UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA
Departamento de Lenguajes y Sistemas Informáticos
Escuela Técnica Superior de Ingeniería Informática

EVALUACIÓN DE SISTEMAS DE BÚSQUEDA Y VALIDACIÓN DE RESPUESTAS

TESIS DOCTORAL

Alvaro Rodrigo Yuste
Ingeniero en Informática
2010

UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA
Departamento de Lenguajes y Sistemas Informáticos
Escuela Técnica Superior de Ingeniería Informática

EVALUACIÓN DE SISTEMAS DE BÚSQUEDA Y VALIDACIÓN DE RESPUESTAS

Alvaro Rodrigo Yuste
Ingeniero en Informática por la Universidad Complutense de Madrid

Director:

Anselmo Peñas Padilla

Profesor Titular de Universidad del Departamento de Lenguajes y Sistemas Informáticos
de la Universidad Nacional de Educación a Distancia

a mis padres y a mi hermano

Agradecimientos

Agradecimientos institucionales

El presente trabajo ha sido financiado por el Ministerio de Ciencia y Tecnología, con cargo a los presupuestos del proyecto SyEMBRA (Sistemas y Evaluación Multilingües de Búsqueda de Respuestas) TIC-2003-07158-C04-02, el proyecto QEAVis-Catiex (TIN2007-67581-C02-01) del Ministerio de Ciencia e Innovación, el programa de becas predoctorales UNED, la Comunidad de Madrid bajo la Red de Investigación MAVIR (S-0505/TIC-0267), la Consejería de Educación de la Comunidad de Madrid y el Fondo Social Europeo (F.S.E.).

Agradecimientos personales

Resumen

En esta tesis se propone un marco para la evaluación de módulos de Validación de Respuestas (AV) que tienen el propósito de mejorar los resultados de los sistemas de Búsqueda de Respuestas (QA). La motivación para la definición de este marco surge del análisis de los resultados de las evaluaciones de QA, donde se observan las siguientes situaciones en las cuáles se podrían mejorar los resultados mediante la incorporación de módulos de AV:

- Los conjuntos de respuestas devueltas contienen respuestas incorrectas que provocan que los resultados empeoren. El hecho de eliminar el mayor número de respuestas incorrectas de un conjunto de candidatas supondría una mejora de los resultados.
- Los distintos sistemas de QA se complementan entre si de modo que, aunque individualmente obtienen resultados similares, la combinación efectiva de los mismos da lugar a resultados mejores que los de cualquiera de los sistemas individuales.
- El procesamiento en cadena, típico de las arquitecturas clásicas utilizadas en QA, provoca que haya una alta dependencia entre módulos y los errores se propaguen de unos módulos a otros. La posibilidad de romper este procesamiento en cadena permitiría disminuir la dependencia entre módulos, permitiendo mejorar los resultados.

El primer paso para la definición del marco de evaluación consiste en la propuesta de un modelo de AV basado en el Reconocimiento de Implicación Textual (RTE). Para comprobar la validez de este modelo se construye una colección de pares texto-hipótesis (que siguen un formato similar al de las colecciones de los RTE Challenges) enfocados a la tarea de AV. El análisis de esta colección permite comprobar la validez del modelo propuesto y supone el punto de partida para la definición del marco de evaluación.

La metodología propuesta permite la evaluación de sistemas de AV que actúan en diversos escenarios dentro de un sistema de QA, y la comparación de sus resultados con otros sistemas de QA, para así comprobar si el uso de estos módulos supone mejoras de rendimiento. Además, como parte de la metodología se describen diversos métodos para construir colecciones de evaluación reutilizando los juicios humanos de las evaluaciones de QA.

El marco definido se puso en práctica dentro de una tarea de evaluación internacional, el Answer Validation Exercise (AVE), que se desarrolló durante tres ediciones dentro del marco del Cross Language Evaluation Forum (CLEF). La experiencia obtenida durante las tres ediciones de la tarea sirvió para refinar la metodología hasta su versión final, la cuál está a disposición de la comunidad científica junto con los recursos de evaluación generados, para la evaluación de futuros sistemas de AV.

Los resultados obtenidos por los sistemas participantes en las campañas del AVE permiten observar que la utilización de módulos de AV mejoraría los resultados en QA, en las tres líneas que se observaron al analizar las evaluaciones de sistemas de QA (eliminar respuestas candidatas incorrectas, combinar sistemas de QA y romper el procesamiento en cadena de un sistema de QA). De hecho, estas observaciones han servido para que haya sistemas de QA que incorporen módulos de AV, logrando mejorar sus resultados. Además, la mayoría de estos sistemas hizo uso del modelo basado en RTE que se presenta en esta tesis, por lo que se ha demostrado su validez y utilidad en entornos reales.

Finalmente, en esta tesis se observa que los módulos de AV podrían ser también de utilidad en escenarios de QA donde es mejor no responder a una pregunta que responderla incorrectamente, como podría suceder por ejemplo en diagnóstico médico. Sin embargo, las evaluaciones de QA no han prestado especial atención a este tipo de escenarios. Por este motivo, en esta tesis se propone una nueva medida para evaluar sistemas de QA que permite premiar a los sistemas que mantienen el número de preguntas respondidas correctamente y logran reducir la cantidad de respuestas incorrectas al dejar preguntas sin responder. Las pruebas realizadas sobre esta medida han mostrado su eficacia a la hora de detectar los mejores enfoques para este tipo de escenarios en comparación con otras medidas de evaluación típicas en QA.

Abstract

This thesis proposes a methodology for evaluating Answer Validation (AV) modules that are focused on improving Question Answering (QA) results. The motivation for the definition of this methodology arises from the analysis of current QA evaluations. This analysis shows the following situations where the performance of QA results could be improved by including AV modules in QA:

- The sets of answers returned by QA systems contain incorrect answers that contribute to worsen results. An improvement in the performance of QA systems could be achieved by removing incorrect answers from the set of the candidate ones.
- Although different QA systems obtain similar results, an effective combination of them sometimes obtains better results than the ones achieved by each individual system.
- The pipe-line approach, which is typical in classical QA architectures, has a high dependency among modules that is highly sensitive to error propagation. The possibility of breaking this pipe-line would allow to reduce the dependency among modules, contributing to an improvement in results.

The proposal of an AV model based of the Recognition of Textual Entailment (RTE) is the first step for the definition of the evaluation methodology. A collection of text-hypothesis pairs (similar to the ones used in the RTE collections) is built in order to check the viability of the proposed model. The analysis of this collection shows that the model is suitable, and this analysis represents also the starting point for the development of the evaluation methodology proposed in this thesis.

The proposed methodology allows to evaluate AV systems with different functions inside a QA system. The methodology allows also the comparison of AV results with the ones of QA systems, what permits checking whether the use of AV modules could contribute to the improvement of QA results. Furthermore, different methods for building evaluation collections are proposed. All these methods reuse human assessments from QA evaluations.

The methodology was tested in an international evaluation task, the Answer Validation Exercise (AVE), celebrated during three years at the Cross Language Evaluation Forum (CLEF). The experience obtained in the three editions of the

task was used to improve the methodology to its final version, which is available to the scientific community and can be used for evaluating new AV systems.

The results obtained by participants at AVE show how the use of AV modules would improve QA performance in the three lines detected in the analysis of QA evaluations (the removal of incorrect candidate answers, the combination of QA systems and the break of pipe-line processing used in QA architectures). These observations have motivated the inclusion of AV modules into real QA systems, what leads to the improvement of results in these QA systems. Besides, most of these AV modules employed the model based on RTE proposed in this thesis, showing its usefullness in real scenarios.

Finally, this thesis suggests that AV modules could be useful in QA scenarios where is better not to answer at all than to answer incorrectly, as for example in medical diagnosis. However, the evaluations of QA systems have not paid attention to this kind of scenarios. This is why we define a new measure for evaluating QA systems in these scenarios. This measure rewards QA systems which keep the number of correct answers, while reducing the amount of incorrect ones by leaving some questions unanswered. The measure has shown its adequacy for detecting the best approaches in these scenarios.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

Un sistema de Búsqueda de Respuestas[1] (en inglés Question Answering, QA) es capaz de proporcionar una respuesta precisa a una pregunta formulada en lenguaje natural. La principal diferencia entre los sistemas de QA y los sistemas de Recuperación de Información (en inglés Information Retrieval, IR), cuyos ejemplos más conocidos son buscadores de internet como Google[2] o Yahoo![3], es que estos últimos proporcionan páginas web o documentos en los que se espera que aparezca la información que desea el usuario, la cuál ha de ser buscada por el propio usuario. En cambio, un sistema de QA devuelve una respuesta precisa, es decir, no solo localiza el documento con la respuesta, sino que también extrae de ese documento la respuesta exacta y se la presenta al usuario. Además, en los sistemas de IR la consulta se realiza mediante el uso de palabras clave, mientras que un sistema de QA recibe una pregunta formulada en lenguaje natural, lo cuál es más cómodo para el usuario.

Los resultados de los diversos foros de evaluación de sistemas de QA muestran que a pesar de que las preguntas formuladas no son de gran complejidad, los mejores sistemas no son capaces de contestar correctamente más del 80 % de las preguntas propuestas. Además, en algunas ediciones en las cuáles se aumentó la complejidad de las preguntas, los mejores sistemas no lograron responder correctamente más del 60 % de las preguntas planteadas. De hecho, este 60 % de preguntas respondidas correctamente representa el rendimiento medio alcanzado por la mayoría de los sistemas.

Surge entonces la pregunta de por qué no se logra superar en la mayoría de los

[1] En español también se suele usar el término Pregunta-Respuesta para designar a este tipo de sistemas, ya que se puede criticar que el término Búsqueda de Respuestas hace referencia a un sistema que tiene una base de datos con una serie de preguntas y respuestas donde se busca una respuesta almacenada para una determinada pregunta sin realizar ningún tipo de procesamiento. Sin embargo, en este trabajo el autor ha preferido hacer uso del término Búsqueda de Respuestas con el sentido de que se busca la respuesta en documentos, bibliotecas digitales, etc, sin que ésta esté previamente asociada a una determinada pregunta.

