombre del reformulador que la generó, una prioridad que se utilizará más adelante para la ponderación de los fragmentos obtenidos por la consulta, y finalmente una bandera que indica si la consulta fue o no enviada al buscador. La razón de esta bandera es que el flujo es condicional por lo que al momento de enviar las consultas al buscador, deben estar identificadas las que ya fueron realizadas. Los detalles del flujo condicional se explican en el punto 4.1 de este capítulo.

Luego se pasa el CAS al componente “Buscador”, el cual se encarga de leer las reformulaciones y enviar las consultas al motor de búsqueda. El análisis engine “Buscador” está compuesto por los AE correspondientes a los buscadores que se utilizarán (ver figura 4.3). Si se utiliza Google, el AE “Buscador” contendrá únicamente al “BuscadorGoogle”. También existe la posibilidad de utilizar más de un buscador, en ese caso se harán las consultas en todos los buscadores seleccionados. Con los resultados obtenidos se generarán nuevas anotaciones en el CAS. Por cada uno de los fragmentos obtenidos, se crea una anotación Fragmento, que representa el resumen del documento retornado por el buscador.

El Filtrador y Ponderador se encarga de eliminar y modificar, respectivamente, anotaciones de los fragmentos obtenidos por el buscador. Se realiza un análisis morfológico del fragmento, y al igual que para la pregunta se generan anotaciones con sus respectivos tokens etiquetados. Luego, el filtrador elimina del CAS los fragmentos que no tienen ningún token del tipo de

respuesta esperado. Para los fragmentos restantes, el ponderador les asigna un puntaje.

Finalmente el componente “Análisis Ngramas” genera las anotaciones de los ngramas obtenidos, guardando para cada uno la lista de tokens con que se componen, y también sus respectivos puntajes.

El CAS final tendrá toda la información relacionada a la pregunta, reformulaciones, fragmentos y ngramas, obtenida a lo largo de todo este proceso de análisis. Está listo para ser consumido de la manera que se desee. En nuestro caso se presentan al usuario los fragmentos ordenados y con la respuesta marcada según el ngrama de mayor puntaje.

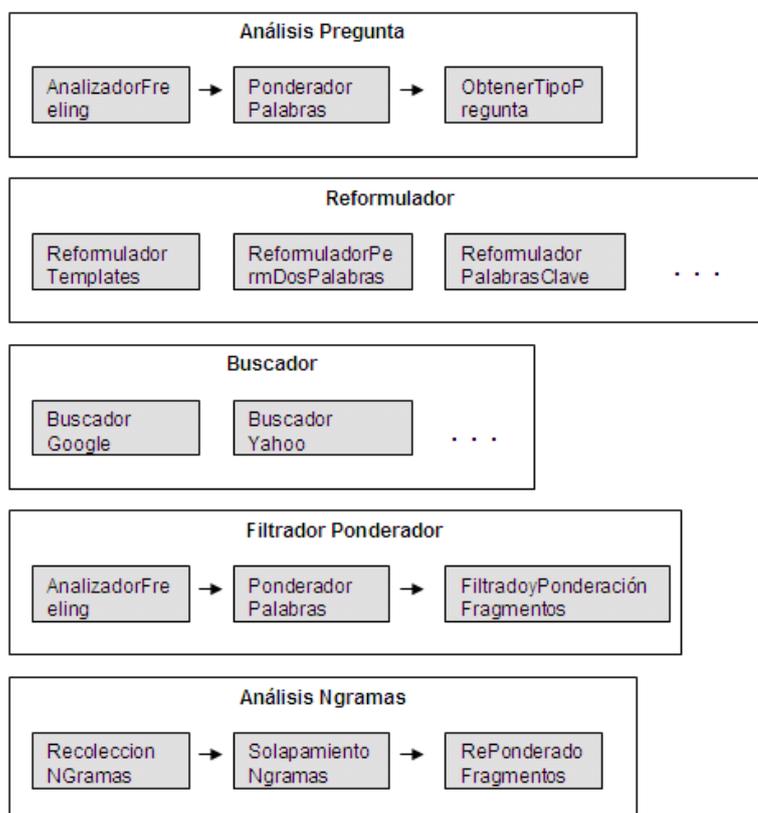


Figura 4.3: Composición de los AE agregados.

Flujo condicional de componentes

Como se mencionó en el capítulo 2, se utilizaron tres técnicas para formular las consultas. Cada una de estas técnicas fueron implementadas en

distintos reformuladores: “ReformuladorPatrones”, “ReformuladorDosPalabras” y “ReformuladorPalabrasClave” respectivamente. Sin embargo, dada una pregunta, podría ocurrir que con el primer reformulador ya se obtengan una cantidad de resultados suficientes, es decir, en muchos casos no sería necesario que se ejecuten los tres, debido a que producirían una sobrecarga innecesaria. Por esta razón se decidió que el flujo sea condicional. Esto es, se ejecuta siempre el primero, se envían las consultas al buscador y se pasa a las etapas siguientes de filtrado y ponderado de los fragmentos obtenidos. Finalmente antes de iniciar la etapa de análisis de ngramas para extraer la respuesta, se chequea que la cantidad de fragmentos a analizar no sea muy pequeña (por ejemplo: mínimo diez fragmentos). Si no es suficiente, se vuelve a la etapa de reformulación de la pregunta, y se aplica el siguiente reformulador. Así se repite el proceso hasta que haya una cantidad de fragmentos aceptable para analizar en la etapa de ngramas, o hasta que se utilice el último reformulador.

El criterio para determinar el orden de ejecución de los reformuladores, se establece en base a la especificidad de las reformulaciones que generan. Se desea realizar primero las consultas más específicas y luego las más generales, por lo que el orden de ejecución es el siguiente: primero el ReformuladorPatrones, luego el ReformuladorDosPalabras y por último el ReformuladorPalabrasClave.

“ReformuladorPatrones” y “ReformuladorDosPalabras”, ambos generan frases basándose en la forma declarativa de la pregunta, por lo que es de esperar que se obtengan pocos resultados pero muy buenos. El “ReformuladorPalabrasClave” asegura encontrar resultados, pero es muy probable que devuelva ruido. Es por esta razón que es el último en ejecutar en el flujo condicional.

Agregar un nuevo reformulador que utilice otras técnicas, quitar un reformulador, o modificar el orden de ejecución es sumamente sencillo debido a la modularidad de la arquitectura UIMA [15]. En la figura 4.4 se puede ver el diagrama del flujo de componentes

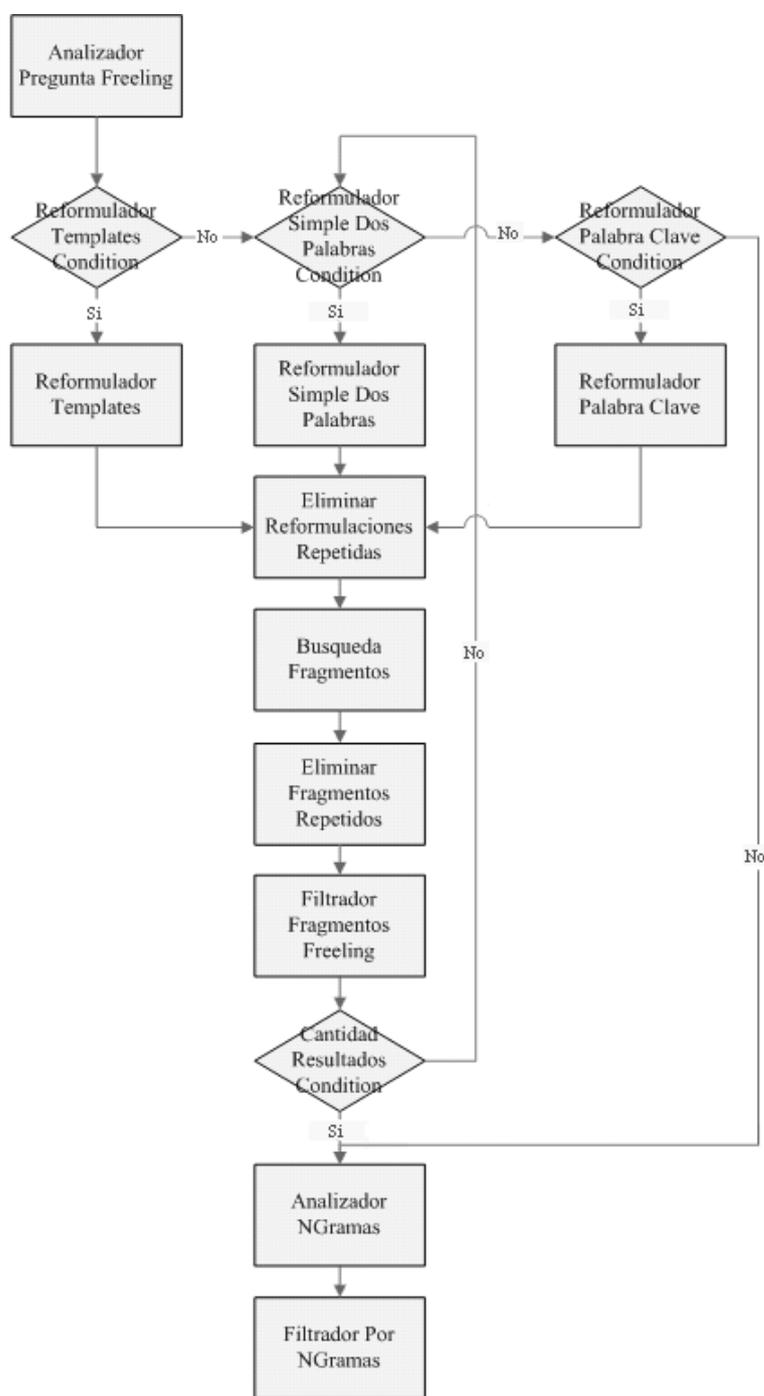


Figura 4.4: Flujo condicional de componentes.

Es importante aclarar que debido a que se realizan varias consultas y que se podría utilizar más de un motor de búsqueda, es muy probable que haya documentos repetidos. Por esta razón, luego de la búsqueda se realiza un filtrado de documentos repetidos por parte del AE “EliminarFragmentosRepetidos”.

La condición previa al “Analizado Ngramas” determina si la cantidad de resultados obtenidos es suficiente. Si falla se vuelve a la etapa de reformulación de la pregunta, para ejecutar el siguiente reformulador.

Este flujo podría tener variantes, por ejemplo en el caso de querer correr dos reformuladores en una misma iteración, o de volver a iterar sobre todo el sistema pero cambiando el buscador, etc.

4.2. Implementación del flujo de componentes

La versión 2.0 de UIMA ([15]) tiene la ventaja de permitir flujos condicionales. Esto es, de acuerdo a determinado estado del sistema (en términos de UIMA está dado por el documento a procesar), se determine el AE a ejecutar.

El framework sin embargo no ofrece ningún mecanismo para expresar el flujo, por lo que lo debe implementar el desarrollador.

Para que el flujo sea fácilmente modificable se decidió expresarlo en un archivo xml, donde se señalan nodos que representan los AE y condiciones que son clases que determinan el flujo a seguir. Estas clases se levantan por reflection¹, ya que de esa forma alcanza con modificar el archivo xml para agregar o quitar componentes, o para alterar el flujo de ejecución.

En el cuadro 4.1 se presenta uno de los xml para expresar el flujo de WebQA que se utilizó en la versión final, con Google como herramienta de búsqueda y Freeling como POS-tagger.

Como se puede apreciar, se tienen nodos “*Análisis Engine*” y nodos “*Condition*”. Cada nodo de tiene el nombre del AE a ejecutar o el nombre de la clase que implementa la condición respectivamente. Como atributos se tiene un índice, (el cual referencian otros nodos, actúa de identificador) y una *flag* ‘*final*’ que indica si en ese nodo termina el flujo.

Para los nodos *conditions* se tienen dos atributos: *success* y *fail*. Indican el índice del nodo con el cual se debe seguir (puede ser un AE u otra condición), dependiendo si la condición evalúa verdadero o a falso. Las *conditions*

¹Reflection: forma de crear e invocar clases de forma dinámica, es decir en tiempo de ejecución.

```

<?xml version="1.0"?>
<Flow>
  <AnalysisEngine index="1" final="false" next="2">
    AnalizadorPreguntaFreeling
  </AnalysisEngine>
  <Condition index="2" final="false" success="fail="4»3
    ReformuladorPatronsCondition
  </Condition>
  <AnalysisEngine index="3" final="false" next="8">
    ReformuladorPatrones
  </AnalysisEngine>
  <Condition index="4" final="false" success="5" fail="6">
    ReformuladorSimpleDosPalabrasCondition
  </Condition>
  <AnalysisEngine index="5" final="false" next="8">
    ReformuladorSimpleDosPalabras
  </AnalysisEngine>
  <Condition index="6" final="false" success ="7" fail="13">
    ReformuladorPalabraClaveCondition
  </Condition>
  <AnalysisEngine index="7" final="false" next="8">
    ReformuladorPalabraClave
  </AnalysisEngine>
  <AnalysisEngine index="8" final="false">
    EliminarReformulacionesRepetidas
  </AnalysisEngine>
  <AnalysisEngine index="9" final="false">
    BusquedaFragmentos
  </AnalysisEngine>
  <AnalysisEngine index="10" final="false">
    EliminarFragmentosRepetidos
  </AnalysisEngine>
  <AnalysisEngine index="11" final="false">
    FiltradorFragmentosFreeling
  </AnalysisEngine>
  <Condition index="12" final="false" success ="13" fail="2">
    CantidadResultadosCondition
  </Condition>
  <AnalysisEngine index="13" final="false">
    AnalizadorNGramas
  </AnalysisEngine>
  <AnalysisEngine index="14" final="true">
    FiltradorPorNGramas
  </AnalysisEngine>
</Flow>

```

Cuadro 4.1: Xml para expresar el flujo de WebQA que se utilizó en la versión final.

se utilizan para modelar la reformulación a demanda. Se tiene una condición antes del análisis de ngramas que evalúa si se hay una cantidad de resultados aceptable para analizar (parámetro del sistema). Si la cantidad no es suficiente se vuelve a reformular la pregunta con el siguiente reformulador. Por cada reformulador hay una condición que chequea si ya fue utilizado. Para los nodos AE está el atributo *next* que indica el índice del nodo seguir en el flujo (puede ser un nodo o una condition). Si no se indica este atributo *next* se asume que se ejecuta el siguiente.

En la figura 4.1 se puede ver el diagrama de flujo de componentes.

4.3. Implementación de patrones

4.3.1. Patrones para el análisis de la pregunta

En el prototipo inicial, este análisis se realizaba por medio de métodos de una clase que identificaban ciertos patrones de la formulación de la pregunta, determinando de esa manera el tipo de pregunta y el tipo de respuesta esperado. Por ejemplo si la pregunta comenzaba con la palabra “Cuándo”, inmediatamente se identificaba que el tipo de la pregunta es “CUANDO” y el tipo de la respuesta esperada es una fecha. El problema es que la variedad de maneras de formular una pregunta es muy grande y de esta forma era complejo y poco claro agregar nuevas formulaciones a ser reconocidas. Es por esto que finalmente se optó por implementar la solución de formulación de patrones, explicada en la sección 3.1.

Los patrones se leen de un archivo .xml y la estructura general de cada uno es la siguiente:

```
<formulacion forma='... '>
  <condiciones>
    <palabras> p1 p2 | p3 | ... </palabras>
    <tipotokens> t1 | t2 t3 | ...</tipotokens>
  </condiciones>
  <info>
    <tipoPregunta> ... </tipoPregunta>
    <tipoRespuesta> ... </tipoRespuesta>
    <cantidadTokensPregunta> ... </cantidadTokensPregunta>
  </info>
</formulacion>
```

Cada patrón se inicia con la etiqueta *formulacion*, el cual tiene un atributo “*forma*” (forma de la pregunta). Este atributo se utiliza para identificar

con que palabra/s comienza la pregunta. Por ejemplo si en determinado patrón *forma* = “CUANDO”, entonces las preguntas que empiecen con “Cuándo” le corresponden. Algunos de los valores posibles del atributo *forma* son los siguientes:

CUANDO | DONDE | QUIEN | EN QUE | A QUE | CUANTOS | CUANTAS | CUAL | QUE | COMO | POR QUE.

El atributo *forma* no determina por sí solo que la pregunta se corresponda al patrón. Cada formulación tiene dos subetiquetas. Una es *condiciones* y la otra es *info*.

La etiqueta *condiciones* determina requisitos adicionales que debe cumplir la formulación para que efectivamente se corresponda con el patrón. Esta no es obligatoria, por lo tanto si no se incluye, se chequeará únicamente que en la formulación coincida el atributo *forma*. Las condiciones se aplican sobre los tokens de la pregunta siguientes a los que se corresponden con el atributo *forma*. Se puede establecer que coincidan palabras específicas, o palabras que cumplan cierta etiqueta gramatical. También se pueden especificar conjunciones o disyunciones.

Para la etiqueta palabras, por ejemplo, el valor “p1 p2 | p3” indicaría que la forma de la pregunta debería estar seguida de la palabras p1 y p2, o solamente de la palabra p3. Se trabaja de la misma manera para la etiqueta *tipotokens*. Si el valor fuera “NP | NC NP”, indicaría que la forma de la pregunta debe estar seguida de un nombre propio, o de un nombre común y un nombre propio. Adicionalmente, para ampliar la gama de posibilidades, se agregó el símbolo “%%” para indicar que un token podría ser cualquiera. Entonces, por ejemplo, el valor “%% NP” indicaría que se espera cualquier token seguido de un nombre propio. Un caso de aplicación es el siguiente: dadas las preguntas “¿Cuál es el año en que se descubrió América?” y “¿Cuál es la fecha de nacimiento de Carlos Gardel?”, ambas son del tipo “CUANDO”, esperan una fecha como respuesta y podrían ser reconocidas con el siguiente patrón: <“CUAL ES” %% (FECHA | AÑO | MES ...)>, donde “%%” coincidiría con “el” y “la” respectivamente.

La etiqueta *info* contiene la información de la pregunta que se guardará como resultado final del análisis. Los datos que contiene esta etiqueta son el tipo de la pregunta y el tipo de respuesta esperado.

Los tipos de pregunta posibles son:

CUANDO | DONDE | QUIEN | CUANTOS | COMO | POR QUE

Preguntas del tipo “DONDE” podrían ser: “Dónde ...?”, “En qué país ...?”, “En qué lugar ...?”, etc. Para las formulaciones de preguntas que no coinciden con ningún patrón existe el tipo “INDEF”.

Los posibles tipos de respuesta esperada son:
FECHA | LUGAR | PERSONA | NOMBRE | DESCRIPCION | NUMERO

El tipo de respuesta esperado es muy importante, dado que cumple un papel fundamental al momento de determinar la respuesta. Para la determinación de la respuesta, en etapas posteriores, se compara la etiqueta del token a analizar y el tipo de respuesta esperada., y se verifica que correspondan. El cuadro de correspondencias se puede ver en el cuadro 4.2.

Tipo de respuesta esperada	Etiqueta	Descripción
FECHA	W	Fecha
LUGAR	NP	Nombre propio
PERSONA	NP	Nombre propio
NOMBRE	NC	Nombre común
NUMERO	Z	Cifras
DESCRIPCION	-	no se espera como respuesta ningún tipo de token en particular

Cuadro 4.2: Correspondencias entre tipos de respuesta esperada obtenidos de los patrones, con las etiquetas gramaticales de los tokens.

La última sub etiqueta de *info* es *cantidadTokensPregunta*. Esta determina cuantos tokens se deben quitar al principio de la pregunta para pasar la oración de su forma interrogativa a su forma declarativa. Tener esta información es de mucha utilidad para la etapa siguiente de reformulación de la pregunta. En el cuadro 4.3, se pueden ver ejemplos.

Forma interrogativa	Forma declarativa	cantidadTokensPregunta
¿Dónde se jugó el primer mundial de fútbol?	se jugó el primer mundial de fútbol	1
¿En que año se fundó la ciudad de Montevideo?	se fundó la ciudad de Montevideo	3
¿Con quién se casó John Lennon?	se casó John Lennon	2

Cuadro 4.3: Ejemplos de asignación de valores para el atributo cantidadTokensPregunta.

A continuación se presentará una pregunta de ejemplo y se mostrará el proceso completo de análisis de la pregunta.

Pregunta a analizar: ¿Quién fue José Gervasio Artigas?

La primera etapa es la del etiquetado de la pregunta y determinación de la frecuencia de cada token. El etiquetado es realizado por FreeLing. Según

este resultado, luego se le asigna un determinado peso a cada palabra, esta asignación se puede observar en el cuadro 4.4.

Token	¿	Quién	fue	José_Gervasio _Artigas	?
Etiqueta	Fia	PT0CS000	VMIS3S0	NP00000	Fit
Peso	0	1	2	2	0

Cuadro 4.4: Análisis morfológico de la pregunta “Quién fue José Gervasio Artigas?”

La etiqueta de todos los signos de puntuación es la “F”, diferenciando ‘¿’ y ‘?’ con “Fia” y “Fit” respectivamente. La palabra “quién” es identificada como pronombre interrogativo (“PT”), “fue” un verbo (“V”), y “José_Gervasio_Artigas” un nombre propio “NP”. El peso valores entre 0 y 2, según la correspondencia establecida entre etiqueta-frecuencia.

El siguiente paso es determinar con qué patrón se corresponde la pregunta, para obtener el tipo de la pregunta y el tipo de respuesta esperado. El respectivo patrón es el siguiente:

```
<formulacion forma='QUIEN'>
  <condiciones>
    <palabras>ES|FUE|ERA</palabras>
    <tipotokens>%% NP</tipotokens>
  </condiciones>
  <info>
    <tipoPregunta>QUIEN</tipoPregunta>
    <tipoRespuesta>DESCRIPCION</tipoRespuesta>
    <cantidadTokensPregunta>1</cantidadTokensPregunta>
  </info>
</formulacion>
```

Este patrón determina formulaciones del tipo “Quién <es | fue> <nombre propio>”, en el cual se espera una descripción como respuesta. El atributo *forma* es QUIEN, y la pregunta empieza con la palabra “Quién”, por lo tanto coincide. Las *condiciones* de este patrón se componen de las dos posibles: <palabras> y <tipotokens>. La condición <palabras> indica que después de la palabra “quién” se espera “es” o “fue”. En el ejemplo coincide “fue”. Luego la condición *tipotokens* indica que después de la palabra “quién” se espera cualquier token (“%%”) seguido de un nombre propio (“NP”). En la pregunta del ejemplo el token siguiente a “quién”, es “fue”, seguido de “José_Gervasio_Artigas” que es un nombre propio, por lo tanto cumple con las condiciones y a la pregunta le corresponde este patrón. La información que se obtiene es que el tipo de la pregunta es “QUIEN” y el tipo

de respuesta esperado es “DESCRIPCION”. Como información adicional se guarda el atributo *cantidadTokensPregunta*. Dado que la forma declarativa de la pregunta es “fue José Gervasio Artigas”, su valor es igual a 1.

4.3.2. Reformulador basado en patrones

Se definieron una serie de patrones, con formato similar a los utilizados en el análisis de la pregunta, que se basan en identificar las palabras con que empieza la pregunta y luego definir una serie de reformulaciones.

Los patrones se leen de un archivo .xml y la estructura general de cada uno es la siguiente:

```
<formulación forma='...'>
  <reformulación> ... </reformulación>
  <reformulación> ... </reformulación>
  ...
</formulación>
...
```

Al igual que los patrones correspondientes al análisis de la pregunta, cada uno está dentro de la etiqueta *formulación*, la cual también tiene el atributo “forma”, que determina con qué palabras empieza la pregunta (por ej. “Cuándo”, “En qué”, etc.). Si el comienzo de la pregunta coincide con la forma, entonces la pregunta se corresponde al patrón y se toman las reformulaciones correspondientes.

La etiqueta *formulación*, se compone de varias sub etiquetas, *reformulación*, las cuales tienen las reformulaciones de la pregunta correspondientes. Las reformulaciones están basadas en la siguiente expresión regular:

$$(X \mid V \mid Y) (X \mid V \mid Y)^* (+ (X \mid V \mid Y) (X \mid V \mid Y)^*)^*$$

donde:

- X = frase anterior al verbo de la pregunta
- V = verbo de la pregunta
- Y = frase posterior al verbo de la pregunta

En el cuadro 3.2 se pueden ver distintos ejemplos

4.4. Inclusión de herramientas para análisis

Para el análisis tanto de la pregunta, como de los fragmentos, fue necesario la inclusión de un Part-of-Speech Tagger. El OpenNLP Tagger [19], y Freeling 1.4 [11]. fueron las herramientas que se manejaron para este proyecto.

Para el prototipo inicial se utilizó el OpenNLP Tagger, elaborado por la organización OpenNLP [19], dedicada a implementar, fomentar y colaborar en proyectos open source relacionados con el procesamiento de lenguaje natural. Debido a que estaba desarrollado en java y utilizaba la arquitectura UIMA, se incluyó rápidamente en el sistema.

Luego se decidió cambiar por otro POS-tagger, siempre open source, que fuera más preciso en el etiquetado. El elegido finalmente fue FreeLing 1.4. Ambos analizadores utilizan el conjunto de etiquetas propuestas por el grupo Eagles [8] para la representación de la información morfológica de las palabras. Tanto OpenNLP como Freeling tienen su componente (AE) UIMA respectivo. La particularidad es que se implementó de forma que con los dos se generen exactamente las mismas anotaciones (anotación *Token*, ver figura 4.2). Esta fue una decisión básica de diseño, para que la sustitución de un analizador por otro, sea totalmente transparente. Para el caso de OpenNLP, que ya es un componente de UIMA, las anotaciones que genera son convertidas al formato definido para WebQA. De esta forma en el resto de las etapas siempre se trabaja con un mismo tipo de anotación definido e independiente del tagger utilizado.

Sin embargo la sustitución de OpenNLP por Freeling no fue trivial y surgieron algunos problemas. En primer lugar, Freeling tiene la particularidad de que un token puede estar formado por varias palabras si se refieren al mismo concepto. Por ejemplo, si en el texto está incluido el nombre “Juan Pérez”, el OpenNLP Tagger toma al token “Juan” como un nombre propio y al token “Pérez” como otro nombre propio. En cambio Freeling infiere de que se está refiriendo al mismo concepto y retorna un solo token “Juan_Perez” como un nombre propio. En el caso de análisis de ngramas se utiliza la repetición de las palabras en los fragmentos. Como con Freeling un token puede estar formado de varias palabras hubo que realizar cambios en la programación. El cambio consistió en realizar la desagregación de cada token para obtener las palabras que lo componen. Esta lógica permite trabajar tanto con Freeling como con OpenNLP, ya que para este último al realizar la separación del token se obtiene el mismo texto del token.

Un segundo problema que surgió en la inclusión de Freeling fue que las API están en C++. De todas maneras ofrece una capa con un conjunto de

clases de java que integran C++, con JNI. Se intentó compilar estas clases pero no se pudo lograr en un período razonable de tiempo. Por esta razón, la alternativa que quedó fue utilizar el ejecutable. Como consecuencia de esto surgieron problemas de performance en el análisis de los fragmentos retornados por el buscador, que se debieron resolver. En cada iteración con cada reformulador se tienen para analizar una cantidad de fragmentos del orden de:

*cantidad consultas * cantidad fragmentos obtenidos por la consulta*

La cantidad de fragmentos obtenidos por la consulta está acotada a un número máximo configurable (actualmente 10 resultados por consulta). Supongamos que por reformulador se tiene en promedio de 4 reformulaciones, y por lo tanto cuatro consultas al buscador. Entonces dada una pregunta, como máximo en cada iteración se tienen que analizar alrededor de 40 fragmentos. Con OpenNLP no hay mayor problema porque el análisis se hace en memoria, por lo que no hay diferencia en tiempos entre analizar un texto grande, o muchos textos pequeños. Como contrapartida, en el caso de Freeling se requiere el llamado a un ejecutable, grabar un archivo en disco con el texto a analizar y leer un archivo con el texto procesado. Realizar este procedimiento muchas veces se transforma en un problema de performance. Para solucionar este inconveniente se tuvo que hacer un cambio en la programación, generando un solo archivo de entrada a Freeling con todos los fragmentos recuperados por el motor de búsqueda, de modo que el análisis se haga en una única llamada. Todo esta lógica esta encapsulada en el annotator del AE creado para Freeling.