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

[3] www.yahoo.com/

casos esta barrera del 60 %. Además, sería también interesante conocer cómo se podría superar esta barrera y mejorar los resultados actuales.

En los siguientes apartados se analizan situaciones que se han observado en las evaluaciones de QA en las que se podría trabajar para mejorar los resultados de este tipo de sistemas. A continuación se enumeran los objetivos y la metodología seguida en esta tesis. Finalmente se muestra la estructura de la tesis.

1.1. Evolución de los Sistemas

El análisis de los resultados de las evaluaciones de QA permite observar que los sistemas de QA cometen errores al crear el ranking de respuestas candidatas para cada pregunta, lo cual da lugar a situaciones en las cuales la primera respuesta del ranking no es correcta pero sí lo es la segunda o la tercera respuesta. Este dato es un indicador de que el ranking de respuestas que se realiza se podría mejorar, lo que conllevaría una mejora en los resultados actuales en QA, permitiendo superar los resultados actuales. Una forma de alcanzar esta mejora podría llevarse a cabo mediante la utilización de módulos de Validación de Respuestas (en inglés Answer Validation, AV) que eliminasen de entre las respuestas candidatas aquellas para las cuáles no hay suficientes evidencias acerca de su corrección.

Un sistema de AV recibe una *Pregunta* y una *Respuesta* y devuelve un valor indicando si la *Respuesta* es o no correcta y en qué grado. Este tipo de sistemas ha sido utilizado para filtrar respuestas candidatas erróneas con el fin de mejorar los resultados de sistemas de QA (Magnini et al., 2002b). Sin embargo, la mayoría de las propuestas de sistemas de AV realizadas antes de esta tesis no se han preocupado de realizar un análisis profundo de las relaciones semánticas que deben de existir entre una pregunta y una respuesta para que ésta sea considerada correcta, lo cuál permitiría mejorar el rendimiento de este tipo de sistemas. Además, la evaluación de sistemas de AV no ha recibido especial interés por parte de la comunidad científica, lo cuál ha tenido como consecuencia que no se haya prestado especial interés al desarrollo de este tipo de sistemas.

Para solucionar este problema, en esta tesis se propone una metodología para evaluar sistemas de AV donde además de realizarse una evaluación del rendimiento de estos sistemas, se pone énfasis en evaluar también el impacto que supondría su incorporación en QA. La definición de esta metodología parte de un modelo de sistemas de AV, propuesto en esta tesis, que promueve un mayor análisis de las respuestas. De este modo, mediante la disponibilidad de esta metodología se fomenta la incorporación de módulos de AV que mejoren los resultados de los sistemas actuales de QA.

1.2. Especialización y Colaboración

Un dato significativo sobre el comportamiento de los sistemas de QA en las evaluaciones se puede ver en la Figura 1.1 (página 27), donde se muestra el número

de respuestas correctas devueltas por los sistemas participantes en la tarea de QA del CLEF 2006 en español (Magnini et al., 2007), así como la tipología de las preguntas formuladas. Además, en la Figura se puede ver también el rendimiento de la mejor combinación posible de los sistemas representados.

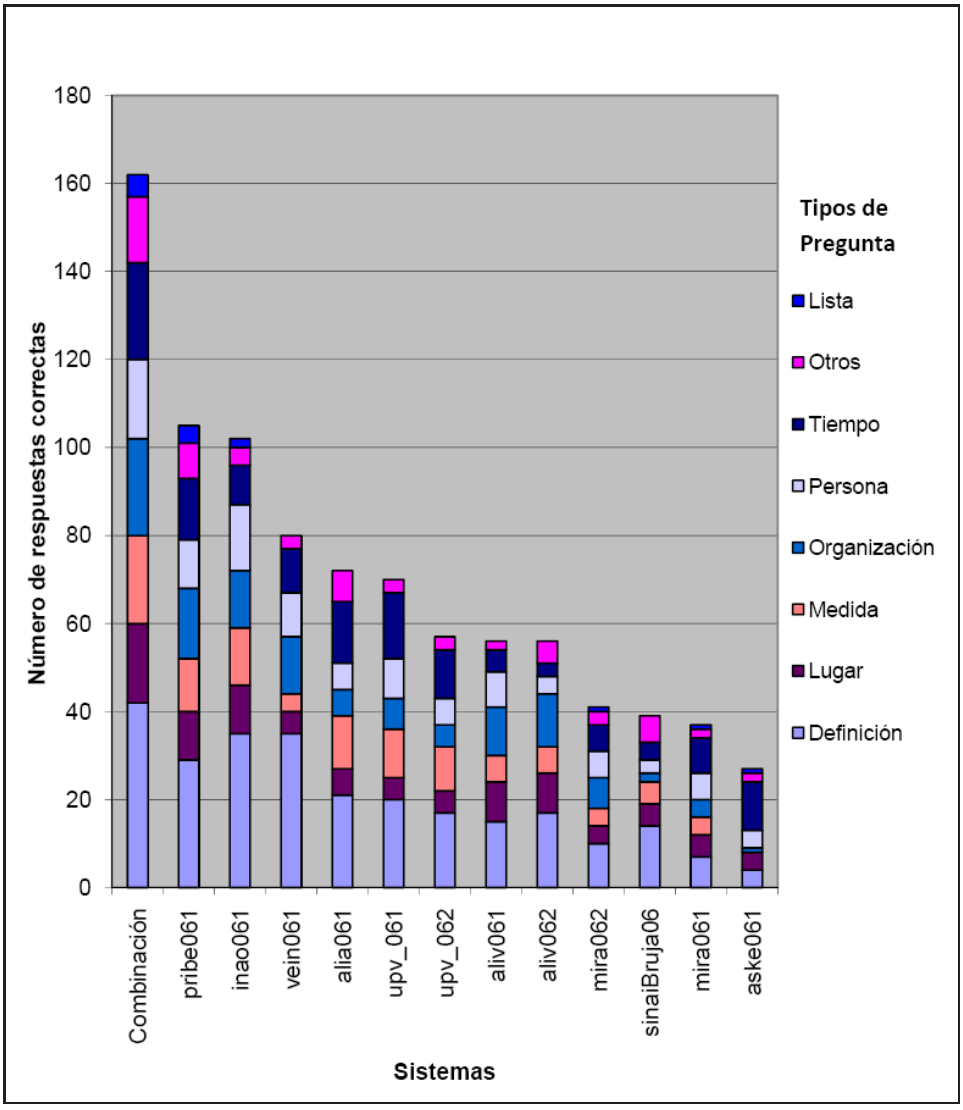

Figura 1.1: Distribución de respuestas correctas por sistema participante en la evaluación de sistemas de QA en español en el CLEF 2006.

Como puede observarse en la Figura 1.1, aunque el 81 % de las preguntas fueron respondidas correctamente por algún sistema (los sistemas participantes tenían que responder a 200 preguntas), el mejor de todos ellos (el sistema *pribe061*) tan sólo fue capaz de responder correctamente al 52.5 % de las preguntas. También se puede ver que el mejor sistema no tiene por qué ser el que mejor responde a los dis-

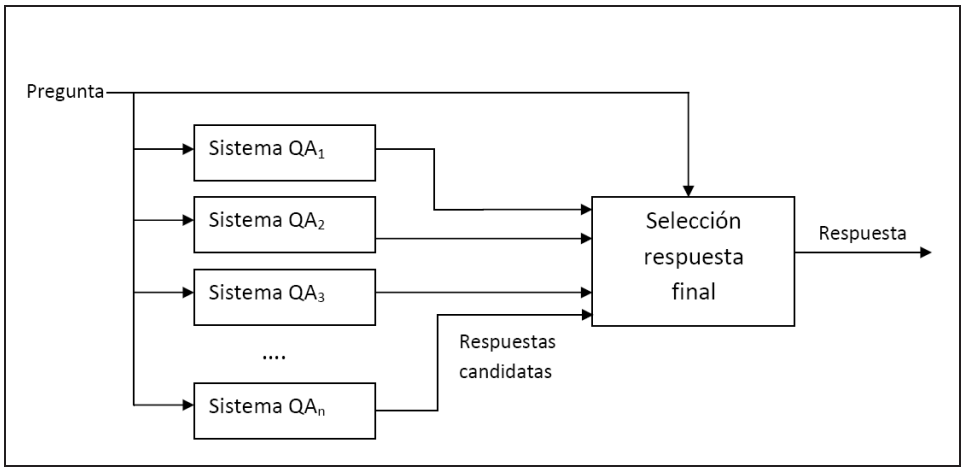

Figura 1.2: Arquitectura de un sistema multi-flujo de QA

tintos tipos de preguntas. De hecho, el quinto mejor sistema (el sistema *upv_061*) es el que mejores resultados obtuvo en las preguntas de tipo temporal.

Por tanto, el gráfico de la Figura 1.1 ilustra cómo los distintos sistemas de QA se complementan entre si, de modo que una combinación de los mismos podría mejorar los resultados alcanzados por un solo sistema. En concreto, en la Figura se observa que una combinación perfecta de todos los sistemas mejoraría los resultados del mejor sistema en más de un 50 %, pasando del 52.5 % de respuestas respondidas correctamente por el mejor sistema al 81 %, lo cuál es una mejora significativa. Esto sugiere que una de las posibilidades para avanzar en el área consiste en fomentar la especialización de los sistemas en determinados tipos de preguntas para que luego colaboren entre si creando un sistema multi-flujo (en inglés multi-stream) de QA.

Un sistema multi-flujo de QA está compuesto por varios sistemas de QA que reciben las mismas preguntas y generan respuestas a dichas preguntas como se muestra en la Figura 1.2 (página 28). Además, el sistema multi-flujo hace uso de un subsistema que recibe las respuestas generadas por los distintos sistemas de QA y se encarga de seleccionar la respuesta final del sistema multi-flujo.