También vale la pena destacar que al sustituir el tagger de OpenNLP por Freeling, se mejoró en el marcado final de la respuesta. Para presentar la respuesta, cada fragmento se reconstruye como texto html a partir sus tokens, obtenidos de las anotaciones realizadas en etapas anteriores. En el caso de los tokens que deban ser marcados en negrita, se agrega entre ellos las etiquetas html `y `. Como se mencionó anteriormente Freeling, a diferencia del OpenNLP tagger, si más de un token se refieren al mismo concepto los agrupa y etiqueta juntos, esto permite, por ejemplo, que si se tiene como respuesta el unigrama “Artigas”, luego en el fragmento se marcará en negrita la frase completa “José Gervasio Artigas”.

4.5. Motor de búsqueda

Para la recuperación de documentos de la Web se utilizó el motor de búsqueda de Google [13]. La decisión es de no atar el sistema a un solo motor de búsqueda, es por eso que se implementó de modo de que de manera sencilla se pueda quitar uno, incluir otro, o dejar más de uno a la vez. Los

annotators de los AE correspondientes a los buscadores, utilizan las API del motor de búsqueda correspondiente para realizar las consultas generadas en la etapa de reformulación de la pregunta.

Las consultas son filtradas por idioma, solamente se recuperan documentos en español. También para acotar la cantidad de fragmentos a analizar en etapas posteriores, y mejorar la performance general del sistema, se estableció un tope de resultados por consulta. Este tope es un parámetro de ajuste que actualmente es de diez fragmentos.

Aparte del motor de Google se utilizó el de Yahoo [26]. Se realizaron pruebas con cada uno por separado y con los dos a la vez. En el capítulo 5 de evaluación del sistema se detallan los resultados obtenidos de estas pruebas.

Capítulo 5

Evaluación

En este capítulo se presentan los resultados obtenidos en la evaluación de WebQA, así como una comparación con otros sistemas de Búsqueda de Respuestas. La evaluación se realiza en base a un conjunto de preguntas elegido arbitrariamente. Se estudia el rendimiento del sistema con las distintas combinaciones de etiquetadores y buscadores y para la mejor combinación se evalúa el rendimiento en cada tipo de pregunta. Las medidas de evaluación utilizadas son indicadores de la precisión de la respuesta extraída (marcada en los fragmentos) y del ranking de fragmentos presentados como respuesta. Se tienen dos formas de evaluar porque para algunas preguntas no se realiza la etapa de extracción de la respuesta (porque esperan una descripción o son de tipo indefinido) y además porque el usuario puede recibir una respuesta marcada en los fragmentos equivocada, pero tener en el ranking otros fragmentos (o hasta el mismo que tiene la respuesta marcada) que sí contienen la respuesta correcta.

Para comparar WebQA con otros sistemas se presenta un análisis de las alternativas posibles y los resultados de la comparación con los sistemas participantes en las TREC 2005 [24], en las CLEF 2003 [17] y con el sistema de Alejandro del Castillo [9], desarrollado en el año 2005 y para el idioma español, que utiliza la Web como repositorio de información. El catalogo de preguntas utilizado en todas las comparaciones es el de las CLEF del año 2003.

5.1. Forma de Evaluación

Se realizaron dos tipos de evaluaciones:

1. En base a la respuesta extraída, utilizando para ello el cálculo de la Precisión.
2. En base a los fragmentos presentados con dos medidas de desempeño: MRR(Mean Reciprocal Rank) y Precisión.

Ambas evaluaciones se hicieron analizando de forma manual las respuestas obtenidas de WebQA.

5.1.1. En base a la respuesta extraída

En esta evaluación se mide la precisión de la respuesta extraída por la técnica de Ngramas, respuesta que se resalta en los fragmentos presentados al usuario. No se toman en cuenta los tipos de pregunta en las que se espera una descripción como respuesta ni las preguntas con tipo Indefinido, ya que en esos casos no se realiza la etapa de extracción de respuesta.

Precisión = cantidad respuestas correctas/cantidad preguntas realizadas.

5.1.2. En base a los fragmentos presentados

En esta evaluación se evalúa el ranking de fragmentos presentados, analizando los primeros n fragmentos. Se calculan dos medidas de performance:

- Precisión.
- MRR (Mean Reciprocal Rank).

La precisión se define igual que para la evaluación de la respuesta extraída, la diferencia es que se analiza todo el conjunto de preguntas, la evaluación anterior solo toma un subconjunto para el cual se realiza la etapa de extracción. Además el criterio con el cual una respuesta es correcta cambia, pues no se analiza la respuesta extraída sino los fragmentos. Se define que el sistema brinda una respuesta correcta si se encuentra en los primeros n fragmentos presentados.

El MRR [9] se define de la siguiente forma:

$$MMR = \frac{\sum_{i=1}^n \frac{1}{pos(i)}}{n}$$

Donde n corresponde al número de preguntas de prueba y $pos(i)$ indica la posición de la primera respuesta correcta para la pregunta i . El valor de

$\frac{1}{pos(i)}$ será cero si no se ha encontrado la respuesta. Cada pregunta se califica asignándole el inverso de la posición del primer fragmento retornado por el sistema, que contiene la respuesta esperada. El MRR es la media del recíproco del ranking de los n primeros fragmentos presentados. La importancia del MRR es que considera la posición en el ranking de la respuesta correcta a diferencia de la Precisión. O sea, no solo es importante que el sistema devuelva una respuesta correcta en los primeros fragmentos, también es importante en la posición que se da de la respuesta.

La evaluación del ranking de los fragmentos presentados con sus dos medidas (Precisión y MRR) no es independiente de la evaluación de la respuesta extraída, pues si ésta es correcta va a estar en el primer fragmento presentado debido al filtrado por n-gramas (3.4.3).

5.2. Conjunto de preguntas utilizado

Para la evaluación de WebQA se utiliza un conjunto de 150 preguntas (Apéndice A.3). Este conjunto fue construido de forma manual, buscando que sea representativo en los tipos de pregunta y en las reformulaciones que cada uno puede tener. La mayor parte de las preguntas pertenecen a la categoría de fácticas (esperan una sola respuesta correcta y concreta) y se clasifican en las siguientes categorías: “Dónde”, “Cuándo”, “Cuánto”, “Quién”, “Cuál”.

Recordemos que WebQA siempre busca dar respuesta aunque la pregunta no esté dentro de los tipos de preguntas explícitamente soportadas, por lo tanto se agregaron adicionalmente preguntas con categoría “indefinida” con el objetivo de analizar el comportamiento en casos límite. Algunos ejemplos de preguntas con categoría indefinida son “¿En qué consiste el teorema de Pitágoras?”, “¿Qué ingredientes tiene el Cuba Libre?” y “¿Cuáles son los planetas del sistema solar?”.

Dentro de cada categoría hay varias formas en las que se formulan las preguntas, algunas de estas se presentan en el cuadro 5.1, junto a la cantidad de preguntas en cada categoría.

5.3. Resultado de la Evaluación

En esta sección se presentan los resultados de la evaluación, tanto de la respuesta extraída, como del ranking de fragmentos presentados. Además de

Categoría	Cantidad	Ejemplos de formulaciones
Dónde	30	Dónde... , En qué lugar... , En qué país...
Cuándo	41	Cuándo... , En qué fecha... , En qué año...
Quién	39	Quién...
Cuántos	20	A qué temperatura... , Cuántas...
Cuál	13	Cuál...
Indefinidas	7	...

Cuadro 5.1: Cantidad y ejemplos de reformulaciones por categoría de preguntas.

presentar el rendimiento general del sistema se analiza el rendimiento utilizando distintos buscadores (Google, Yahoo) y distintos etiquetadores de texto (OpenNLP, Freeling). Para la mejor combinación de buscador-etiquetador se analiza el rendimiento del sistema en cada tipo de pregunta.

5.3.1. En base a la respuesta extraída

En esta sección se presenta el resultado de la evaluación de la respuesta extraída, en base lo expuesto en la sección (5.1.1). El mejor rendimiento se logra con la combinación de Google y Freeling, donde se responde correctamente un 59 % de las preguntas realizadas. Mientras que con las restantes tres combinaciones se logra un rendimiento que varía entre un 51 % a 55 % (ver 5.2).

Combinación	Precisión respuesta extraída
Freeling-Google	0,5906
Freeling-Yahoo	0,5484
OpenNLP-Yahoo	0,5391
OpenNLP-Google	0,5137

Cuadro 5.2: Precisión de los resultados de ngramas, según la combinación etiquetador-buscador.

Analizando los buscadores se observa que con Google se responde bien en promedio el 56 % de las preguntas, mientras que con Yahoo se responde bien un promedio de 54,5 %. Las razones por la cual se puede tener mejor resultado con un buscador que con otro pueden ser la relevancia de los primeros n fragmentos (los tomados en cuenta por WebQA) y la calidad de los resúmenes (*snippet*). Para el caso de la extracción de respuesta las

diferencias de rendimiento con Google o Yahoo son pequeñas y no podemos asegurar que la utilización de un buscador sea mejor que otro.

Analizando los etiquetador se tiene que con Freeling se responde un promedio de 57% de preguntas correctas, mientras que con OpenNLP se responde bien un promedio de 52,5%. La diferencia de rendimiento es más significativa con el cambio de los etiquetadores que con los buscadores. Freeling es un etiquetador de textos más preciso que OpenNLP, por lo cual era de esperar este resultado. El impacto del etiquetador en la etapa de extracción de la respuesta con ngramas es esencial, pues se analiza la repetición de tokens que sean del tipo de respuesta esperado y para esto se utiliza el resultado del etiquetador.

5.3.2. En base a los fragmentos presentados

En esta sección se presenta el resultado de la evaluación de los fragmentos presentados, en base lo expuesto en la sección (5.1.2). Para el cálculo de la precisión se consideró que una respuesta es correcta si se encuentra en los primeros 3 fragmentos presentados.

Combinación	MRR	Precisión
Freeling-Google	0,5776	0,6929
Freeling-Yahoo	0,5793	0,6857
OpenNLP-Yahoo	0,4933	0,6204
OpenNLP-Google	0,4353	0,5631

Cuadro 5.3: MRR y Precisión del ranking de fragmentos según combinación etiquetador de texto - buscador.

Al evaluar los fragmentos presentados, las combinaciones de Freeling-Google y Freeling-Yahoo tienen un rendimiento similar, mientras que el rendimiento en la etapa de extracción de respuesta es superior para Freeling-Google. Al igual que en la evaluación de la respuesta extraída, las combinaciones que tienen a Freeling son mejores a las que tienen OpenNLP (el promedio de precisión es de 0.6893% para Freeling y de 0,5917% para OpenNLP). El rendimiento de los buscadores es levemente superior para Yahoo. Al igual que en la evaluación de la extracción de la respuesta nos se puede asegurar que el rendimiento del sistema sea significativamente mayor con un buscador que con otro.

Es lógico que la precisión de los fragmentos sea mayor a la precisión de la extracción de la respuesta 5.3.2, ya que como se mencionó, si el ngrama es correcto, éste se encontrará en el primer fragmento retornado, pero si el

ngrama es incorrecto perfectamente se podría dar el caso que la respuesta correcta estuviera en cualquiera de los tres primeros fragmentos.

Combinación	Precisión Fragmento	Precisión Ext. Respuesta
Freeling-Google	0,6929	0,5906
Freeling-Yahoo	0,6857	0,5484
OpenNLP-Google	0,5631	0,5317
OpenNLP-Yahoo	0,6204	0,5391

Cuadro 5.4: Comparación de precisión en los fragmentos y ngramas, discriminado por etiquetador de texto -buscador.

5.3.3. Análisis según tipo de pregunta

Para el análisis de la evaluación del sistema por tipo de pregunta se fijó la mejor combinación de etiquetador de texto-buscador con mejor resultado: Freeling-Google. Se observa en el cuadro 5.5 que se tiene un rendimiento superior en las preguntas de tipo “Cuándo” y “Quién” y “Dónde”.

Tipo Pregunta	MRR	Precisión Fragmento	Precisión Ext. Respuesta
Cuál	0,4487	0,6923	0,4615
Cuándo	0,5833	0,7333	0,6896
Cuántos	0,4902	0,5294	0,2220
Donde	0,6595	0,7429	0,6563
Indefinido	0,3125	0,3750	0,1666
Quién	0,6383	0,7568	0,6857
Total	0,5776	0,6929	0,5906

Cuadro 5.5: Rendimiento del sistema por tipo de pregunta.

Las preguntas de tipo “Cuánto” son las que tienen un rendimiento mas bajo, independientemente de las herramientas utilizadas. Esto se debe a dos grandes motivos: los números pueden venir expresados de distintas formas, y en muchos casos no hay medidas exactas. Por ejemplo, ante la pregunta “¿Qué profundidad tiene el Océano Pacífico?”, se recuperaron los siguientes resultados:

- *El Pacífico es el océano más profundo, con una profundidad media de 12.900 pies.*
- *Para tener una idea tomemos la profundidad habitual del Océano Pacífico, que es de 4.000 m.*

5.4. Comparación de WebQA con otros sistemas

Para comparar WebQA con otros sistemas las principales dificultades son que la mayoría están contruidos para el idioma inglés y que un grupo reducido utiliza la Web como repositorio de información. Cualquier comparación con un sistema que no tenga estas características tendrá un factor de incertidumbre importante, ya sea traduciendo la pregunta de un idioma a otro o ignorando el hecho de que el repositorio de información sea distinto y con características particulares. Más allá de estas consideraciones se tienen las siguientes alternativas:

1. Comparar el sistema con los participantes en las TREC en la tarea Q&A.
2. Comparar el sistema con los participantes en las CLEF en la tarea “Múltiple Language Question Answering”.
3. Comparar el sistema con los disponibles online.
4. Comparar el sistema con otros para los cual se tenga información de su desempeño.

En la comparación es indispensable utilizar las mismas medidas de performance y los mismos criterios de evaluación entre los sistemas. Por ejemplo, en las últimas ediciones de las TREC, como de las CLEF, para las preguntas fácticas se considera una única respuesta y no se analiza un ranking de respuestas posibles. Los sistemas de las TREC, como los de las CLEF, buscan en un corpus de documentos, no en la Web, lo que hace que algunas preguntas tengan respuestas en un universo y no en otro.

El sistema para el idioma español del INAOE de Méjico, desarrollado por Alejandro del Castillo [9] utiliza la Web como repositorio de información y presenta una cantidad importante de datos sobre su evaluación, en vista de esto se realizará una comparación de WebQA con este sistema.

5.4.1. Conjunto de preguntas de evaluación

La comparación óptima exige utilizar el mismo conjunto de preguntas de evaluación. Para esto es necesario tener el catálogo de preguntas que se utilizaron tanto en las TREC como en las CLEF. Para los sistemas disponibles online se tiene la posibilidad de evaluarlos con el conjunto de preguntas con el que se evaluó WebQA o tener información sobre evaluaciones realizadas.

Para comparar los sistemas participantes en las TREC surge un inconveniente: desde las TREC 2004 en la categoría Q&A se cambió la forma de las preguntas, realizándolas en base a un contexto [23]. El catalogo de preguntas se compone en series basadas en un objetivo común, con la siguiente forma:

Objetivo: Segunda Guerra Mundial

Pregunta: ¿En qué año terminó?

Pregunta: ¿En qué año se incorporó EEUU a la guerra?

Como WebQA no maneja esta forma de expresar las preguntas, en base a un contexto, se requiere el trabajo extra de reformularlas para que tengan la información incorporada, por ejemplo: *¿En qué año terminó la segunda guerra mundial?* Pero esto agrega más distorsión a la prueba, no sólo se debe traducir la pregunta al español, sino además reformularla para que incluya el contexto. En resumen, para utilizar el mismo conjunto de preguntas de prueba:

- Se tiene que tener el catálogo de preguntas.
- Se tiene que usar datos de las TREC anteriores al 2004, o el trabajo extra de reformular las preguntas.
- Se tienen que traducir al español las preguntas de las TREC. Para las CLEF se puede conseguir el catalogo en español porque es uno de los lenguajes objetivo.
- Para los sistemas online se puede realizar la evaluación con el mismo conjunto de preguntas que se uso en WebQA. Para esto se debe acceder a los sitios Web de los distintos sistemas y someter cada una de las preguntas de forma interactiva, para luego evaluar el resultado obtenido. Tanto el trabajo de realizar las preguntas como de evaluar su correctitud tiene un costo muy alto.

En vista del análisis costo-beneficio se descartan las opciones de comparar WebQA con los sistemas online, además se descarta la alternativa de comparar WebQA con los sistemas participantes en las TREC utilizando su conjunto de preguntas. Para los sistemas participantes en las CLEF se tiene un catálogo de preguntas de la edición 2003 en español con sus respectivas respuestas. Este catálogo será utilizado tanto para comparar WebQA con los sistemas de las CLEF de la edición 2003, como con los sistemas participantes en las TREC de la edición 2005. El sistema de Alejandro del Castillo también fue evaluado con este catálogo de preguntas, con lo cual la comparación

con este sistema es la más directa: el mismo lenguaje, el mismo catálogo de preguntas y el mismo repositorio de información ¹.

5.4.2. Comparación de WebQA con sistemas participantes en las TREC

Las TREC tienen como objetivo, entre otros, definir un marco que permita evaluar sistemas de recuperación de información y en particular de Búsqueda de Respuestas. Para esto se utilizan distintas medidas de performance. El catalogo de preguntas de prueba se clasifican en tres categorías: fácticas, de enumeración y de definición. Cada categoría tiene sus medidas de performance y cada sistema se evalúa combinando las evaluaciones en cada categoría de preguntas.

Como WebQA se centra en preguntas fácticas vamos a comparar la performance solo en esta categoría. En las TREC se evalúan los sistemas en base a la precisión y dos indicadores más: NIL Prec y NIL Recall que miden la Precisión y el Recall de las preguntas que no tienen respuesta, donde Recall se define como la capacidad del sistema de devolver datos relevantes.

$$Recall = \frac{N^{\circ} \text{ documentos relevantes recuperados}}{N^{\circ} \text{ documentos relevantes colección}}$$

Estos dos indicadores no tienen sentido para nuestro sistema, ya que recupera información de la Web y no se puede determinar cuando existe respuesta a una pregunta.

En las TREC 2005 los sistemas brindan una única respuesta a las preguntas fácticas. Siguiendo esta forma de evaluación con el conjunto de preguntas de prueba de las CLEF 2003 A.4 WebQA obtuvo el resultado que se presenta en el cuadro 5.6. En el cuadro 5.7 se muestra el resultado de los sistemas participantes en las TREC 2005 [24] vs. el resultado de WebQA.

Como se puede apreciar, el mejor sistema de las TREC tiene una precisión de 0,713 mientras que WebQA tiene una precisión de 0,385. Sin embargo existe una brecha entre los dos primeros sistemas de las TREC y el resto, por lo que WebQA queda en el tercer lugar del ranking. Los dos sistemas que están sobre el resto aplican técnicas de PLN de nivel semántico y si bien lideran en las preguntas de tipo facticas, no lo hacen para las preguntas de enumeración o definición.

¹Aunque cabe aclarar que la Web del 2005 no es la misma que la del 2007 ya que la cantidad de información ha crecido y los buscadores han mejorado su rendimiento.

Tipo pregunta	MRR	Precisión
Cómo	0,2273	0,1818
Cuál	0,6250	0,6000
Cuándo	0,6250	0,5000
Cuántos	0,3700	0,2400
Dónde	0,6528	0,5833
Indefinido	0,3141	0,2949
Quién	0,5776	0,5517
Total	0,4845	0,3850

Cuadro 5.6: Precisión y MRR de WebQA con el conjunto de preguntas de las CLEF 2003 A.4 y criterio de evaluación de las TREC 2005.

Sistema	Organización	Precisión
lcc05	Language Computer Corp.	0,713
NUSCHUA 1	National Univ. of Singapore	0,666
WebQA	FING - UdelaR 2006	0,385
IMB05L3P	IBM T.J. Watson Research	0,326
ILQUA2	Univ. of Albany	0,309
Insun05QA 1	Harbin Inst. of Technology	0,293
csail2	MIT	0,273
FDUQA14B	Fudan University	0,260
QACTIS05v2	National Security Agency (NSA)	0,257
mk2005qar2	Saarland University	0,235
Edin2005b	Univ. of Edinburgh	0,215

Cuadro 5.7: Precisión de los sistemas presentados en las TREC 2005 vs. WebQA, utilizando como conjunto de preguntas el de las CLEF 2003 A.4, y como criterio de evaluación el de las TREC 2005.

Cabe recalcar que esta comparación es la que tiene mayor incertidumbre de las realizadas pues el repositorio de información, el idioma, y el conjunto de preguntas de evaluación son distintos.

5.4.3. Comparación de WebQA con los sistemas presentados en las CLEF

Muchos de los sistemas de las CLEF son multilingües y se analiza su performance para cada idioma, para el caso de WebQA solo interesa el rendimiento para el idioma español. Esta comparación es más interesante que con los sistemas de las TREC pues se utiliza el mismo conjunto de preguntas

de evaluación.

Para esta edición se considera correcta una respuesta si se encuentra dentro de las primeras tres respuestas para cada pregunta, teniendo esto en cuenta el resultado de WebQA fue el que se presenta en el cuadro 5.8

Tipo Pregunta	MRR	Precisión
Cómo	0,2273	0,2727
Cuál	0,6250	0,6500
Cuándo	0,6250	0,7500
Cuántos	0,3700	0,4800
Dónde	0,6528	0,7500
Indefinido	0,3141	0,3205
Quién	0,5776	0,5862
Total	0,4845	0,5442

Cuadro 5.8: Precisión y MRR de WebQA con el conjunto de preguntas y criterio de evaluación de las CLEF 2003

El resultado de las CLEF [17] Vs. WebQA con el criterio de evaluación mencionado anteriormente es el que figura en el cuadro 5.9.

Sistema	Organización	MRR	Precisión
WebQA	FING - UdelaR 2006	0,4845	0,5442
ITC-irst	Trento, Italia	0,4220	0,5050
ISI	U. of Southern California, USA	0,3280	0,3850
DLSI-UA	U. de Alicante, España	0,3200	0,4350
UVA	U. de Amsterdam, Holanda	0,3350	0,4450
DLTG	U. de Limerick, Irlanda	0,1200	0,1650

Cuadro 5.9: Precisión de los sistemas participantes de las CLEF 2003 vs WebQA.

En comparación con los sistemas participantes en las CLEF 2003 WebQA es el que tiene mejor rendimiento. La mayor pareja MRR-Precisión obtenida en las CLEF fue de (0.442,0.505) mientras que WebQA alcanzó los valores de (0.484, 0.544)

Cabe aclarar que para las preguntas dependientes del tiempo en que se efectúan, como por ejemplo “¿Quién es el presidente de Rusia?”, se tomo como respuesta correcta la respuesta de la CLEF. Además para las 20 preguntas que no tenían respuesta en las CLEF se evaluó si la respuesta dada por WebQA era correcta o no. El documento que contiene el catálogo de preguntas de las CLEF con sus respuestas esta disponible online, por lo tan-

to para muchas de las preguntas se recuperó la respuesta correcta de este documento. El criterio adoptado fue obviar esa respuesta y continuar con la siguiente del ranking. Como se mencionó anteriormente, los sistemas de las CLEF son multilingües, por lo cual probablemente algunos de ellos tienen un rendimiento superior en su idioma nativo. Más allá de estas salvedades el resultado de esta comparación fue más que satisfactorio.

5.4.4. Comparación de WebQA con el sistema de Alejandro del Castillo

El sistema de Alejandro del Castillo así como dos sistemas presentados en su documento de maestría [9] fueron evaluados con el conjunto de preguntas de las CLEF 2003. Estos tres sistemas fueron realizados para el idioma español, el de Alejandro del Castillo utiliza la Web como repositorio de información mientras que los dos restantes utilizan el corpus de documentos proporcionado por la CLEF. Los resultados obtenidos por estos sistemas se encuentran en el cuadro 5.10 [9].

Sistema	MRR	Precisión
WebQA	0,4845	0,5442
INAOE (Pérez)	0,3958	0,4250
INAOE (Del Castillo)	0,3580	0,4150
Alicante	0,3075	0,4000

Cuadro 5.10: Resultados comparativos de sistemas BR en Español.

Luego de observar los datos de el cuadro 5.8 se puede afirmar que WebQA tiene un rendimiento superior a los sistemas presentados en el cuadro 5.10.

5.5. Problemas detectados

Algunos problemas que se detectaron, tanto en la etapa de Ngramas, como de Filtrado se deben a determinados patrones que tienen los documentos disponibles en la Web.

- Un problema importante, es en las preguntas de tipo “Cuándo”, la confusión que causan los fragmentos recuperados de foros, donde en la firma de los mismos aparece la fecha de publicación. En la etapa de filtrado puede producir que un fragmento que debería filtrarse, porque no contiene ninguna fecha, no se filtre debido a que incluye la fecha

de publicación. Luego en la técnica de Ngramas, esta fecha se podría retornar como respuesta si el buscador retornó varios fragmentos en la misma fecha, o en un mismo año.

- Otro inconveniente es que, ante una consulta de dos o más términos, Google presenta con puntos suspensivos los enunciados que contienen cada uno de los términos, sin que estén relacionados en algunos casos. Por ejemplo, ante la consulta: “la independencia del Uruguay” + “año”, dos de los resultados obtenidos fueron:
 - ...en mérito al centenario de la independencia del Uruguay y al destacado rol ...
 - En 1962 En Europa durante este año un grupo musical británico denominado ...

Si bien en este caso ambos enunciados no están estrechamente ligados, en otros casos si se refieren a lo mismo y ayudan a dar la respuesta:

- El 12 de octubre se evoca la fecha en que la expedición de Cristóbal Colón llegó a las costas de... .
- En España se celebra como Fiesta de la Raza desde 1915,

Por lo cual no se puede tomar un criterio inmediato sobre si tener en cuenta o no estos fragmentos. Se requeriría un análisis contextual para resolver su relación y en la mayoría de los casos habría que ir al documento completo.

- Otro problema encontrado se produce en la detección de nombres propios. Por ejemplo dada la pregunta “¿Dónde nació Artigas?”, la respuesta esperada es el nombre propio del lugar. Sin embargo en WebQA se maneja análisis morfológico, pero no se dispone de un reconocimiento de entidades, por ejemplo países, ciudades, lugares, personas, etc. En este ejemplo en la mayoría de los textos aparece el nombre completo “José Gervasio Artigas”, tanto “José”, como “Gervasio” son nombres propios, por lo cual, si se diera el caso de que se repitiera más veces “José” o “Gervasio” que “Montevideo”, la técnica de ngramas podría retornar como respuesta “José Gervasio”. De todas maneras no es algo que influya fuertemente en la calidad general de los resultados retornados por el sistema, son contadas las ocasiones en que ocurre este problema.

5.6. Conclusión

La comparación mas importante es la realizada con los sistemas participantes en las CLEF 2003 y el sistema de Alejandro del Castillo pues se

utiliza el mismo conjunto de preguntas para la evaluación. El rendimiento de WebQA como se observó anteriormente es superior a estos sistemas y muy satisfactorio.

El sistema de Alejandro del Castillo además utiliza la Web como repositorio de información con lo cual la comparación tiene más valor.

A pesar de que los resultados de WebQA fueron mejores al de este sistema, la mayor performance de WebQA puede deberse a que los buscadores mejoraron significativamente su rendimiento y a que la Web tiene mayor información y por ende mayor redundancia, que es uno de los pilares en los que se basan ambos sistemas.

Capítulo 6

Conclusiones y trabajos a futuro

6.1. Conclusiones

El objetivo de este proyecto plantea construir un sistema de búsqueda de respuestas para el idioma español, utilizando la Web como fuente de información. Este tipo de sistemas localizan y extraen respuestas concretas, a partir de preguntas precisas y un análisis sobre un conjunto de documentos escritos en lenguaje natural. Es un área que está en plena investigación y desarrollo y que aún no se ha difundido masivamente. La mayor parte de las investigaciones son para el idioma inglés, pero hay otras independientes del lenguaje. Es poco lo que se ha hecho específicamente para el español.

El grado de complejidad del sistema depende en gran medida del tipo de preguntas a responder. Entre otras se pueden diferenciar las preguntas fácticas, las de recopilación de información, preguntas dentro de un contexto determinado, de un tema específico, las que requieren de cierto análisis profesional, etc. Este proyecto está enfocado en responder las preguntas de tipo fácticas.

Se pueden dividir las conclusiones en tres grupos: las que se derivan del rendimiento del sistema, las que se derivan del uso de la Web como fuente de información y las que se derivan de la arquitectura y herramientas utilizadas.