El reto que surge en este tipo de arquitecturas es el desarrollo de buenos criterios para realizar la selección de las respuestas candidatas provenientes de los diversos sistemas individuales. Por tanto se incide de nuevo en el problema de la validación, ya que hay que decidir cuál de las respuestas candidatas es la que más probabilidades tiene de ser correcta, para proponerla como respuesta final del sistema multi-flujo y mejorar los resultados globales.

La metodología de evaluación propuesta en este trabajo pretende promover este cambio y favorecer el progreso en el área. Para ello, esta metodología permite evaluar módulos de AV que seleccionan la respuesta de un sistema multi-flujo, permitiendo comprobar qué enfoques son los más prometedores.

1.3. El Problema de la Arquitectura en Cascada

Otro de los motivos por los cuáles no se produce una mejora en los resultados de los sistemas de QA se debe a la arquitectura clásica que se utiliza en este tipo de sistemas. Un sistema de QA desarrolla un complejo procesamiento donde la principales etapas son:

1. Análisis de la pregunta
2. Recuperación de pasajes
3. Extracción de la respuesta
4. Ranking de respuestas

Sin embargo, uno de los problemas asociados a este tipo de arquitectura en cascada es la propagación de errores. De este modo, por ejemplo, aunque los módulos de recuperación de pasajes y de extracción de la respuesta alcanzasen por separado una precisión del 80 %, como consecuencia de la dependencia entre ambos módulos se tendría un límite superior del 64 % de precisión a la hora de encontrar respuestas correctas. Además, no es seguro que el hecho de mejorar un determinado módulo provoque la mejora del rendimiento global del sistema (Sonntag, 2004).

Un modo de romper las limitaciones de este procesamiento en cadena consistiría en permitir que cada etapa conociese si se ha encontrado o no una respuesta correcta a una determinada pregunta. De este modo, cada módulo podría cambiar su comportamiento no sólo en función de la información que recibe de la etapa anterior, sino también teniendo en cuenta la corrección de las respuestas generadas anteriormente. En la Figura 1.3 (página 30) se muestra una arquitectura de este tipo, donde en caso de no encontrarse evidencias sobre la corrección de las respuestas candidatas, se hace llegar esta información a las etapas anteriores.

En una arquitectura como la planteada en la Figura 1.3, un módulo de AV podría ser el encargado de decidir si hay o no alguna respuesta correcta entre las candidatas. En caso de que el módulo de AV no encontrase ninguna respuesta correcta, lo haría saber a las etapas anteriores.

Mediante la propuesta del modelo de sistemas de AV realizada en este trabajo se pretende fomentar la mejora de esta tecnología para que pueda ser utilizada, entre otras cosas, para romper el procesamiento en cadena típico en QA. Además, la evaluación planteada en la metodología propuesta en esta tesis permite estimar los resultados que obtendría un sistema de QA que busca nuevas respuestas cuando un módulo de AV se las solicita.

1.4. Objetivos y Metodología

Este trabajo se centra en promover y evaluar sistemas de Validación de Respuestas que mejoren el rendimiento de los sistemas actuales de Búsquedas de Respuestas, haciendo especial hincapié en:

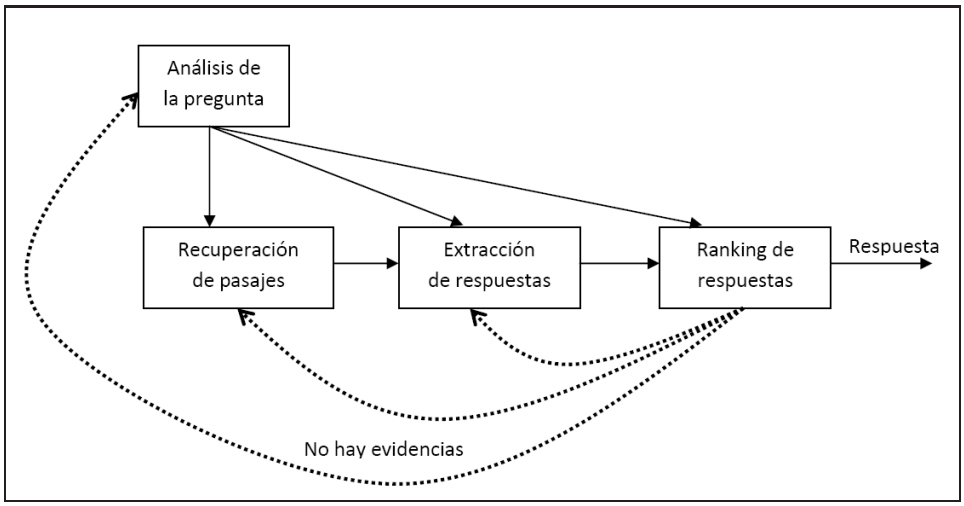

Figura 1.3: Arquitectura de un sistema de Búsqueda de Respuestas que en caso de no encontrar evidencias acerca de la corrección de las respuestas candidatas, se lo comunica a todos sus módulos.

- Arquitecturas de sistemas de Búsquedas de Respuestas que validen las respuestas que generan con el fin de devolver una mayor cantidad de respuestas correctas.
- Arquitecturas de sistemas de Búsquedas de Respuestas multi-flujo cuyo criterio de selección de la respuesta final se base más en análisis textual que en el conteo de redundancias.
- Sistemas de Búsquedas de Respuestas en los que sea preferible no responder a una pregunta a dar una respuesta errónea.
- El uso de técnicas de aprendizaje automático para la validación.
- El desarrollo de sistemas en diversos idiomas.

Para ello, se llevan a cabo las siguientes acciones:

1. Realizar una propuesta metodológica para la evaluación de sistemas de Validación de Respuestas, donde las respuestas son devueltas por sistemas reales de Búsquedas de Respuestas y la validación se fundamenta en el Reconocimiento de la Implicación Textual.
2. Utilizar dicha metodología para llevar a cabo una nueva tarea de evaluación internacional donde los sistemas se puedan comparar entre si.
3. Crear los recursos necesarios para llevar a cabo la tarea de evaluación y permitir el desarrollo futuro de sistemas.

4. Evaluar las medidas de evaluación utilizadas en la metodología propuesta con el propósito de estudiar la confianza que se puede depositar en los resultados obtenidos con dichas medidas.

5. Plantear líneas de trabajo futuro consecuentes con los resultados obtenidos.

1.5. Estructura del Trabajo

En este apartado se describe la estructura que tiene esta tesis, resumiendo el contenido de cada capítulo de la misma.

En el Capítulo 2 (página 33) se exponen los principales trabajos relacionados con el ámbito de esta tesis. En concreto, en este capítulo se estudian las evaluaciones realizadas en distintas tareas de Procesamiento del Lenguaje Natural como la Recuperación de Información, la Extracción de Información, la Búsqueda de Respuestas, etc, definiendo primero el propósito y ámbito de cada tarea para posteriormente realizar un estudio sobre los métodos de evaluación más comúnmente empleados. Además, para estas tareas se detallan las principales características de los sistemas propuestos debido a que su estudio es de relevancia para el ámbito de esta tesis.

En el Capítulo 3 (página 89) se propone un modelo basado en el Reconocimiento de la Implicación Textual para realizar la tarea de Validación de Respuestas. Este modelo promueve el uso de técnicas de análisis más complejas que las utilizadas anteriormente. Para comprobar la viabilidad del mismo se construye una colección que permite estudiar el enfoque propuesto, de modo que el análisis de dicha colección muestra la validez de la propuesta realizada.

A continuación, el Capítulo 4 (página 105) se centra en el estudio de una serie de medidas para realizar la evaluación de sistemas de Validación de Respuestas. Estas medidas se dividen en dos grupos en función de si se centran en la evaluación de sistemas de AV que realizan validación, o en la evaluación de sistemas de AV que realizan selección de respuestas. Este capítulo muestra también un estudio sobre la confianza que ofrecen los resultados obtenidos utilizando algunas de las medidas propuestas en comparación con otras medidas utilizadas para la evaluación en Aprendizaje Automático.

El Capítulo 5 (página 125) propone una metodología para evaluar sistemas de Validación de Respuestas partiendo del modelo propuesto en el Capítulo 3. Esta metodología se puso en práctica como una tarea de evaluación internacional, denominada Answer Validation Exercise (AVE), que permitió refinar la metodología hasta su versión final. Además, en este capítulo se describen también los objetivos que se plantearon a la hora de proponer la metodología de evaluación y el método seguido para generar los recursos de evaluación necesarios. Finalmente, en este capítulo se realiza un análisis de los resultados obtenidos en cada edición de la tarea por parte de los distintos participantes, así como las técnicas y métodos empleados por los mismos.

Por otro lado, en el Capítulo 6 (página 161) se propone el uso de módulos de Validación de Respuestas dentro de sistemas de Búsqueda de Respuestas que prefieren no responder a una pregunta antes de responderla incorrectamente, los cuáles podrían ser de utilidad en dominios como el diagnóstico médico. Dado que la evaluación en este tipo de escenarios no ha recibido demasiada atención, en este capítulo se propone una nueva medida para evaluar sistemas de Búsqueda de Respuestas que trabajan en este escenario. La medida propuesta da mayor valor a la decisión de no responder una pregunta que a responderla incorrectamente, demostrando en la práctica su utilidad para detectar los enfoques más prometedores en este tipo de escenarios.

Finalmente, en el Capítulo 7 (página 181) se recogen las principales conclusiones obtenidas a lo largo del trabajo desarrollado en esta tesis, así como el impacto del mismo y las líneas de trabajo futuro.

Capítulo 2

Preliminares

En este capítulo se realiza una introducción a una serie de campos especializados del Procesamiento del Lenguaje Natural que están relacionadas con el trabajo desarrollado en esta tesis. Debido a que esta tesis se centra en la evaluación de sistemas, este capítulo se focaliza en la evaluación de tareas definidas en los campos mostrados. No se pretende ser exhaustivo al mencionar todos los foros y medidas de evaluación existentes, sino destacar los más relacionados con el ámbito de esta tesis.