La principal conclusión sobre el rendimiento del sistema es que se obtuvieron buenos resultados explotando la Web. Para las preguntas fácticas se encontró la respuesta correcta en los primeros tres fragmentos presentados para el 70% del conjunto de preguntas de prueba utilizado. El rendimiento

de WebQA es muy bueno comparado con otros sistemas y superior en la mayoría de los casos. Sin embargo, hay que tener en consideración que en casi todas las comparaciones es distinto el repositorio de información y/o el conjunto de preguntas de evaluación. De todas formas, la comparación con el sistema en español de Alejandro del Castillo del año 2005 es adecuada pues se utiliza el mismo repositorio de información y el mismo catálogo de preguntas, sin despreciar que la Web y los motores de búsqueda crecieron mucho del 2005 a esta parte.

Las preguntas que el sistema responden bien son las precisas, la forma en que el usuario formula la pregunta puede tener un impacto importante en la respuesta obtenida. Esto es algo no deseado en sistemas de *búsqueda de respuestas*, sin embargo la solución que presenta WebQA tiene la ventaja de ser escalable, con lo cual, mediante el agregado de nuevos tipos de pregunta y de nuevas reformulaciones, se pueden contemplar nuevos casos.

La redundancia de la Web evita la necesidad de utilizar técnicas sofisticadas de PLN y recursos léxicos importantes y es suficiente para que se puedan encontrar la mayoría de las respuestas en sus formulaciones más naturales. Sin embargo tiene algunas desventajas: WebQA analiza sólo los fragmentos y no el documento entero (página Web) para tener mejores tiempos de respuesta. Muchas de las respuestas incorrectas el contenido de los fragmentos no es suficiente en cantidad y calidad para poder responderlas correctamente con las técnicas utilizadas. Muchas veces también incluyen datos irrelevantes, como por ejemplo, detalles particulares de los foros de noticias. Por otro lado, la Web no es una fuente de información completamente fiable. La confiabilidad en la respuesta retornada está dada por que ésta se repite muchas veces y en diferentes lugares.

Sobre la recuperación de documentos en la Web, es sencillo lograrlo mediante la utilización de los servicios que brindan los grandes motores de búsqueda como Google o Yahoo. Sin embargo hay que ver la tendencia que siguen en este sentido. En diciembre del 2006 Google decidió dejar de ofrecer el servicio de búsqueda mediante SOAP para nuevos usuarios. La calidad y cantidad de información recuperada de la Web depende en gran medida de los sistemas de recuperación de información, y en nuestro caso de que esas herramientas sean libres y gratuitas.

Sobre la arquitectura del sistema y las herramientas utilizadas, se puede concluir que el uso de UIMA favorece el desarrollo pues permite construir una arquitectura de componentes que representan intuitivamente las etapas clásicas de los sistemas de *búsqueda de respuestas*, permitiendo además reemplazarlos y combinarlos obteniendo nuevas versiones del sistema. Una de las ventajas potenciales que UIMA promete es la utilización de componentes de procesamiento de lenguaje natural desarrollados por terceros. Lamentable-

mente el *framework* no tiene aún una cantidad importante de componentes libres que permitan utilizarse.

Los sistemas de *búsqueda de respuestas* utilizan en mayor o menor medida herramientas de procesamiento de lenguaje natural. En particular, WebQA utiliza un etiquetador de categorías gramaticales. En la evaluación se comprobó que el rendimiento de este componente incide en el rendimiento del sistema. Por otro lado, existen pocas herramientas de procesamiento de lenguaje natural para el español, libres y gratuitas, siendo algunas de ellas de baja calidad. Por lo tanto, la disponibilidad de estos componentes es otro factor a tener en cuenta.

En conclusión WebQA cumplió con los objetivos planteados al inicio del proyecto y obtuvo un resultado más que aceptable. El nivel de técnicas utilizado se basa en el supuesto de redundancia de información. Muchas de las respuestas incorrectas se deben a que este supuesto falla. Sin embargo la posibilidad de analizar el documento entero o agregar otras herramientas, como por ejemplo un reconocedor de entidades, pueden mejorar el rendimiento. Debido a la arquitectura que el sistema ofrece, y a la relativa independencia de sus etapas, se pueden crear nuevos componentes que apliquen otro tipo de técnicas o que resuelvan otros tipos de pregunta diferentes de las fácticas.

6.2. Trabajos a futuro

WebQA no responde bien preguntas en las cuales se busca un tipo de entidad en particular, por ejemplo “En qué ciudad”, o “En qué país”. En la etapa de análisis de la pregunta, por medio de los patrones, es posible identificar a que entidad corresponde la respuesta esperada. Sin embargo, tanto en la etapa de filtrado y ponderación de fragmentos, como en la de extracción de la respuesta por ngramas, solamente se distinguen categorías gramaticales y no entidades. Para el caso del ejemplo, se espera como respuesta un nombre propio y no una ciudad o un país. Si se identificaran entidades como personas, ciudades, países, etc., el sistema sería capaz de responder de forma más precisa y correcta éste tipo de preguntas. Las herramientas que se pueden utilizar son los *reconocedores de entidades* y *diccionarios*.

Por otro lado la performance esta limitada por el contenido de los fragmentos, por lo cual analizar el documento completo ayudaría a mejorar la respuesta en determinadas situaciones. El escenario en el cual se podría consultar el documento entero puede ser:

- Poca cantidad de fragmentos.

- Que no exista una respuesta destacada en el análisis de ngramas.
- En el caso de fragmentos en el que se detecte cierta ambigüedad, por ejemplo, que se encuentre el texto separado por puntos suspensivos.

Para analizar el documento completo se tendrían que usar técnicas que consideren el contexto del fragmento. El caso de tener poca cantidad de fragmentos, hace pensar que se requiere un análisis del documento completo aplicando un nivel de técnicas de PLN mayor, pues no se tiene redundancia de información, sino que se debe explotar con mayor intensidad la información disponible.

El uso de un diccionario de sinónimos podría traer resultados positivos en la etapa de filtrado de fragmentos o también en la etapa de reformulación de la pregunta.

Para las preguntas de categoría “Cuánto” se tiene un desempeño inferior al resto. Esto se debe a que ngramas se aplica sobre los ngramas del texto sin llevar a su forma canónica. Por ejemplo el ngrama “doce” y el ngrama “12” son analizados como ngramas distintos. Además, diferente información puede ser considerada válida. Por ejemplo ante la pregunta “¿Qué altura tiene el Cerro Catedral?”, tanto la respuesta 513 (metros), como 514 son similares y ambas pueden ser consideradas correctas. Sin embargo ngramas las analizará como respuestas separadas. Para esto se podría incluir alguna funcionalidad que mida la similaridad de los ngramas.

Otra funcionalidad a agregar sería la de manejar la especificidad de conceptos, por ejemplo ante la pregunta “¿En qué lugar se encuentra el cerro Catedral?” la respuesta “Maldonado” es más específica que “Uruguay”. Este problema se encuentra en todos los sistemas de *búsqueda de respuestas* y tiene un costo alto su resolución, pues se requieren ontologías para representar los conceptos y la relación (específico, general) entre ellos.

Glosario

Análisis morfológico Consiste en determinar la forma, clase o categoría gramatical de cada palabra de una oración.

APIS Interfaz de programación de aplicaciones (Applications Programming Interface): una serie de funciones que están disponibles para realizar programas para un cierto entorno.

Búsqueda de Respuestas (Question & Answering) Herramientas capaces de obtener respuestas concretas a necesidades de información muy precisas, en base a la comprensión de preguntas y a partir del análisis de documentos escritos en lenguaje natural.

Determinantes Tipo de palabra funcional que está estrechamente vinculada al nombre, por ejemplo “El”, “La”.

EuroWordNet Sistema de lexicones semánticos para idiomas Europeos.

Expansión de la pregunta Proceso que consiste en añadir al conjunto de términos originales de la pregunta, aquellos otros términos que pueden utilizarse para expresar las mismas ideas o conceptos.

Expresión Regular Es una forma de representar a los lenguajes regulares (finitos o infinitos) y se construye utilizando caracteres del alfabeto sobre el cual se define el lenguaje.

Extracción de Información (EI) Consiste en extraer, de un texto o un conjunto de textos, entidades, eventos y relaciones entre ellos.

Fragmento Segmento de un documento con significado propio.

Intejecciones Palabras o frases cortas que se emplean principalmente en el lenguaje oral, y en el escrito que reproduce el lenguaje oral, muchas veces como expresión expletiva de desahogo o explosiones de emotividad, a menudo de modo casi reflejo.

Mean Reciprocal Rank (MRR) Media del recíproco del índice en el cual se presenta la respuesta correcta dentro los n primeros fragmentos presentados.

Motor de Búsqueda Sistema informático que indexa archivos almacenados en servidores web para luego poder recuperarlos en base a palabras clave o árboles jerárquicos.

NGrama Subsecuencia de n caracteres obtenidos de una cadena de caracteres.

Nombre Común Designan un grupo de clases (por ejemplo “hombre”).

Nombre Propio Designan entes individuados y únicos (por ejemplo “Pedro”), suelen escribirse con letra inicial mayúscula.

Palabra funcional Son palabras pequeñas o frecuentemente usadas que generalmente son obviadas por los motores de búsqueda.

Patrón Representación abreviada de la secuencia consenso de un motivo, por ejemplo, expresiones regulares.

POS-tagger Programa de etiquetado léxico cuya finalidad es la de indicar que función cumple cada una de las palabras dentro de la oración.

Precisión Dado un conjunto de preguntas es la proporción de éstas que fue respondida correctamente.

Preguntas fácticas Preguntas que requieren una única respuesta concreta y correcta.

Procesamiento de Lenguaje Natural (PLN) Se ocupa de la formulación e investigación de mecanismos eficaces computacionalmente para la comunicación entre personas o entre personas y máquinas por medio de lenguajes naturales.

Pronombre Categoría gramatical de aquellas palabras que señalan o sustituyen a otras que normalmente ya se han nombrado.

Pronombre Interrogativo Pronombres propios de la interrogación o que sirven para preguntar.

Realimentación (Relevance Feedback) Técnica que consiste en enriquecer la pregunta inicial realizada por el usuario, mediante la utilización de información relevante obtenida de los documentos recuperados inicialmente.

Recall Cantidad de documentos relevantes recuperados sobre cantidad de documentos relevantes en la colección

Reconocedor de Entidades (Name Entity Tagger) Sistema capaz de dedetectar y clasificar entidades nombradas en fragmentos de texto.

Recuperación de Información (RI) Conjunto de tareas mediante las cuales el usuario localiza y accede a los recursos de información que son pertinentes para la resolución del problema planteado.

Snippet Pequeño segmento de un documento retornado por los buscadores web.

SOAP Siglas de Simple Object Access Protocol. Protocolo estándar creado por el W3C que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

Término clave Es aquel término que es de relevancia dentro de una pregunta o fragmento, por ejemplo, el verbo principal de una pregunta.

Verbo Categoría gramatical que funciona como núcleo del predicado y suele indicar movimiento, acción o estado.

WordNet Lexicón semántico para el idioma inglés.

XML Acrónimo del inglés eXtensible Markup Language (lenguaje de marcado ampliable o extensible).

Bibliografía

- [1] Sistemas de question answering, Abril 2007.
<http://question-answering.galeon.com/Question-Answer.html>.
- [2] Eric Brill, Susan Dumais, and Michele Banko. An analysis of the askmsr question-answering system. In *In Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 257–264, Morristown, NJ, USA, June 2002. Association for Computational Linguistics.
- [3] Eric Brill, Jimmy Lin, Michele Banko, Susan Dumais, and Andrew Ng. Data-intensive question answering. In *Text Retrieval Conference*, Redmond, Estados Unidos, Julio 2002.
- [4] CLEF. Cross language evaluation forum (clef), Abril 2007.
<http://www.clef-campaign.org/>.
- [5] Richard J Cooper and Stefan M Ruger. A simple question answering system. In *Proceedings of The Ninth Text REtrieval Conference (TREC 9)*, Londres, Inglaterra, Agosto 2000.
- [6] M. Perez Coutino. The use of lexical context in question and answering for spanish. In *Text REtrieval Conference*, pages 0–, 2004.
- [7] Cesar de Pablo-Sanchez, Ana Gonzalez-Ledesma, Jose Luis Martinez-Fernandez, Jose Maria Guirao³, Paloma Martinez, and Antonio Moreno. Miracle’s 2005 approach to cross-lingual question answering. In *Text REtrieval Conference*, pages 0–, 2005.
- [8] EAGLES. Eagles, Abril 2007.
<http://www.ilc.cnr.it/EAGLES96/home.html>.
- [9] Alejandro Del Castillo Escobedo. Busqueda de respuestas mediante redundancia en la web. Master’s thesis, INAOE: Intituto Nacional de Astrofisica, Optica y Electronica, Puebla, Mexico, Febrero 2005.

- [10] EuroWordNet. Eurowordnet, Abril 2007.
<http://www.illc.uva.nl/EuroWordNet/>.
- [11] Freeling. Freeling, Abril 2007.
<http://garraf.epsevg.upc.es/freeling>.
- [12] José Luis Vicedo González. Recuperación de información de alta precisión: Los sistemas de búsqueda de respuestas. Alicante, España, Julio 2003.
- [13] Google. Google apis, Abril 2007.
<http://code.google.com/apis.html>.
- [14] E. Hovy, L. Gerber, U. Hermjakob, M. Junk, and C. Lin. Question answering in webclopedia. In *Proceedings of The Ninth Text REtrieval Conference (TREC 9)*, Maryland, Estados Unidos, Noviembre 2000.
- [15] IBM. Unstructured information management architecture (uima) sdk user's guide and reference. Junio 2006.
- [16] L. Kosseim, L. Plamondon, and L.J. Guillemette. Answer formulation for question-answering. In *Proceedings of The Sixteenth Canadian Conference on Artificial Intelligence (AI'2003)*, Halifax, Canada, Junio 2003.
- [17] Bernardo Magnini, Simone Romagnoli, Alessandro Vallin, Jesús Herrera, Anselmo Peñas, Víctor Peinado, Felisa Verdejo, and Maarten de Rijke. The multiple language question answering track at clef 2003. In *CLEF*, pages 1–12, Trondheim, Noruega, Agosto 2003.
- [18] Sun Microsystems. Java platform standard edition 5.0, Abril 2007.
<http://java.sun.com/j2se/1.5.0>.
- [19] OpenNLP. Opennlp, Abril 2007.
<http://opennlp.sourceforge.net>.
- [20] Luis Villaseñor Pineda, Manuel Montes y Gómez, and Alejandro del Castillo. Búsqueda de respuestas basada en redundancia: un estudio para el español y el portugués. Puebla, Mexico, Noviembre 2004.
- [21] TREC. Text retrieval conference (trec), Abril 2007.
<http://trec.nist.gov/>.
- [22] Vicedo, Jose Luis Llopis, and Fernando Ferrández Antonio. University of alicante experiments at trec-20021. In *University of Alicante Experiments at TREC-20021*, pages 0–, 2002.
- [23] Ellen M. Voorhees. Overview of trec 2004. In *TREC*, pages 1–12, Maryland, Estados Unidos, Noviembre 2004.