El capítulo comienza con la descripción de la tarea de Recuperación de Información y la de Extracción de Información para luego pasar al estudio de la Búsqueda de Respuestas, la cuál sintetiza las dos anteriores. A continuación se examina la tarea de Reconocimiento de la Implicación Textual y la de Validación de Respuestas, ya que tienen gran importancia en este trabajo. Finalmente se estudian diversos métodos que han sido empleados en las tareas mencionadas (principalmente en Recuperación de Información) para estudiar la confianza que se puede establecer en los resultados obtenidos en una evaluación.

2.1. Evaluación en Recuperación de Información

La Recuperación de Información (en inglés Information Retrieval, IR) tiene como objetivo encontrar los materiales (normalmente documentos) que satisfacen las necesidades de información expresadas por un usuario. Definida de esta manera podría parecer que la IR es interesante sólo para algunas personas como bibliotecarios u otros profesionales de la búsqueda de información. Sin embargo, actualmente son millones de personas las que hacen uso de los sistemas de IR al utilizar motores de búsqueda Web como Google[1], Yahoo[2] o MSN[3]. De hecho, la IR se está convirtiendo en la forma predominante de acceso a la información, sobrepasando a las tradicionales búsquedas en bases de datos (Manning et al., 2008).

[1] http://www.google.com/
[2] http://www.yahoo.com/
[3] http://www.msn.com/

El factor que más ha contribuido a extender el uso de la IR ha sido el crecimiento en el uso de la Web. La Web se ha convertido en un repositorio universal de conocimiento que permite compartir ideas e información en una escala nunca vista anteriormente. Sin embargo el uso de la Web lleva asociados varios problemas, siendo uno de los más importantes el poder encontrar información relevante, tarea que suele ser tediosa y compleja. Además, este problema se acentúa en el caso de usuarios sin experiencia.

Los trabajos en IR se encargan de la representación, almacenamiento, organización y acceso a los elementos de información (normalmente documentos), con el objetivo de facilitar el acceso de un usuario a la información (Baeza-Yates and Ribeiro-Neto, 1999). Por ejemplo, la investigación en IR se interesa en métodos para ayudar al usuario a filtrar colecciones de documentos o a realizar un procesamiento adicional sobre los documentos recuperados.

Sin embargo, las necesidades de información de un usuario (como por ejemplo *encontrar documentos que contengan información relevante sobre los juicios celebrados en Italia contra las mafias napolitanas*) no se pueden expresar tal cuál usando los interfaces actuales de los motores de búsqueda Web si se quiere obtener un resultado satisfactorio. Para realizar dicha búsqueda el usuario debe de expresar su necesidad de información en forma de una *consulta* para que ésta sea procesada por el motor de IR. El procedimiento normal consiste en crear dicha consulta como un conjunto de palabras clave que resumen la necesidad de información (como por ejemplo *Italia juicios mafias napolitanas*). El objetivo de un sistema de IR es recuperar a partir de esta consulta información que pueda ser útil para satisfacer la necesidad de información del usuario.

Los sistemas de IR se pueden clasificar según la escala a la que actúan de la siguiente manera según Manning et al. (2008):

- **Búsquedas en la Web**. En este caso el sistema tiene que realizar una búsqueda sobre miles de millones de documentos almacenados en millones de ordenadores. En este contexto surgen problemas como el de realizar el trabajo de manera eficiente a tan gran escala o no dejarse engañar por las empresas que manipulan el contenido de las páginas para conseguir mejorar su ranking dentro de los motores de búsqueda.

- **Búsquedas en empresas, instituciones y dominios específicos**. En este caso las búsquedas pueden realizarse, por ejemplo, sobre una colección de documentos internos de una empresa. El procedimiento normal consiste en tener almacenados los documentos en máquinas dedicadas a dichas tareas.

- **IR personal**. Recientemente ha habido sistemas operativos que han incorporado módulos de IR (como por ejemplo la búsqueda instantánea de Windows Vista o Spotlight en Mac OS X). Además, los programas para gestionar el correo electrónico incorporan también dicha tecnología para facilitar la búsqueda sobre los correos almacenados. En estos escenarios hay que ser capaz de manejar los distintos tipos de documentos que se poseen, así como hacer

que el funcionamiento y mantenimiento del sistema de búsqueda no interfiera demasiado en el rendimiento global del sistema.

2.1.1. Características de la Evaluación

La evaluación de los sistemas de IR consiste en juzgar en qué grado los sistemas son capaces de satisfacer las necesidades de información de los usuarios. Por ejemplo, la velocidad a la hora de obtener los resultados o la cantidad de resultados relevantes son factores importantes a evaluar. Sin embargo, las percepciones del usuario no siempre coinciden con las nociones de calidad de los diseñadores del sistema (Manning et al., 2008). Por ejemplo, la satisfacción del usuario puede depender de cómo se muestran u organizan los resultados en un interfaz, aspectos totalmente independientes de la calidad de los resultados mostrados. Es por ello que el tipo de evaluación elegida depende de los objetivos del sistema de IR. De hecho, a la hora de realizar la evaluación existen principalmente dos criterios que pueden ser medidos: eficiencia y eficacia.

Cuando se pretende medir la eficiencia de un sistema de IR, las medidas de evaluación más comúnmente utilizadas son el tiempo de ejecución y el espacio necesario en disco (Baeza-Yates and Ribeiro-Neto, 1999). Cuanto menor sea el tiempo de respuesta y el espacio utilizado, mejor se considera que es el sistema. Desafortunadamente ambos aspectos suelen están relacionados de forma inversa, de modo que si uno disminuye, el otro aumenta.

Por otro lado, al evaluar la eficacia de un sistema hay que tener en cuenta que las necesidades de información del usuario se expresan en forma de consultas y que estas consultas son imprecisas. Este hecho provoca que los documentos recuperados no siempre sean respuestas exactas y tengan que ser ordenados de acuerdo a su importancia respecto a la consulta. De este modo, los sistemas de IR suelen ser evaluados respecto a lo preciso que es el conjunto de respuestas devuelto.

Dados los intereses de esta tesis, la evaluación de sistemas de IR que se explora en este capítulo se centra en el estudio del rendimiento de los sistemas, dejando a un lado la vertiente interactiva centrada en la satisfacción del usuario.

Las evaluaciones de sistemas de IR a nivel de laboratorio comenzaron con los proyectos Cranfield (Cleverdon, 1967), los cuáles introdujeron un paradigma para la evaluación de sistemas de IR que ha sido el método predominante y se ha utilizado en distintos foros de evaluación como el Text REtrieval Conference (TREC)[4], Cross-Language Evaluation Forum (CLEF)[5] y NII-NACSIS Test Collection for IR Systems (NTCIR)[6]. Este tipo de evaluación se realiza utilizando una colección de evaluación que cumpla los siguientes requisitos:

- Constar de una colección de documentos.

[4] http://trec.nist.gov/
[5] http://www.clef-campaign.org/
[6] http://research.nii.ac.jp/ntcir/

- Disponer de un conjunto de necesidades de información expresadas en forma de consultas.
- Tener un conjunto de juicios de relevancia sobre los documentos. Normalmente dichos juicios suelen ser binarios y expresan si el documento es o no relevante dada una determinada consulta. Se considera que el documento es relevante si trata acerca de la necesidad de información expresada, no porque simplemente contenga todos los términos de la consulta.

De este modo, el enfoque tradicional para evaluar a los sistemas de IR trabaja sobre la noción de documentos relevantes y documentos no relevantes. En términos generales se puede decir que una medida de evaluación cuantifica la similitud entre el conjunto de documentos devueltos por un sistema para un determinado conjunto de consultas y el conjunto de documentos relevantes para dichas consultas. Además, la experiencia ha demostrado que para poder obtener unos resultados fiables es necesario realizar la evaluación sobre un conjunto razonablemente grande de consultas y documentos. Esto se debe a que los resultados varían en gran medida según el número de consultas utilizadas y el tamaño de la colección de documentos sobre la cuál se realiza la búsqueda, de modo que al utilizar más consultas se minimiza, por ejemplo, el efecto que puede tener una consulta demasiado sencilla en el resultado final.

El conjunto de medidas comúnmente utilizadas se puede dividir en dos grandes grupos. Esta división se realiza en función de si al evaluar el conjunto de respuestas devuelto se tiene en cuenta o no el orden de relevancia otorgado por el sistema a los documentos devueltos. En los siguientes subapartados se describen los principales métodos que se utilizan para la evaluación de cada uno de estos grupos.

2.1.2. Evaluación de Conjuntos no Ordenados

Las dos medidas más frecuentemente utilizadas cuando no se atiende al ranking de relevancia son:

Precisión
- *precisión*: fracción de documentos recuperados que son relevantes (Fórmula (2.1)).

Cobertura
- *cobertura* (más conocida por su nombre en inglés, *recall*): fracción de documentos relevantes que son recuperados (Fórmula (2.2)).

$$precisión = \frac{\#documentos\ relevantes\ recuperados}{\#documentos\ recuperados} \tag{2.1}$$

$$recall = \frac{\#documentos\ relevantes\ recuperados}{\#documentos\ relevantes} \tag{2.2}$$

También se pueden definir ambas medidas haciendo uso de la tabla de contingencia mostrada en el Cuadro 2.1 (página 37), de modo que la *precisión* se define en la Fórmula (2.3) y la *cobertura* en la Fórmula (2.4).

Cuadro 2.1: Tabla de contingencia en IR.

	Juicio	
	relevante	**no relevante**
recuperado	true positives (tp)	false positives (fp)
no recuperado	false negatives (fn)	true negatives (tn)

$$precisión = \frac{tp}{tp + fp} \tag{2.3}$$

$$cobertura = \frac{tp}{tp + fn} \tag{2.4}$$

Hay que tener en cuenta que un sistema de IR puede ser visto como un clasificador binario, que trabaja con las clases *relevante* y *no relevante*, que se centra en recuperar el subconjunto de documentos relevantes. Dado que *accuracy* (Fórmula (2.5)) es la medida más utilizada para la evaluación de clasificadores binarios, se podría pensar en su uso para la evaluación de sistemas de IR. Sin embargo, hay razones para descartar el uso de *accuracy* en las evaluaciones de sistemas de IR.

$$accuracy = \frac{tp + tn}{tp + fp + fn + tn} \tag{2.5}$$

Normalmente las colecciones de IR no están balanceadas, ya que una gran mayoría de los documentos son considerados no relevantes. Esto provoca que si por ejemplo se utiliza una colección donde alrededor del 99.9 % de los documentos no son relevantes, un sistema optimizado para obtener valores altos de *accuracy* podría considerar a todos los documentos como no relevantes y obtener 0.99 de *accuracy*, el cuál es un valor muy alto. Sin embargo es obvio que etiquetar todos los documentos como no relevantes es un comportamiento que un usuario no desea, ya que él sólo desea documentos relevantes. Para evitar este comportamiento, las medidas de *precisión* y *cobertura* centran la evaluación en la detección de documentos relevantes.

Un problema que presenta el uso de *precisión* y *cobertura* es que ambas medidas suelen variar de forma inversa entre si. Por ejemplo, se puede obtener el valor 1 de *cobertura* recuperando todos los documentos disponibles para todas las consultas, pero esto da lugar a un valor bajo de *precisión*. De hecho, la *precisión* suele disminuir al aumentar el número de documentos recuperados. Normalmente se desea alcanzar un cierto valor de *cobertura* tolerando un determinado porcentaje de falsos positivos.

Dado este problema y con el fin de facilitar la comparación entre sistemas haciendo uso de las dos medidas, es deseable poder contar con una medida que combine *precisión* y *cobertura*. La medida de combinación más utilizada es la media armónica ponderada de ambos valores, conocida como *medida F* (Fórmula

(2.6)) . En esta medida, β es un valor positivo usado para dar más importancia a la *cobertura* o a la *precisión*. Si $\beta < 1$ se da más importancia a la *precisión*, mientras que si $\beta > 1$ cobra más importancia la *cobertura*. Cuando se usa el valor β = 1 se obtiene la Fórmula (2.7), a la cuál se la suele denominar también *medida F* y que representa la media armónica de la *precisión* y la *cobertura*. Para dicho valor de β se da la misma importancia a la *cobertura* y a la *precisión*, por lo que éste suele ser el valor más usado en las evaluaciones de IR. Medida F

$$F = \frac{(\beta^2 + 1) * cobertura * precisión}{\beta^2 * cobertura + precisión} \tag{2.6}$$

$$F_{\beta=1} = \frac{2 * cobertura * precisión}{cobertura + precisión} \tag{2.7}$$

Se utiliza la media armónica para combinar *precisión* y *cobertura* en lugar de la media aritmética debido a que cuando los dos valores difieren en gran medida, la media armónica da un valor más cercano al menor de los dos valores que a la media aritmética de ambos valores. De este modo se proporciona una mejor medida de rendimiento. Por ejemplo, se puede obtener siempre una cobertura del 100 % con el simple hecho de devolver todos los documentos de una colección, con lo que la media aritmética sería siempre del 50 % como mínimo. Esto sugiere que el uso de la media aritmética no es adecuado. Si al decidir devolver todos los documentos se tiene por ejemplo que la colección consta de 5000 documentos y que solo 1 de ellos es relevante, la media armónica de la *precisión* y la *cobertura* sería del 0.04 %, lo cuál parece más realista que el 50.02 % de media aritmética que se obtendría.

2.1.3. Evaluación de Conjuntos Ordenados

Las definiciones de *cobertura* y *precisión* expuestas arriba asumen que todos los documentos devueltos como respuesta a una consulta serán examinados. Sin embargo a un usuario no se le suelen mostrar todos los resultados a la vez, sino ordenados y presentados en una lista de acuerdo con dicho orden. En este contexto el usuario examina la lista empezando por los documentos que el sistema considera más relevantes. A continuación se describen las medidas de evaluación más utilizadas en este contexto.

2.1.3.1. Curva Precisión-Cobertura

El conjunto de documentos devuelto se puede dividir en subconjuntos, para cada uno de los cuáles se puede calcular un valor de *precisión* y otro de *cobertura*. Una forma de medir el rendimiento de un sistema consiste en dibujar una curva *precisión-cobertura* partiendo de dichos valores.

El enfoque que se suele utilizar (se utilizó por ejemplo en las primeras ocho evaluaciones del TREC) consiste en medir la *precisión* en 11 valores distintos de *cobertura* (0.0, 0.1, 0.2, ..., 1.0). En caso de realizarse la evaluación sobre un conjunto de consultas, primero se mide la *precisión* en cada punto de *cobertura* para

cada consulta. Finalmente, para cada nivel de *cobertura* se calcula la media de las *precisiones* de todas las consultas a ese nivel. El resultado se puede dibujar en una curva *precisión-cobertura* a partir de los once puntos calculados. Se puede ver un ejemplo de una curva de este tipo en la Figura 2.1 (página 39).

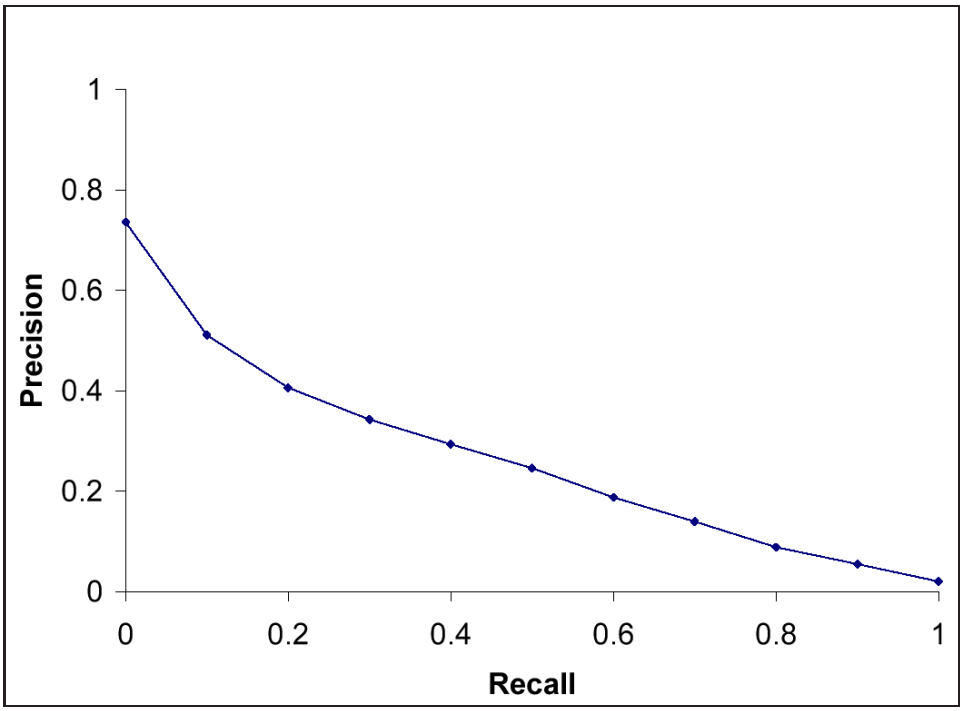

Figura 2.1: Ejemplo de una curva precisión-cobertura en once puntos.

Sin embargo, hay ocasiones en las que se desea poder reducir el rendimiento de un sistema a un simple valor numérico que facilite la comparación entre distintos sistemas.

2.1.3.2. Mean Average Precision

Una de las medidas más comunes en la comunidad del TREC es *Mean Average Precision* (MAP), la cual genera un único valor que resume el rendimiento de un sistema a distintos niveles de cobertura. Además, esta medida ha mostrado tener un buen poder de discriminación[7] y una buena estabilidad[8].

Cuando se realiza la evaluación utilizando *MAP*, para cada consulta se calcula la media de los valores de precisión obtenidos cada vez que se encuentra un documento relevante. El valor final para el conjunto de consultas es la media de

[7]Cuanto más discriminativa es una medida, menos empates habrá entre sistemas y menor será la diferencia necesaria para concluir qué sistema es mejor (Buckley and Voorhees, 2000)

[8]La estabilidad es el error asociado a la conclusión *el sistema A es mejor que el sistema B* (Buckley and Voorhees, 2000)

los valores calculados para cada consulta. Es decir, si el conjunto de documentos relevantes para una consulta $q_j \epsilon Q$ es $\{d_1, ..., d_{m_j}\}$ y R_{jk} es el conjunto de documentos recuperados ordenados desde el primero hasta el documento d_k, entonces se tiene que la fórmula de *MAP* es la definida en la Ecuación (2.8).

$$MAP = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} Precision(R_{jk}) \tag{2.8}$$

Para una única consulta, *MAP* aproxima el área bajo la curva *precisión-cobertura*, mientras que dado un conjunto de consultas *MAP* aproxima el área bajo la curva *precisión-cobertura* de dicho conjunto de consultas. Además, mediante el uso de la media aritmética en *MAP* cada consulta tiene el mismo peso en el valor final.

2.1.3.3. Precision at K

Las anteriores medidas miden la *precisión* en diversos puntos de *cobertura*. Sin embargo, para muchos tipos de aplicaciones, como por ejemplo los buscadores, se tiene que estas medidas pueden no ser de interés para los usuarios. Esto se debe a que en algunos casos lo que interesa a los usuarios es saber cuántos buenos resultados hay entre los primeros documentos, como por ejemplo en las tres primeras páginas de los resultados de un buscador. Un enfoque para evaluar este comportamiento consiste en medir la *precisión* a valores fijos de un bajo número de documentos recuperados, como por ejemplo diez o treinta. A esta medida se le llama *precision at k*, siendo *k* el número de documentos recuperados para los cuáles se calcula la *precisión*. Por ejemplo, para diez documentos se le denomina *precision at 10*. Además, esta medida presenta la ventaja de que no se tiene que estimar el número total de documentos relevantes. Sin embargo, *precision at k* es la menos estable de las medidas más comúnmente utilizadas.