-
- [24] Ellen M. Voorhees. Overview of the trec 2005 question answering track. In *TREC*, pages 1–11, Maryland, Estados Unidos, Mayo 2005.
- [25] WordNet. Wordnet, Abril 2007.
<http://wordnet.princeton.edu/>.
- [26] Yahoo. Yahoo, Abril 2007.
<http://developer.yahoo.com/search/>.
- [27] Zhiping Zheng. Answerbus question answering system. In *Proceeding of HLT Human Language Technology Conference (HLT 2002)*, pages 24 – 27, San Diego, Estados Unidos, Marzo 2002.

Apéndice A

Apéndice Evaluación

A.1. Evaluación del impacto de la variación de Parámetros

Esta sección busca mostrar cuales son los parámetros que tienen mas incidencia en el sistema y cuales son sus valores óptimos. Se considera el rendimiento del sistema como un análisis de costo y beneficio que involucra no solo medidas de precisión sino también el tiempo de respuesta. Para este análisis se tomó la mejor combinación de componentes (Freeling-Google) obtenida en la evaluación del sistema (capítulo 5), y una configuración de parámetros que empíricamente se determinó como “óptima”, que llamaremos *configuración estándar*. Es de esperarse que el resultado del sistema con las variaciones de parámetros sea inferior en todos los casos al de la configuración estándar.

Debido al alto costo de evaluar las respuestas se tomó un conjunto reducido de las preguntas de evaluación general del sistema (50 preguntas) elegidas al azar A.2. Con este conjunto reducido de preguntas se seleccionaron las variaciones de parámetros que pudieran aportar mas valor (se acotaron debido a la misma razón de alto costo de evaluación) y se evaluó el sistema para cada una de ellas.

En este análisis, los parámetros se dividen en grupos. La descripción de cada parámetro del sistema se puede ver en las Tablas A.1 y A.2.

En primera instancia se hizo una corrida con la configuración estándar sobre el conjunto de 50 preguntas. Los resultados obtenidos fueron un poco mejores a los de la evaluación con el catalogo completo de preguntas. Se evaluaron los resultados según la precisión de la respuesta extraída y al MRR

Descripción grupo de parámetros	Parámetros	Valor por defecto
Cantidad de fragmentos retornados por el buscador.	Cantidad	10
Tamaño de la ventana de análisis de los fragmentos.	Tamaño de la ventana	8
Constantes de la fórmula de ponderación de los fragmentos (pto. 3.3).	C1 (valor a multiplicar por la cantidad de tokens de la pregunta que están en el fragmento y en la ventana de éste)	1
	C2 (valor a multiplicar por la cantidad de tokens de la pregunta que están en el fragmento, pero fuera de la ventana de éste)	0,5
Ponderación de las reformulaciones:		
Reformulador en base a la permutación de las dos primeras palabras (pto. 3.2.2).	Todas las palabras.	1,6
	Sin la primera palabra.	1,5
	Sin la primera y segunda palabra.	1,4
	Primera palabra al final.	1,5
	Primera y segunda palabra al final.	1,6
Reformulador en base a conjunción de palabras clave (pto. 3.2.3).	Puntaje para reformulación máxima.	1,1
	Puntaje para reformulación mínima.	1
Reformulador según patrones establecidos (dependiendo de X, V, Y) (pto. 3.2.1).	X V Y X Y V ...	1,6
	X + V Y X + Y V ...	1,5
	X + V + Y	1,4

Cuadro A.1: Parámetros que influyen en el filtrado y ponderación de fragmentos.

Descripción grupo de parámetros	Parámetros	Valor por defecto
Cantidad mínima de fragmentos para realizar el análisis de ngramas.	Cantidad mínima de fragmentos	8
Puntaje de Ngramas (pto. 3.4.1).	P2 (puntaje inicial de ajuste de cada ngrama)	5
	P3 (puntaje extra que se suma por cada token del ngrama si la longitud es mayor a 1).	2
	P4 (puntaje extra por cada token del ngrama que pertenezca a la ventana del fragmento).	1
Solapamiento de Ngramas (pto. 3.4.2).	Cantidad (determina hasta que ngrama se intentará solapar con el primero, según el orden por puntaje.).	3
	Porcentaje (indica que para que se intenten solapar los ngramas n1 con ni, el puntaje de ni, deberá ser mayor a dicho porcentaje del puntaje de n1).	60

Cuadro A.2: Parámetros que influyen en el análisis de ngramas 3.4.

calculado a partir los fragmentos retornados.

La precisión y el MRR fueron los siguientes:

- Precisión Extracción Respuesta: 0,64
- MRR: 0,72

Luego se inició con la variación de los parámetros, la que se hizo por grupo, es decir, para evaluar se varían los parámetros de un grupo manteniendo el resto igual a la configuración estándar. En el cuadro A.3 se muestra cada variación y el impacto resultante.

Las variaciones de los parámetros tuvieron un impacto bajo en el sistema, como se puede apreciar en el cuadro A.3, tanto el porcentaje de respuestas correctas como el MRR no cambiaron significativamente. Se pudo determinar que las variaciones de parámetros impactan principalmente en las preguntas para las cuales se tienen varias respuestas candidatas con puntaje similar. Los parámetros que tuvieron más relevancia fueron el tamaño de la ventana y la cantidad máxima de fragmentos a recuperar en cada consulta a los buscadores. El primer resultado se debe a que la mejor ventana debe haber al menos una palabra del tipo de respuesta esperado y una palabra de la pregunta. Si esta ventana no existe se descarta el fragmento. Por lo tanto con tamaño de ventana bajo, es más difícil que se cumpla esta condición. El segundo parámetro se creó para limitar la cantidad de fragmentos retornados por los buscadores ante una consulta, de forma tal de tener una performance adecuada. Con poca cantidad de fragmentos a recuperar no se tiene una cantidad de fragmentos significativa para analizar. Con mayor cantidad de fragmentos aumenta la precisión pero no el desempeño general del sistema ya que los tiempos de respuesta aumentan.

Tamaño de la ventana de análisis de los fragmentos

Variar el tamaño de la ventana influye negativamente si éste es muy pequeño o muy grande. Se obtuvieron peores resultados con tamaño de ventana 4, en el orden del 8% peor. Esto es porque en la mejor ventana debe haber al menos una palabra del tipo de respuesta esperado y una palabra de la pregunta. Si esta ventana no existe se descarta el fragmento. Por lo tanto con tamaño de ventana bajo, es más difícil que se cumpla esta condición, por lo que se descartan una mayor cantidad de fragmentos. A medida que se aumenta el tamaño de la ventana los resultados van mejorando, pero hasta cierto punto donde empiezan a empeorar. Esto es lógico por la idea de la ventana: tomar una medida de cuán lejos están las palabras de la pregunta y el tipo de respuesta esperado entre sí. Si se tuviera una ventana tan grande

Tamaño de la ventana de análisis de los fragmentos													
N° Corrida	Parámetros											Impacto NGr.	MRR
	Tamaño Ventana												
1	4											Medio	0,7092
2	12											Bajo	0,7177
Default	8												0,7160
Cantidad de fragmentos retornados por Google													
N° Corrida	Parámetros											Impacto NGr.	MRR
	Cantidad de fragmentos												
1	5											Medio	0,6972
2	20											Bajo	0,7177
Default	10												0,7160
Constantes de la fórmula de ponderación de los fragmentos													
N° Corrida	Parámetros										Impacto NGr.	MRR	
	C1					C2							
1	4					0,5					Bajo	0,7279	
2	1					1					Bajo	0,7126	
3	8					0,5					Bajo	0,7126	
Default	1					0,5						0,7160	
Ponderación de las reformulaciones													
N° Corrida	Parámetros (orden tabla A.1)											Impacto NGr.	MRR
	Dos primeras palabras					Pal. clave		Patrones					
1	1	1	1	1	1	1	1	1	1	1	1	Bajo	0,6935
2	4	3	2	3	4	1,1	1	4	3	2		Bajo	0,7065
Default	1,6	1,5	1,4	1,5	1,6	1,1	1	1,6	1,5	1,4			0,7160
Cantidad mínima de fragmentos para realizar el análisis de ngramas													
N° Corrida	Parámetros											Impacto NGr.	MRR
	Cantidad mínima de fragmentos												
1	3											Bajo	0,7320
2	18											Bajo	0,7245
Default	8												0,7160
Puntaje de ngramas													
N° Corrida	Parámetros											Impacto NGr.	MRR
	P2			P3				P4					
1	5			0				0				Bajo	0,6867
2	1			5				1				Bajo	0,6752
3	1			1				5				Bajo	0,7059
4	1			5				5				Bajo	0,7059
5	1			2				2				Bajo	0,7059
Default	5			2				1					0,7160
Solapamiento de ngramas													
N° Corrida	Parámetros											Impacto NGr.	MRR
	Cantidad					Porcentaje							
1	1					0						Bajo	0,7269
2	3					20						Bajo	0,7269
3	3					80						Bajo	0,7269
4	6					80						Bajo	0,7269
5	6					58						Bajo	0,7269
6	6					20						Bajo	0,7269
Default	3					60							0,7160

Cuadro A.3: Configuraciones de parámetros evaluadas.

como el fragmento esto no tendría sentido. El valor por defecto es tamaño igual 8. Con ventana de tamaño 12 los resultados decayeron muy levemente, del orden del 2%.

Cantidad de fragmentos obtenidos del buscador en una consulta

Se estableció un límite en la cantidad de fragmentos obtenido por los buscadores en una consulta. Esta restricción se debe a que los fragmentos a analizar deben ser un número razonable para que no se pierda performance de respuesta. A su vez un número muy bajo no es bueno, ya que se deben tener una cantidad de fragmentos considerable para analizar. El valor por defecto es un máximo de 10 fragmentos por consulta. Se probó con 5 fragmentos y el rendimiento disminuyó significativamente (6 % peor). Luego se realizó una corrida con un máximo de 20 fragmentos. En este caso no hubo variación, pero decayó la performance del sistema, ya que se tienen más fragmentos para analizar.

Constantes de la fórmula de ponderación de los fragmentos (pto. 3.3)

La variación de éstas constantes (C_1 , multiplica por la cantidad de palabras de la pregunta en el fragmento y dentro de la ventana; y C_2 , multiplica por la cantidad de palabras de la pregunta en el fragmento pero que quedaron fuera de la ventana) tuvieron un impacto bajo en los resultados generales del sistema. Los valores por defecto son 1 y 0,5 respectivamente. Como era de esperar empeoraron los resultados en el orden del 4 % cuando C_1 y C_2 tomaron valores iguales (1). Con la otra corrida que se hizo ($C_1=4$, $C_2=0,5$) los resultados mejoraron muy levemente, en el orden del 2 %. Debido a esta mejora, se realizó otra corrida con $C_1=8$, $C_2=0,5$, pero en este caso el rendimiento se mantuvo. Por lo tanto es importante que C_1 sea mayor a C_2 , pudiendo concluir que se cumple el objetivo de la fórmula, de dar más prioridad a los fragmentos que tengan la mayor cantidad posible de palabras de la pregunta en su ventana, y en consecuencia cerca entre ellas.

Cantidad mínima de fragmentos para el análisis de ngramas

Aquí se testearon los resultados de los ngramas según la cantidad de fragmentos analizados. Por defecto son un mínimo de 8, pero se probó con mínimos de 3 y 18. En el caso de 3, en la mayoría de los casos no se utilizó la reformulación más general, de palabras claves de la pregunta, ya que la cantidad de resultados obtenidos con los otros reformuladores era suficiente. En este caso empeoró levemente, lo que era de esperar ya que la técnica de ngramas, utiliza como base que la respuesta se repite muchas veces en los fragmentos, cosa que no se puede asegurar si éstos son pocos. En el caso de 18 fragmentos mínimos, prácticamente siempre es necesario efectuar consultas

con los tres reformuladores. Aquí los resultados se mantuvieron, no hubo variación.

Puntaje de ngramas (pto. 3.4.1)

Esta etapa trata de evaluar las constantes P2, P3 y P4 (puntaje inicial de ajuste de cada ngrama, puntaje extra por el largo del ngrama y puntaje extra por cada token del ngrama que pertenezca a la ventana del fragmento) utilizados en el cálculo del puntaje de un ngrama. La principal variación, en la que empeoró significativamente (10%), fue (5,0,0). Aquí no se suma puntaje adicional a los ngramas más largos, y tampoco si un token pertenece a la ventana. Con la configuración (1,5,1) empeoró levemente. Esta configuración implica que se prioriza demasiado a los ngramas más largos, influyendo muy poco que un token pertenezca o no a la ventana. Con el resto de las pruebas se dio la particularidad que el rendimiento mejoró levemente (del orden del 2%) respecto a la configuración estándar (1,2,1) Con esta configuración se priorizan levemente los ngramas más largos (P3), y se asigna un puntaje menor por cada token que pertenezca a la ventana (P4). Con las configuraciones que mejoró ((1,1,5), (1,5,5), (1,2,2))se da un puntaje alto a P4, mientras que P3 toma distintos valores.

Finalmente concluimos que es importante asignar puntaje adicional a los ngramas más largos equiparándose de esa forma con los más cortos, pero lo fundamental es asignar puntaje extra por cada token del ngrama que pertenezca a la ventana. En las dos configuraciones de parámetros que empeoró el rendimiento éste valor era muy bajo e influía muy poco. Esto comprueba la teoría inicialmente planteada: cuanto más cerca estén en un fragmento las palabras de la pregunta con la palabra del tipo de respuesta esperado, será más probable que esta última sea la respuesta.