2.1.3.4. R-Precision

Otra alternativa es el uso de *R-precision*. Para su uso es necesario tener un conjunto conocido de documentos relevantes *R*, calculándose la precisión para los primeros $|R|$ documentos recuperados (hay que tener en cuenta que el conjunto *R* podría ser incompleto, como cuando por ejemplo se crea el conjunto a partir de los juicios de relevancia otorgados para los primeros documentos recuperados por determinados sistemas sin comprobar todos los documentos de la colección, tal y como se hace por ejemplo en el TREC). La principal ventaja de esta medida radica en que se ajusta al tamaño del conjunto de documentos relevantes. Esto significa que un sistema perfecto podría obtener el valor 1 de *R-precision* para cualquier consulta mientras que el mismo sistema podría obtener solamente el valor 0.4 de *precision at 20* si solo hay 8 documentos relevantes para una consulta. Además,

R-precision ha mostrado en diversos experimentos empíricos estar bastante correlacionada con *MAP* (Manning et al., 2008).

2.2. Evaluación en Extracción de Información

La Extracción de Información (en inglés Information Extraction, IE) es un tipo de Recuperación de Información donde el objetivo es extraer de forma automática información estructurada a partir de documentos no estructurados. Cuando se desea obtener una determinada información a partir de una colección de textos mediante el uso de sistemas de IR, se obtienen documentos o párrafos donde hay que buscar manualmente la información deseada, como por ejemplo las empresas que se han fusionado durante un determinado año. Sin embargo, los sistemas de IE permiten obtener información más concreta (como por ejemplo compañías, personas, países) y además de forma precisa. Por ejemplo, una aplicación típica de la IE consiste en tomar como entrada un conjunto de documentos en lenguaje natural y poblar a partir de ellos una base de datos con la información extraída.

La información que se extrae en IE consiste principalmente en entidades y relaciones entre dichas entidades, así como atributos que describan a las entidades y a las relaciones (Sarawagi, 2007). De este modo, la IE comprende la creación de representaciones estructuradas, como por ejemplo una base de datos, que contengan un determinado tipo de información especificada previamente y extraída a partir de un texto.

El aumento de la cantidad de información no estructurada, principalmente en la Web, contribuyó al desarrollo de sistemas de IE con el objetivo de hacer más accesible dicho conocimiento. Sin embargo, hay que tener en cuenta que en IE se limita de antemano el tipo de información a extraer. Por tanto, esta tarea es más limitada que la de comprensión de textos, en la cuál se pretende obtener como salida toda la información de un documento.

Como se ha mencionado previamente, las principales subtareas de la IE son la extracción de entidades y las relaciones entre ellas. En concreto, en el ámbito de la IE se habla de entidades nombradas, donde el término entidad nombrada (en inglés Named Entity, NE) ha sido utilizado de forma más o menos semejante en diversas fuentes y se empezó a utilizar a partir de la sexta edición de las conferencias Message Understanding Conference[9] (MUC) (Grishman and Sundheim, 1996). En dichas conferencias, así como en las conferencias Conferences on Computacional Natural Language Learning[10] (CoNLL) (Sang, 2002; Sang and Meulder, 2003), se definió a las NEs como las frases que son identificadores únicos de entidades (organizaciones, personas y localidades), las expresiones temporales (fechas o expresiones de tiempo como puede ser *mediodía*) y las expresiones numéricas (porcentajes o cantidades monetarias).

[9] http://www-nlpir.nist.gov/related_projects/muc/
[10] http://www.cnts.ua.ac.be/conll/

Reconocimiento y Clasificación de Entidades Nombradas

Las primeras conferencias MUC estaban enfocadas a la IE y los organizadores notaron que en dicha tarea era esencial reconocer nombres de personas, organizaciones y lugares, expresiones numéricas y fechas. De este modo fue reconocido que el hecho de detectar en un texto referencias a dichas entidades es una de las subtareas más importantes de la IE y se le denominó Reconocimiento y Clasificación de Entidades Nombradas (en inglés Named Entity Recognition and Classification, NERC) (Nadeau and Sekine, 2007). La importancia que tiene esta detección se debe a que los nombres, fechas y números son importantes a la hora de tratar textos que sirven de fuente a bases de datos (Chinchor et al., 1998), para poblar ontologías de determinados dominios (Tanev and Magnini, 2006) o para aportar información a sistemas de QA o sistemas de IE (Cucerzan and Yarowsky, 1999), motivo por el cuál esta tarea ha recibido tanta atención por parte de la comunidad de procesamiento del lenguaje natural. En el caso de los sistemas de IE se ha convertido en una tarea de suma importancia por su capacidad de proveer de información útil para correferencia y rellenar plantillas (Palmer and Day, 1997).

Normalmente la tarea consiste en dadas unas categorías predefinidas de NEs de interés, localizar de forma automática en un texto todas las palabras o secuencias de palabras que sean instancias de dichas categorías. Por otro lado, una vez detectadas las NEs se entiende por clasificación de las mismas el proceso de otorgar a cada NE una determinada categoría semántica (ej: *George Bush* es de tipo persona). Si por ejemplo se consideran las categorías persona, organización, localidad y expresión temporal, el reconocimiento y clasificación de NEs sobre el texto de la Figura 2.2 (página 42) daría lugar a una anotación similar a la de la Figura 2.3 (página 43).

```
William Henry Gates III (Seattle, Washington, Estados
Unidos, 28 de octubre de 1955) más conocido como Bill
Gates, es un empresario y filántropo estadounidense,
cofundador de la empresa de software Microsoft,
productora del sistema operativo para computadoras
personales más utilizado en el mundo. (...) En 1976,
abandonó la universidad y se trasladó a Albuquerque.
```

Figura 2.2: Ejemplo de texto para anotar entidades nombradas.

Para el programa Automatic Content Extraction[11] (ACE) no solo es importante detectar las entidades sino detectar también lo que se denominan menciones a entidades, que son cualquier referencia que se haga a una entidad (Doddington et al., 2004). Se puede hacer referencia a una entidad haciendo uso de un sustantivo común, de un sintagma nominal o usando un pronombre (LDC, 2005). Por ejemplo, la expresión *el científico más conocido e importante del siglo XX* en el texto de la Figura 2.4 (página 43) es una mención a *Albert Einstein*.

En cuanto al desarrollo de sistemas y publicaciones sobre NERC, hubo un gran crecimiento a partir de 1996 como consecuencia del MUC-6 (Grishman and Sund-

[11] http://www.itl.nist.gov/iad/mig/tests/ace/

[William Henry Gates III]$_{persona}$ ([Seattle]$_{localidad}$, [Washington]$_{localidad}$, [Estados Unidos]$_{localidad}$, [28 de octubre de 1955]$_{exp.temporal}$) más conocido como [Bill Gates]$_{persona}$, es un empresario y filántropo estadounidense, cofundador de la empresa de software [Microsoft]$_{organizacion}$, productora del sistema operativo para computadoras personales más utilizado en el mundo. (...) En [1976]$_{exp.temporal}$, abandonó la universidad y se trasladó a [Albuquerque]$_{localidad}$

Figura 2.3: Ejemplo de texto con las entidades nombradas anotadas.

Albert Einstein, nacido en Alemania y nacionalizado en Estados Unidos en 1940, es el científico más conocido e importante del siglo XX

Figura 2.4: Texto de ejemplo de menciones a entidades.

heim, 1996), y desde entonces se ha mantenido el ritmo de investigación, habiendo numerosos eventos científicos dedicados a la tarea como: HUB-4 (Chinchor et al., 1998), MUC-7 y MET-2 (Chinchor, 1999), IREX (Sekine et al., 2002), CoNLL (Sang, 2002; Sang and Meulder, 2003), ACE (Doddington et al., 2004) y HAREM (Santos et al., 2006). Además, dentro del Language Resources and Evaluation Conference (LREC) ha habido varios talleres y actividades relacionadas con la tarea desde 2000. Una gran proporción de estos trabajos se ha dedicado al estudio de sistemas que trabajan con textos en inglés, pero también ha habido una gran parte dedicada al estudio de los problemas de la multilingualidad y la independencia del idioma. En estos trabajos se puede observar que mientras que los primeros sistemas hacían uso de algoritmos basados en reglas producidas a mano, los sistemas actuales hacen uso cada vez más de métodos de aprendizaje automático.

Las técnicas de aprendizaje automático usadas para el reconocimiento y clasificación de NEs pueden agruparse en métodos supervisados, métodos no supervisados y métodos ligeramente supervisados según el grado de supervisión que se necesite en el proceso de entrenamiento.

Los *métodos supervisados* consiste en aprender las características de ejemplos positivos y negativos de NEs en una gran colección de documentos anotados para luego ser capaces de reconocer dichos ejemplos en documentos no anotados. Sin embargo, la necesidad de disponer de grandes colecciones anotadas, que no siempre están disponibles y tienen asociado un alto coste de creación, es la principal limitación de estos métodos. Para tratar de paliar esta limitación surgen los métodos de aprendizaje *no supervisado* (donde la anotación de la colección se realiza de forma automática) y *semi supervisado* (donde se anota a mano un pequeño subconjunto de la colección y a partir de él se anota automáticamente el resto de la colección).

El desarrollo y evaluación de sistemas de IE ha estado fuertemente influenciado por dos competiciones: el MUC (que se desarrolló en los años 90) y el ACE (que se desarrolló a principios de siglo). Actualmente el programa ACE se ha reformulado como una nueva tarea: Knowledge Base Population[12] (KBP) en la Text Analysis Conference[13] (TAC). A continuación se describen las principales características de estas competiciones.

2.2.1. Message Understanding Conference (MUC)

Las conferencias MUC se desarrollaron entre 1987 (MUC-1) y 1997 (MUC-7) con el objetivo de evaluar sistemas de IE, contribuyendo en gran medida al desarrollo de estos sistemas durante la década en la cuál se celebraron. Los participantes de las evaluaciones, que se desarrollaron dentro del MUC, recibían inicialmente una descripción detallada del escenario sobre el cuál se iba a realizar la evaluación, así como un conjunto de documentos fuente y una descripción de los elementos de información a extraer a partir de dichos documentos. Estos elementos constituían la colección de entrenamiento que se suministraba a los participantes.

Los participantes disponían de un tiempo determinado (de entre 1 y 6 meses) para adaptar sus sistemas al escenario propuesto, después del cuál cada participante recibía un nuevo conjunto de documentos que configuraba la colección de evaluación. Los participantes tenían que usar sus sistemas sobre dicha colección para extraer las piezas de información definidas inicialmente y enviarlas a la organización para su evaluación.

Cada sistema era evaluado comparando su salida con la salida creada por expertos humanos. Las principales medidas de evaluación que se utilizaron fueron *precisión* (Fórmula (2.9)) y *cobertura* (Fórmula (2.10)), siendo N_{clave} el número total de elementos de información de las soluciones, $N_{respuestas}$ el número total de elementos de información dados por un sistema, y $N_{correctas}$ el número de elementos de información que fueron correctamente identificados por el sistema (o lo que es lo mismo, los que coinciden con las soluciones).

$$precisión = \frac{N_{correctas}}{N_{respuestas}} \tag{2.9}$$

$$cobertura = \frac{N_{correctas}}{N_{clave}} \tag{2.10}$$

Como combinación de ambos valores se utilizó la *medida F* (Fórmula (2.11)), la cuál se define como la media armónica entre la *precisión* y la *cobertura*.

$$F = \frac{2 * cobertura * precisión}{cobertura + precisión} \tag{2.11}$$

[12] http://apl.jhu.edu/ paulmac/kbp.html
[13] http://www.nist.gov/tac/

2.2.2. Automatic Content Extraction (ACE)

El programa se inició en 1999 con el objetivo de desarrollar sistemas capaces de extraer conocimiento a partir de diversas fuentes multimedia y con las mismas motivaciones, en general, que el MUC (Doddington et al., 2004).

Al principio del programa se realizó un estudio para tratar de identificar cuáles eran las tareas más importantes de extracción de contenidos para así tenerlas presentes a lo largo del desarrollo del programa. En general, las tareas que se identificaron fueron la extracción de entidades, relaciones y eventos. Además se indicó que no sólo es importante detectar entidades, sino detectar también las menciones a entidades.

Durante el periodo 2000-2001 el programa se centró solo en la detección de entidades, mientras que en el periodo 2002-2003 se incluyó el reconocimiento de relaciones entre entidades. Finalmente, el reconocimiento de eventos fue añadido en 2004. Los eventos son esencialmente una generalización de las relaciones del ACE donde hay que detectar sus participantes. Los participantes son las entidades que toman parte en el evento detectado, cada una de las cuáles tiene un papel en dicho evento como por ejemplo agente, objeto, objetivo, etc.

A la hora de realizar la evaluación se calcula una puntuación distinta para cada tarea (una para reconocimiento de entidades, otra para relaciones y otra para eventos). La puntuación que se otorga en cada tarea se define como la suma de los valores de todos los elementos que da como salida el sistema, normalizado por la suma de los valores de los elementos de referencia (los anotados por expertos humanos) como se muestra en la Fórmula (2.12). La máxima puntuación que se puede obtener en cada tarea es 100 %.

$$Valor_{sistema} = \frac{\sum_i valor\_elemento_i\_de\_sistema}{\sum_j valor\_elemento_j\_de\_referencia} \tag{2.12}$$

El valor que se otorga a cada elemento devuelto por un sistema se calcula comparando sus atributos con los atributos de su elemento de referencia (de entre los anotados por los expertos humanos), con el objetivo de medir con qué precisión se ha realizado la detección. Un sistema consigue tener una salida perfecta cuando su salida concuerda sin ningún error con la de referencia.

2.2.3. Knowledge Base Population (KBP)

El programa ACE dejó de desarrollarse en 2008 para pasar a integrarse dentro del TAC como una tarea llamada Knowledge Base Population (KBP), donde su primera edición tuvo lugar en 2009. El principal objetivo del KBP es promover la investigación y evaluación de sistemas automáticos dedicados a descubrir información sobre entidades e incorporar dicha información a una base de conocimiento ya creada. Hay que tener en cuenta que para actualizar una base de conocimiento ya existente se requiere sintetizar información proveniente de diversos documentos

y asociar las entidades de dichos documentos a elementos que ya existen en la base de conocimiento.

Para realizar la evaluación se distribuye una base de conocimiento junto con una colección de documentos y se plantean dos tareas:

- **Asociación de entidades:** en esta tarea se dan una serie de pares {entidad, documento que contiene a dicha entidad} y los sistemas deben de indicar a qué entidad de la base de conocimiento se refiere la entidad de cada par, teniendo en cuenta que el documento sirve para aportar el contexto necesario para desambiguar dicha entidad. Como medida de evaluación se utiliza el acierto de los sistemas realizando la asociación (Fórmula (2.13)).

- **Rellenar slots de plantillas:** esta tarea se puede ver como una tarea tradicional de IE donde deben buscarse todos los atributos que se pueda para una serie de entidades dadas. No se espera que los sistemas corrijan o modifiquen los valores de la base de conocimiento dada inicialmente, sino simplemente que añadan información a ésta. Para evaluar esta tarea se utilizan dos medidas de evaluación. La primera medida valora solamente la corrección en la detección de atributos, calculando cuántos slots se han rellenado correctamente de entre todos los que había que rellenar (Fórmula (2.14)). La segunda medida se centra en evaluar el acierto de los sistemas asociando los atributos detectados a entidades de la base de conocimiento. Para ello, se multiplica el valor de la primera medida (la cuál mide el acierto detectando atributos) por la precisión obtenida asociando cada uno de los atributos a entidades de la base de conocimiento (Fórmula (2.15)).

$$acierto = \frac{\#asociaciones\_correctas}{\#asociaciones\_a\_realizar} \tag{2.13}$$

$$acierto\_rellenar\_slots = \frac{\#respuestas\_correctas}{\#slots} \tag{2.14}$$

$$acierto\_asociar\_slots = acierto\_rellenar\_slots * \frac{\#asociaciones\_correctas}{\#asociaciones\_realizadas} \tag{2.15}$$

La evaluación propuesta en el KBP es similar a la realizada en el Web People Search[14] (WePS) (Artiles, 2009), donde hay una tarea que propone hacer clustering de las páginas Web que contienen información acerca de una determinada persona, y otra tarea donde se deben de extraer atributos de esa persona (de forma similar a la tarea de rellenar slots de plantillas del KBP).

[14] http://nlp.uned.es/weps/

2.3. Evaluación de Búsqueda de Respuestas

La Búsqueda de Respuestas (en inglés Question Answering, QA) es la tarea automática que tiene como finalidad encontrar respuestas concretas a necesidades precisas y arbitrarias de información formuladas por los usuarios (Vicedo, 2003). Esto significa que a diferencia de la tarea de IR, en QA las consultas se realizan con preguntas formuladas en lenguaje natural (como por ejemplo *¿Cuál es la capital de Portugal?*) y se obtiene como resultado un fragmento de texto con la respuesta precisa (como por ejemplo *Lisboa*), en lugar de una lista de documentos donde hay que buscar manualmente la respuesta. El hecho de que este tipo de sistemas devuelva respuestas exactas en lugar de documentos relevantes y que se tengan que procesar preguntas formuladas en lenguaje natural tiene como consecuencia que se haga uso de métodos más complejos que los utilizados en IR.

Los sistemas de QA tienen diversas aplicaciones, las cuales dependen de varios aspectos como, por ejemplo, si las respuestas se obtienen a partir de datos estructurados, datos semiestructurados o texto libre. Además, hay sistemas de QA que trabajan sobre dominios restringidos (como por ejemplo los participantes en la evaluación propuesta sobre documentos legales en el ResPubliQA 2009 (Peñas et al., 2010)), lo que les permite a estos sistemas utilizar conocimiento específico del dominio, a diferencia de los sistemas que no tienen restringido el dominio de aplicación.

Por otro lado, hay sistemas de QA que están enfocados a la interacción con el usuario, permitiendo que el proceso de búsqueda vaya refinándose con la intervención del usuario. Este tipo de sistemas tienen en cuenta a los usuarios y sus interacciones con las preguntas y los documentos donde se realiza la búsqueda (Hersh, 2006).

El estudio realizado en este capítulo se centra en sistemas de QA como los definidos en las evaluaciones del TREC, que realizan la búsqueda de respuestas sobre texto libre en colecciones de documentos suministradas previamente, y sin interactuar con el usuario.

2.3.1. Arquitectura Genérica

En esta sección se describe la arquitectura básica de un sistema de QA. Esta descripción se realiza debido a que es relevante para los propósitos de este trabajo, en concreto para situar a los sistemas de Validación de Respuestas (que se empezarán a tratar en la sección 2.4 de la página 57) y sus funcionalidades dentro del contexto de un sistema de QA.