Solapamiento de ngramas

Con el solapamiento de ngramas se busca obtener respuestas más completas, por lo que los resultados, en cuanto a la cantidad de respuestas correctas, prácticamente no tiene variación. Se mostrarán las diferencias citando ejemplos. La configuración por defecto es solapar los tres primeros ngramas, si sus puntajes no difieren más de un porcentaje X. Estos son los dos parámetros que se variaron, de modo de intentar solapar con más o menos ngramas y con mayor o menor diferencia en sus puntajes. El caso base, es el de no solapar nada, y retornar el ngrama con mayor puntaje. Aquí hubieron diferencias con la configuración por defecto en ciertas preguntas. Por ejemplo para la pregunta “¿Cuándo murió el Che Guevara?” en la configuración por defecto retorna el trigramma “9 octubre 1967”, producto del solapamiento los

ngramas “octubre 1967” y “9 octubre”. Al no solapar el resultado es “octubre 1967”. Casos similares se dan en las preguntas “¿Cuándo comenzó la segunda guerra mundial” y “¿En qué fecha se disolvió la URSS?”, en las que con la configuración estándar responde “1 septiembre 1939” y “25 diciembre 1991” respectivamente, mientras que sin solapar los ngramas las respuestas son “septiembre 1939” y “diciembre 1991” Luego para la configuración (3, 80 %) ocurrió algo similar, ya que la diferencia de puntajes para que se solapen los ngramas tiene que ser menor. Con las configuraciones (3, 20 %) y (6,20 %), se buscó solapar con mayor facilidad. Esto produjo que alguna pregunta sea más completa, pero por otro lado trajo ruido. A continuación se muestran algunos ejemplos de esto:

- ¿Quién fue el primer presidente de Argentina?
 - Default: “Bernardino Rivadavia”
 - (6,20 %): “Además Bernardino Rivadavia”
- ¿Cuándo comenzó la segunda guerra mundial?
 - Default: “1 septiembre 1939”
 - (6,20 %): “día 1 septiembre 1939”
- ¿Cuántos habitantes tiene Perú?
 - Default: “28 millones”
 - (6,20 %): “28 millones 152 mil 256”
- ¿Cuántos departamentos tiene Uruguay?
 - Default: “19”
 - (6,20 %): “tres 19”

Estas son algunas de las diferencias producto de la configuración (6,20 %) Se puede apreciar que en ciertas preguntas trae ruido, o sea, el ngrama incluye la respuesta y también otras palabras que no tienen importancia. Para concluir, veamos un caso particular que se dio para la pregunta “¿Cuántos habitantes tiene Perú?”. La respuesta en la configuración estándar es “28 millones”, pero variando los parámetros a (6,20 %), ésta es “28 millones 152 mil 265”. Este no es un resultado del todo correcto, ya que “152 mil 265” no se corresponde a “28 millones”, si no a un resultado de otro fragmento que es “26 millones 152 mil 265”. 28 millones fue obtenido de fragmentos que indicaban que la población de Perú está cerca de los 28 millones. El solapamiento que se produjo fue el siguiente: “28 millones” + “26 millones” + “millones 152 mil” + “mil 265” + “25 millones” = “28 millones 152 mil 256”.

Ponderación de las reformulaciones

Es importante tener cierta preferencia por los fragmentos recuperados a partir de las reformulaciones más específicas. Por ejemplo ante la pregunta “¿Quién fue el primer presidente de Uruguay?” es más factible que la respuesta esté en un fragmento que contenga “fue el primer presidente de Uruguay” antes que en otro que contenga palabras clave de la pregunta en cualquier orden (“primer” + “presidente” + “Uruguay”). Recordemos que para el cálculo del puntaje de un fragmento, se le multiplica a su puntaje parcial el valor de la reformulación, y luego influye en el puntaje de los ngramas, ya que éste se multiplica por el puntaje del fragmento. Por defecto se le asigna un puntaje levemente mayor a las reformulaciones del reformulador basado en patrones y del reformulador basado en la permutación de las dos primeras palabras. Las que generan éstos son mucho más específicas a las del tercer reformulador que realiza la conjunción de las palabras clave de la pregunta. El objetivo de esta evaluación fue observar dos resultados:

1. Cuanto influye en el sistema asignar mayor prioridad a las reformulaciones más específicas.
2. Cuanta diferencia debe haber entre los puntajes de las reformulaciones más específicas y las más generales.

Para el primer objetivo se asignó puntaje 1 a todas las reformulaciones, de modo que no influyan en el cálculo del puntaje del fragmento. Con este escenario el rendimiento desmejoró levemente, en el orden del 4%. Luego se aumentó significativamente el valor asignado a las reformulaciones de los dos primeros reformuladores. Aquí el rendimiento del sistema se mantuvo. Por lo tanto, es importante dar mayor prioridad a las reformulaciones más específicas respecto a las más generales, aunque no hubo variación al aumentar la diferencia de puntajes entre ellas.

A.2. Conjunto de prueba de 50 preguntas

1. ¿A qué temperatura se funde el oro?
2. ¿Cuál es el idioma más hablado del mundo?
3. ¿Cuál es el libro más famoso de Pablo Neruda?
4. ¿Cuál es el país más grande de Sudamérica?
5. ¿Cuál es el país más grande del mundo?

6. ¿Cuál es la avenida más ancha del mundo?
7. ¿Cuándo asesinaron a Kennedy?
8. ¿Cuándo comenzó la batalla de las piedras?
9. ¿Cuándo comenzó la primer guerra mundial?
10. ¿Cuándo comenzó la segunda guerra mundial?
11. ¿Cuándo fue el primer mundial de fútbol?
12. ¿Cuándo fue la revolución rusa?
13. ¿Cuándo murió el Che Guevara?
14. ¿Cuándo nació Artigas?
15. ¿Cuándo nació el Che Guevara?
16. ¿Cuándo se cayeron las torres Gemelas?
17. ¿Cuándo sucedió la toma de la bastilla?
18. ¿Cuántas veces salió Uruguay campeón del mundo?
19. ¿Cuántos habitantes tiene Perú?
20. ¿Cuántos departamentos tiene Uruguay?
21. ¿Dónde está la Alhambra?
22. ¿Dónde estaba ubicada Chernobyl?
23. ¿Dónde nació Artigas?
24. ¿Dónde nació Carlos Gardel?
25. ¿Dónde nació Jose Gervasio Artigas?
26. ¿Dónde se inventó la pólvora?
27. ¿Dónde se jugaron las primeras olimpiadas?
28. ¿Dónde se jugó el mundial de fútbol de 1962?
29. ¿En qué ciudad se encuentra el famoso Cristo Redentor?
30. ¿En qué continente se encuentra Madagascar?
31. ¿En qué fecha se desarrollo la primer olimpiada de la era moderna?
32. ¿En qué fecha se disolvió la URSS?

33. ¿En qué fecha se produjo el desembarco de los 33 Orientales?
34. ¿En qué país se jugaron los juegos olímpicos de 1992?
35. ¿Quién asesinó a John Lennon?
36. ¿Quién era el hermano de Abel?
37. ¿Quién era el hermano de Caín?
38. ¿Quién es el actual presidente de Estados Unidos?
39. ¿Quién es el autor de Moby Dick?
40. ¿Quién es el conejo de la suerte?
41. ¿Quién es el creador de los Simpsons?
42. ¿Quién es el presidente de Microsoft?
43. ¿Quién escribió rayuela?
44. ¿Quién fue el primer hombre en pisar la luna?
45. ¿Quién fue el primer presidente de Argentina?
46. ¿Quién fue el primer presidente de Estados Unidos?
47. ¿Quién fue el primer presidente de Uruguay?
48. ¿Qué población tiene Francia?
49. ¿Qué superficie tiene Uruguay?
50. ¿Quién inventó el motor eléctrico?

A.3. Conjunto de prueba de 150 preguntas

1. ¿A cuántos kilómetros de Montevideo se encuentra Colonia del Sacramento?
2. ¿A qué edad murió Pablo Neruda?
3. ¿A qué temperatura se funde el oro?
4. ¿Con quién se casó Napoleón?
5. ¿Cuál es el enemigo de Batman?
6. ¿Cuál es el futbolista más caro del mundo?

7. ¿Cuál es el idioma más hablado del mundo?
8. ¿Cuál es el libro más famoso de Pablo Neruda?
9. ¿Cuál es el monte más alto del mundo?
10. ¿Cuál es el país más grande de Sudamérica?
11. ¿Cuál es el país más grande del Mundo?
12. ¿Cuál es el punto más alto de Uruguay?
13. ¿Cuál es el verdadero nombre de Superman?
14. ¿Cuál es la avenida más ancha del mundo?
15. ¿Cuál es la dirección de la embajada Uruguaya en Francia?
16. ¿Cuáles son los planetas del sistema solar?
17. ¿Cuándo asesinaron a Kennedy?
18. ¿Cuándo asesinaron a Zelmar Michellini?
19. ¿Cuándo asumió Tabaré Vázquez?
20. ¿Cuándo comenzó la batalla de las piedras?
21. ¿Cuándo comenzó la primera guerra mundial?
22. ¿Cuándo fue el accidente de chernobyl?
23. ¿Cuándo fue el primer mundial de fútbol?
24. ¿Cuándo fue la depresión del 29?
25. ¿Cuándo fue la primera guerra mundial?
26. ¿Cuándo fue la revolución rusa?
27. ¿Cuándo logró la independencia Uruguay?
28. ¿Cuándo murió Zelmar Michellini?
29. ¿Cuándo murió el Che Guevara?
30. ¿Cuándo nació Artigas?
31. ¿Cuándo nació el Che Guevara?
32. ¿Cuándo obtuvo la independencia Estados Unidos?
33. ¿Cuándo ocurrió la batalla de Trafalgar?

34. ¿Cuándo salió campeón uruguayo Peñarol por última vez?
35. ¿Cuándo se cayeron las torres Gemelas?
36. ¿Cuándo se declaró la independencia en Uruguay?
37. ¿Cuándo se desarrolló la primera olimpiada?
38. ¿Cuándo se fundó el Club Social y Deportivo Villa Española?
39. ¿Cuándo se grabó el disco The Wall de Pink Floyd?
40. ¿Cuándo se hizo el primer trasplante de corazón en Uruguay?
41. ¿Cuándo se hizo el primer trasplante de corazón?
42. ¿Cuándo sucedió la toma de la bastilla?
43. ¿Cuándo terminó la primer guerra mundial?
44. ¿Cuántas veces salió campeón del mundo Nacional?
45. ¿Cuántas copas del mundo ganó Brasil?
46. ¿Cuántas emisoras FM existen en Montevideo?
47. ¿Cuántas medallas de oro ganó Nadia Comaneci?
48. ¿Cuántas personas murieron en la segunda guerra mundial?
49. ¿Cuántas veces salió campeón Uruguay del mundo?
50. ¿Cuántos años vive un elefante?
51. ¿Cuántos campeonatos de formula uno ganó Fangio?
52. ¿Cuántos continentes existen?
53. ¿Cuántos departamentos tiene Uruguay?
54. ¿Cuántos habitantes tiene Perú?
55. ¿Cuántos son los mandamientos Cristianos?
56. ¿De qué color son los pitufos?
57. ¿De qué país son las islas Malvinas?
58. ¿Dónde está la Alhambra?
59. ¿Dónde está ubicada la ciudad de Cuzco?
60. ¿Dónde está ubicada la ciudad de Victoria?

61. ¿Dónde estaba ubicada chernobyl?
62. ¿Dónde está ubicada la calle Buxareo?
63. ¿Dónde está ubicada la ciudad de Bologna?
64. ¿Dónde está ubicado el balneario Atlántida?
65. ¿Dónde fue inventado Genexus?
66. ¿Dónde juega Diego Forlan?
67. ¿Dónde jugaba Pablo Forlan?
68. ¿Dónde jugó Pablo Forlan?
69. ¿Dónde nació Carlos Gardel?
70. ¿Dónde nació Gardel?
71. ¿Dónde nació Jose Gervasio Artigas?
72. ¿Dónde nació Artigas?
73. ¿Dónde se encuentran las islas Malvinas?
74. ¿Dónde se invento la pólvora?
75. ¿Dónde está el museo del Prado?
76. ¿Dónde se jugaron las primeras olimpiadas?
77. ¿Dónde se jugó el mundial de fútbol de 1962?
78. ¿Dónde se lanzó la primera bomba atómica?
79. ¿Dónde tiene las oficinas centrales el FBI?
80. ¿Dónde vive Batman?
81. ¿En qué año cerró la empresa de transporte Onda?
82. ¿En qué año se fundó Coca Cola?
83. ¿En qué año se fundó el Frente Amplio?
84. ¿En qué año se inventó el televisor Color?
85. ¿En qué año se produjo la guerra de la triple alianza?
86. ¿En qué ciudad se encuentra el cristo redentor?
87. ¿En qué ciudad se encuentra el famoso cristo redentor?

88. ¿En qué ciudad se jugaron los juegos olímpicos de 1992?
89. ¿En qué ciudad se jugaron los juegos olímpicos de 1996?
90. ¿En qué consiste el teorema de Pitágoras?
91. ¿En qué continente se encuentra Madagascar?
92. ¿En qué deporte se destacó Nadia Comaneci?
93. ¿En qué fecha cayó la URRS?
94. ¿En qué fecha se desarrolló la primera olimpiada de la era moderna?
95. ¿En qué fecha se disolvió la URSS?
96. ¿En qué fecha se produjo el desembarco de los 33 Orientales?
97. ¿En qué país nació Carol Lewis?
98. ¿En qué país no nació Carlos Gardel?
99. ¿En qué país se inventó la pólvora?
100. ¿En qué país se jugaron los juegos olímpicos de 1992?
101. ¿En qué país se jugaron los juegos olímpicos de 1996?
102. ¿Quién asesinó a John Lennon?
103. ¿Quién mató a John Lennon?
104. ¿Quién descubrió la isla de pascua?
105. ¿Quién dijo la frase "ni vencedores ni vencidos"?
106. ¿Quién era el hermano de Abel?
107. ¿Quién era el hermano de Caín?
108. ¿Quién era el presidente de la URSS cuando se separó?
109. ¿Quién era la esposa de Napoleón?
110. ¿Quién es Michael Jackson?
111. ¿Quién es Superman?
112. ¿Quién es el Presidente de Artech?
113. ¿Quién es el Vice Presidente de Artech?
114. ¿Quién es el Vice-Presidente de Artech?

115. ¿Quién es el Vicepresidente de Artech?
116. ¿Quién es el actual presidente de Estados Unidos?
117. ¿Quién es el autor de Moby Dick?
118. ¿Quién es el conejo de la suerte?
119. ¿Quién es el creador de Genexus?
120. ¿Quién es el creador de los Simpsons?
121. ¿Quién es el hombre más alto del mundo?
122. ¿Quién es el presidente de Microsoft?
123. ¿Quién escribió de la tierra a la luna?
124. ¿Quién escribió el libro Don Quijote de la Mancha?
125. ¿Quién escribió el primer libro de Genexus?
126. ¿Quién escribió rayuela?
127. ¿Quién fue el primer Virrey del río de la Plata?
128. ¿Quién fue el primer hombre en pisar la luna?
129. ¿Quién fue el primer presidente de Argentina?
130. ¿Quién fue el primer presidente de Estados Unidos?
131. ¿Quién fue el primer presidente de Uruguay?
132. ¿Quién ganó el mundial de fútbol de 1994?
133. ¿Quién es el presidente de Rusia?
134. ¿Quién ganó la copa del mundo de fútbol de 1994?
135. ¿Quién introdujo el ganado al río de la plata?
136. ¿Quién inventó el motor eléctrico?
137. ¿Quién inventó la bomba atómica?
138. ¿Quién se enfrentó en la segunda guerra mundial?
139. ¿Quién tiró las torres gemelas?
140. ¿Qué descubrió Cristóbal Colon?
141. ¿Qué edad tenía Artigas cuando murió?

142. ¿Qué es la celulosa?
143. ¿Qué ingredientes tiene el Cuba Libre?
144. ¿Qué libros escribió Cortazar?
145. ¿Qué nacionalidad tenía Pablo Neruda?
146. ¿Qué ocurrió en el año 1929?
147. ¿Qué población tiene Francia?
148. ¿Qué profundidad tiene el Océano Pacífico?
149. ¿Qué profundidad tiene el río Cuareim?
150. ¿Qué superficie tiene Uruguay?

A.4. Conjunto de Prueba de 200 preguntas CLEF 2003

1. ¿A partir de qué sustancia se obtiene el tolueno?
2. ¿A qué primer ministro abrió la Fiscalía de Milán un sumario por corrupción?
3. ¿A qué compañía petrolera pertenece Brent Spar?
4. ¿A qué edad murió Joseph di Mambro?
5. ¿A qué edad murió Thomas Tip O'Neill?
6. ¿A qué grupo pertenecía John Lennon?
7. ¿A qué marca pertenecían los alimentos para bebés en los que se encontraron pesticidas?
8. ¿A qué país pertenece el agente inmobiliario Schneider?
9. ¿A qué país se dirigían las ayudas del programa Turquesa?
10. ¿Cuál es el nombre de pila de Milosevic?
11. ¿Cuál es el nombre de pila de la mujer de Nelson Mandela?
12. ¿Cuál es el nombre técnico del mal de las vacas locas?
13. ¿Cuál es el principal país productor de petróleo en el mundo?

14. ¿Cuál es la capital de Canadá?
15. ¿Cuál es la capital de Corea del Norte?
16. ¿Cuál es la capital de Croacia?
17. ¿Cuál es la capital de Haití?
18. ¿Cuál es la capital de Irlanda?
19. ¿Cuál es la capital de Irán?
20. ¿Cuál es la capital de Malasia?
21. ¿Cuál es la capital de Rusia?
22. ¿Cuál es la capital de Turkmenistán?
23. ¿Cuál es la capital de la República de Sudáfrica?
24. ¿Cuál es la causa más frecuente de los accidentes de coche?
25. ¿Cuál es la distancia entre la Tierra y el Sol?
26. ¿Cuál es la población mundial?
27. ¿Cuál es la profesión de Renzo Piano?
28. ¿Cuál fue el resultado de la final de la Copa de Europa de Clubs de Baloncesto?
29. ¿Cuál fue el resultado del partido Italia-Noruega del mundial de fútbol?
30. ¿Cuáles son las siglas del Fondo Mundial para la Protección de la Naturaleza?
31. ¿Cuándo comenzó el embargo sobre Irak?
32. ¿Cuándo decidió Naciones Unidas imponer el embargo sobre Irak?
33. ¿Cuándo explotó la primera bomba atómica?
34. ¿Cuándo ocurrió la catástrofe de Chernobil?
35. ¿Cuándo se constituyó la República de Sudáfrica?
36. ¿Cuándo se creó el GATT?
37. ¿Cuándo se firmó el Acta Única Europea?
38. ¿Cuándo se firmó el Tratado de Maastricht?

39. ¿Cuándo se firmó el Tratado de Roma?
40. ¿Cuándo se fundó la CEE?
41. ¿Cuándo se produjo la reunificación de Alemania?
42. ¿Cuándo tomó China la posesión de Hong Kong?
43. ¿Cuántas personas fueron rescatadas por los equipos de socorro tras el naufragio del ferry Estonia?
44. ¿Cuántas personas murieron ahogadas al zozobrar y hundirse el Estonia?
45. ¿Cuántas personas murieron en el accidente de un Airbus en el aeropuerto de Nagoya?
46. ¿Cuántas personas murieron en el juzgado de Euskirchen?
47. ¿Cuánto mide el Everest?
48. ¿Cuánto tiempo ha estado en el poder Kim Il Sung en Corea del Norte?
49. ¿Cuánto valen 10 pesos?
50. ¿Cuántos campeonatos del mundo de Fórmula 1 ganó el piloto brasileño Ayrton Senna?
51. ¿Cuántos cantones hay en Suiza?
52. ¿Cuántos habitantes hay en Irak?
53. ¿Cuántos habitantes hay en Moscú?
54. ¿Cuántos habitantes tiene Berlín?
55. ¿Cuántos habitantes tiene Chechenia?
56. ¿Cuántos habitantes tiene Corea del Norte?
57. ¿Cuántos habitantes tiene Rusia?
58. ¿Cuántos habitantes tiene Sidney?
59. ¿Cuántos habitantes tiene Suecia?
60. ¿Cuántos hijos tiene Anthony Quinn?
61. ¿Cuántos motores tiene un avión?
62. ¿Cuántos muertos al año causan las minas antipersonas en el mundo?

63. ¿Cuántos objetos de arte son robados en Europa cada año?
64. ¿Cuántos partidos políticos participaron en las primeras elecciones locales de la historia en Sudáfrica?
65. ¿Cuántos pasajeros murieron en el naufragio del ferry Estonia?
66. ¿Cuántos pasajeros tuvieron que abandonar el Regent Star tras incendiarse el barco?
67. ¿Cuántos países miembros hay en las Naciones Unidas?
68. ¿Cuántos países participaron en la Conferencia Mundial de Población?
69. ¿Cuántos países son miembros de la Unión Europea?
70. ¿Cuántos tripulantes murieron en el submarino Emeraude?
71. ¿Cómo murió Ayrton Senna?
72. ¿Cómo se llama el hijo de Kim Il Sung?
73. ¿Cómo se llama el jefe de gobierno de Australia?
74. ¿Cómo se llama el primer ministro holandés?
75. ¿Cómo se llama el refresco de cola de Richard Branson?
76. ¿Cómo se llama el servicio de seguridad nacional de Israel?
77. ¿Cómo se llama el sucesor del GATT?
78. ¿Cómo se llama la compañía aérea nacional de Suiza?
79. ¿Cómo se llama la moneda china?
80. ¿Cómo se llamaba el cantante y líder de Nirvana?
81. ¿Cómo se llaman las líneas aéreas de Nikki Lauda?
82. ¿De cuántas muertes son responsables los Jemeres Rojos?
83. ¿De qué club de fútbol es presidente Jesús Gil?
84. ¿De qué país es presidente Yeltsin?
85. ¿De qué puerto partió el ferry Estonia?
86. ¿Dónde está Basora?
87. ¿Dónde está Chiapas?
88. ¿Dónde está Euskirchen?

89. ¿Dónde está Haití?
90. ¿Dónde está el Muro de las Lamentaciones?
91. ¿Dónde está el volcán Popocatepetl?
92. ¿Dónde explotó la primera bomba atómica?
93. ¿Dónde nació Adolfo Hitler?
94. ¿Dónde se celebraron los Juegos Olímpicos de 1996?
95. ¿Dónde se celebró la asamblea anual de la Comisión Ballenera Internacional?
96. ¿Dónde se celebró la cumbre del G7?
97. ¿Dónde se hundió el Estonia?
98. ¿En qué año cayó el muro de Berlín?
99. ¿En qué año entró España en la Comunidad Europea?
100. ¿En qué año fueron prohibidas las pruebas de armas biológicas y tóxicas?
101. ¿En qué año se creó el Fondo Monetario Internacional?
102. ¿En qué año se creó el Banco Mundial?
103. ¿En qué ciudad está Broadway?
104. ¿En qué ciudad está el Museo del Prado?
105. ¿En qué ciudad está el parlamento europeo?
106. ¿En qué ciudad está la puerta de Brandeburgo?
107. ¿En qué ciudad europea está la Torre Eiffel?
108. ¿En qué ciudad se celebraron los Juegos Olímpicos de invierno?
109. ¿En qué ciudad se celebró la Conferencia Mundial de Población?
110. ¿En qué ciudad se celebró la final del mundial de fútbol?
111. ¿En qué conferencia se crearon el BM y el FMI?
112. ¿En qué estado de Estados Unidos está San Francisco?
113. ¿En qué festival se entregan los premios León de Oro?
114. ¿En qué mes se produjo el naufragio del Estonia?

115. ¿En qué océano se hundió el Titanic?
116. ¿En qué parte de Rusia se rompió un oleoducto?
117. ¿En qué país está la zona de los Grandes Lagos?
118. ¿En qué país se encuentra la región de Bosnia?
119. ¿En qué tipo de procesador se descubrió un error en la unidad aritmética?
120. ¿Por qué teoría se ha concedido el Premio Nóbel de Economía?
121. ¿Quién construyó el muro de Berlín?
122. ¿Quién dirigió Con la muerte en los talones?
123. ¿Quién era conocido como el Zorro del Desierto?
124. ¿Quién es el creador de Doctor Snuggles?
125. ¿Quién es el director de la CIA?
126. ¿Quién es el entrenador del equipo nacional de fútbol noruego?
127. ¿Quién es el fundador de la Orden del Templo del Sol?
128. ¿Quién es el líder bosnio?
129. ¿Quién es el líder de Nación del Islam?
130. ¿Quién es el líder de los serbios de Bosnia?
131. ¿Quién es el líder del Sinn Fein?
132. ¿Quién es el líder del grupo guerrillero UNITA de Angola?
133. ¿Quién es el mayor exportador europeo de aceite de oliva?
134. ¿Quién es el ministro de economía alemán?
135. ¿Quién es el ministro italiano de Asuntos Exteriores?
136. ¿Quién es el presidente de Alemania?
137. ¿Quién es el presidente de Corea del Norte?
138. ¿Quién es el presidente de Corea del Sur?
139. ¿Quién es el presidente de Estados Unidos?
140. ¿Quién es el presidente de FIAT?

141. ¿Quién es el presidente de Perú?
142. ¿Quién es el presidente de Rusia?
143. ¿Quién es el presidente de Yugoslavia?
144. ¿Quién es el presidente de la Autoridad Nacional Palestina?
145. ¿Quién es el presidente de la Comisión Europea?
146. ¿Quién es el presidente de la República de Italia?
147. ¿Quién es el presidente de la república francesa?
148. ¿Quién es el presidente del Parlamento Europeo?
149. ¿Quién es el primer ministro húngaro?
150. ¿Quién escribió Star Trek?
151. ¿Quién fue el primer presidente de Indonesia?
152. ¿Quién fue la ganadora del torneo de Wimbledon?
153. ¿Quién ganó el Tour de Francia?
154. ¿Quién mató a Andrés Escobar, un jugador de fútbol colombiano?
155. ¿Quién ostenta el poder en Pyongyang?
156. ¿Quién proyectó la construcción de la catedral de San Pedro?
157. ¿Qué cargo ostentaba Rabbani al estallar la guerra civil de Afganistán en 1992?
158. ¿Qué causó el incendio en un cine en la ciudad china de Karamai?
159. ¿Qué ciudadano británico recibió 50 latigazos en Qatar?
160. ¿Qué día comenzó la intifada?
161. ¿Qué día entró en vigor el Tratado de Maastricht?
162. ¿Qué empresa británica pertenece al consorcio Airbus?
163. ¿Qué empresa ha comprado a la fabricante de coches Rover?
164. ¿Qué equipo ganó el torneo de la NBA?
165. ¿Qué equipo ganó la Copa de Europa de Clubs de Baloncesto?
166. ¿Qué espectáculo es considerado el más grande del mundo?

167. ¿Qué ex ministro francés fue encarcelado por corrupción?
168. ¿Qué ferry se hundió en el Sudeste de la isla Utoe?
169. ¿Qué magnitud tuvo el terremoto que sacudió el norte de Japón?
170. ¿Qué organismo impuso el embargo sobre Irak?
171. ¿Qué país de África ha adoptado una nueva constitución?
172. ¿Qué país europeo es el mayor consumidor de alcohol?
173. ¿Qué país ganó la Copa Davis?
174. ¿Qué país invadió Kuwait en 1990?
175. ¿Qué porcentaje del comercio mundial de drogas está controlado por el Cartel de Cali?
176. ¿Qué premio Nóbel consiguió Kenzaburo Oe?
177. ¿Qué premio Nóbel fue concedido a Willy Brandt?
178. ¿Qué premio Nóbel ganó Solzhenitsin?
179. ¿Qué premio fue concedido a Weinberg, Salam y Glashow?
180. ¿Qué premio ganó la película Pulp Fiction, dirigida por Quentin Tarantino, en el Festival de Cine de Cannes?
181. ¿Qué presidente de Corea del Norte murió a los 82 años de edad?
182. ¿Qué presidente ruso asistió a la reunión del G7 en Nápoles?
183. ¿Qué presidente ruso ordenó la intervención en Chechenia?
184. ¿Qué primer ministro británico visitó Sudáfrica en 1960?
185. ¿Qué primer ministro francés se suicidó en los años 90?
186. ¿Qué significa OLP?
187. ¿Qué significa el acrónimo ONU?
188. ¿Qué significan las siglas CEE?
189. ¿Qué significan las siglas ETA?
190. ¿Qué significan las siglas GATT?
191. ¿Qué significan las siglas IRA?
192. ¿Qué significan las siglas OMC?

193. ¿Qué submarino nuclear francés sufrió un accidente?
194. ¿Qué terrorista de ETA es conocida como La Tigresa?
195. ¿Sobre qué continente se detectó el agujero de ozono?
196. Dar el nombre de alguna película de Spike Lee.
197. Dar el nombre de algún tratamiento contra el SIDA.
198. Dar el nombre de un medicamento contra la malaria.
199. Dar el nombre de una ciudad japonesa que haya sido castigada por un terremoto.
200. Dar el nombre de una película protagonizada por Audrey Hepburn.