La idea básica de la arquitectura de un sistema de QA es ir acotando la selección de texto gradualmente con el propósito de eliminar los fragmentos de texto en los que se cree que no está la respuesta, y mantener aquellos en los que se espera que esté contenida. Con este propósito, se utilizan métodos rápidos para preseleccionar el texto con la respuesta y métodos más laboriosos para encontrar y extraer el texto exacto de la respuesta (Hovy et al., 2001; Moldovan et al., 2000; Prager et

al., 2000). Las fases que se realizan principalmente en un sistema de QA son las mostradas en la Figura 2.5 (tomada de (Vicedo, 2003)) de la página 48:

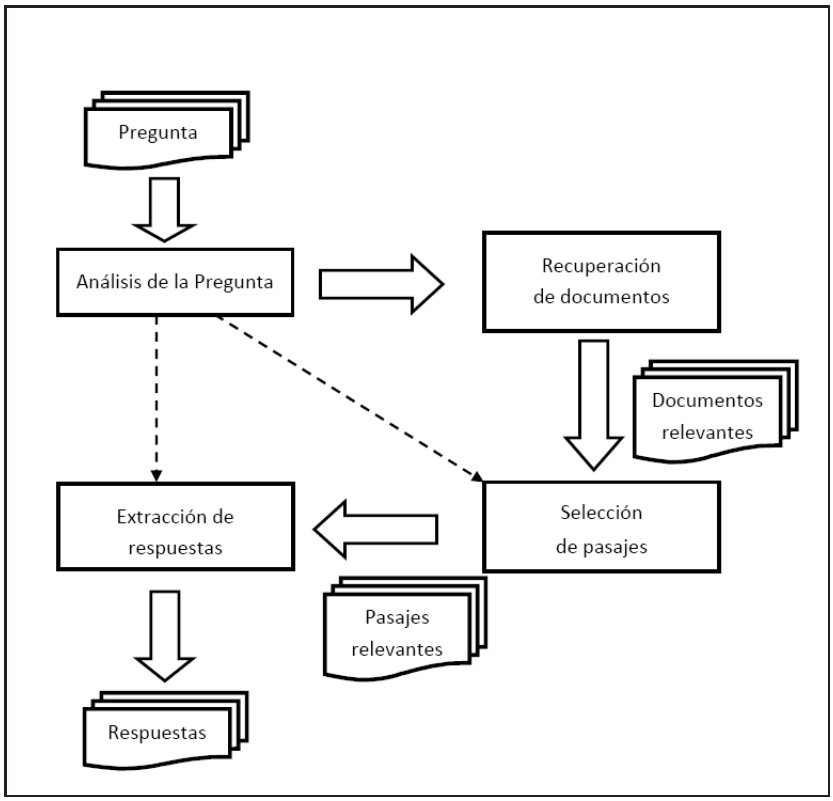

Figura 2.5: Arquitectura básica de un sistema de Búsqueda de Respuestas.

1. **Análisis de la pregunta**. Esta fase constituye uno de los procesos fundamentales de un sistema de QA puesto que la información obtenida en esta etapa ha de servir para guiar la búsqueda en las fases siguientes. Es por ello que esta fase tiene un papel muy importante en el rendimiento final de un sistema de QA. Básicamente, la información que se extrae en esta fase se puede dividir en dos tipos (Vicedo, 2003):

   - Información que permita localizar los documentos y fragmentos de texto relevantes de entre todos aquellos contenidos en la colección donde se realiza la búsqueda.
   - Información sobre la respuesta a obtener. En este caso el objetivo es poder contar con información que ayude a extraer la respuesta precisa de entre los fragmentos de texto que la contienen.

   Con el fin de obtener esta información se realizan varios procesos, siendo los siguientes los más comúnmente utilizados:

- **Determinar el tipo esperado de respuesta.** Para encontrar la respuesta correcta a una pregunta es importante conocer primero qué es lo que se está buscando. Para ello, la pregunta se clasifica dentro de un conjunto limitado de clases predefinidas de antemano (Hirschman and Gaizauskas, 2001). La forma más básica de clasificación distingue entre unas pocas clases de acuerdo con la partícula interrogativa (qué, quién, etc), mientras que las clasificaciones más complejas realizan un análisis más profundo y hacen uso de taxonomías más amplias (Li and Roth, 2002).
- **Detectar el foco de la pregunta.** Hay ocasiones en las cuáles el tipo esperado de la respuesta es muy genérico y no aporta información relevante sobre lo que se está buscando. En estos casos se puede determinar con más exactitud el tipo de respuesta esperada mediante la detección del *foco de la pregunta*. El foco es una palabra o secuencia de palabras que define a la pregunta y que es útil para desambiguar la pregunta. Por ejemplo, para la pregunta *¿En qué año nació Gandhi?*, el foco sería *año*. El foco se encuentra normalmente cerca de la partícula interrogativa (qué, quién, dónde, etc) y no suele aparecer en los documentos o fragmentos donde se encuentra la respuesta, por lo que no debe de ser usado en la lista de palabras clave que se va a utilizar para recuperar documentos relevantes.
- **Extraer las palabras clave de la pregunta.** Antes de realizar el proceso de recuperación de documentos y fragmentos relevantes se ha de formar una consulta con palabras clave. Este proceso se puede realizar de forma sencilla utilizando los términos de la pregunta que no sean palabras vacías (ni el foco de la pregunta) o de forma más compleja expandiendo dichos términos con sinónimos o hiperónimos (Hovy et al., 2001). Por otro lado, hay sistemas que tienen en cuenta el tipo esperado de respuesta para obtener la lista de palabras claves. En función del tipo esperado de respuesta, estos sistemas aplican una serie de heurísticas para obtener las palabras clave que formarán la consulta que se utilizará en la siguiente fase.

2. **Selección de documentos o pasajes relevantes**. La recuperación de documentos y pasajes relevantes es el segundo gran módulo en el proceso de un sistema de QA. Su objetivo consiste en encontrar documentos que se espera que contengan la respuesta correcta. Es importante realizar bien este proceso puesto que la respuesta final será buscada solamente sobre los documentos recuperados.

   Los términos clave de la pregunta se buscan en la colección para seleccionar los documentos que se consideren más relevantes, los cuáles se devuelven ordenados de acuerdo a un ranking de relevancia definido de antemano. Normalmente se suele recuperar un número fijo de documentos, aunque también

se pueden recuperar solamente los documentos que superan un determinado valor umbral de relevancia. La mayoría de los sistemas de QA utilizan sistemas de IR para llevar a cabo esta tarea.

A continuación se pasa por un proceso de refinamiento en el que se seleccionan los fragmentos de texto que se consideran más relevantes a la pregunta, los cuáles son enviados a la siguiente fase.

Para reducir el tiempo de respuesta del sistema, en esta fase se suele realizar la búsqueda de documentos sobre una versión preprocesada de la colección de documentos. Este preprocesado permite obtener un índice donde buscar los documentos que contienen las palabras clave, sin necesidad de tener que explorar todos los documentos en tiempo real.

3. **Extracción de respuestas**. Este proceso tiene como objetivo la localización de la respuesta dentro de los fragmentos recibidos de la fase anterior. La dificultad de esta tarea depende del tipo de respuesta esperada y es en esta fase donde se concentran los esfuerzos de la mayoría de sistemas y donde se han propuesto las técnicas más variadas. Hay que tener en cuenta que el uso de algunas de estas técnicas podría ralentizar en exceso el tiempo de respuesta del sistema, por lo que el procesamiento llevado a cabo en esta fase podría simplificarse en sistemas de QA de tiempo real.

   La extracción de la respuesta se puede realizar eliminando de los fragmentos relevantes los términos que aparezcan en la pregunta (puesto que se está preguntando por algo distinto a lo que contiene la pregunta), y tomando de entre los términos que quedan aquellos que sean del tipo esperado. Una vez se tengan estas respuestas candidatas se puede seleccionar la respuesta final utilizando un determinado criterio.

   Otra forma de realizar la extracción es mediante el uso de patrones de respuestas. Estos patrones pueden utilizar sólo información léxica o también información sintáctica, y se pueden crear de forma manual (donde el principal inconveniente es que si se cambia el dominio hay que volver a generar a mano los patrones) (Soubbotin, 2001), o utilizando aprendizaje automático (Lin and Pantel, 2001).

4. **Validación de respuestas**[15] Hay sistemas que una vez localizada la respuesta tratan de demostrar que realmente la respuesta encontrada responde a la pregunta realizada, es decir, realizar una validación de la respuesta. Dado que este procesamiento tiene relación con la extracción de la respuesta, podría incluirse dentro de dicha fase. Sin embargo, dados los intereses de este trabajo se ha preferido incluirla como una fase adicional y posterior.

[15] Esta fase no está incluida en la Figura 2.5 ya que no forma parte de la arquitectura clásica de un sistema de QA. Sin embargo, se ha decidido incluir una reseña debido a que este trabajo se centra en esta fase

2.3.2. Características de la Evaluación

Durante la última década se han explorado diversos métodos para evaluar a los sistemas de QA en el TREC, NTCIR y CLEF, siendo la tarea de QA del TREC-8 la primera evaluación de este tipo de sistemas (Voorhees and Tice, 1999). Mientras que el TREC ha estado enfocado al desarrollo de sistemas monolingües en inglés, en 2003 se empezó a organizar una tarea de QA dentro del marco del CLEF con el objetivo de evaluar y fomentar el desarrollo de sistemas multilingües (Magnini et al., 2004). Esta tarea se desarrolla tanto para sistemas monolingües como bilingües y en diferentes idiomas europeos. De hecho, en la edición de 2008 fue posible participar en búlgaro, inglés, francés, alemán, italiano, portugués, rumano, griego, vasco y español. Por otro lado, desde 2002 se celebra también una tarea de QA en el ámbito del NTCIR, en este caso enfocada a las lenguas asiáticas.

Los métodos de evaluación utilizados se pueden clasificar de acuerdo a los siguientes criterios:

- **Número de respuestas permitidas por pregunta.** Hay distintas medidas en función de si se requiere una sola respuesta (Herrera et al., 2004; Magnini et al., 2004; Voorhees, 2002, 2003), se permiten varias respuestas pero solo se necesita una correcta (Herrera et al., 2004; Kato et al., 2005; Sakai et al., 2008; Voorhees and Tice, 1999) o se requieren varias respuestas como en el caso de listas de elementos (Gey et al., 2005; Herrera et al., 2004; Voorhees, 2003).

- **Confianza del sistema en la corrección de su respuesta.** La confianza de un sistema en sus respuestas se suele evaluar a partir de los rankings que genera dicho sistema (Gey et al., 2005; Sakai et al., 2008; Voorhees and Tice, 1999; Voorhees, 2002). Sin embargo, ha habido métodos que han hecho uso del valor de confianza generado para cada respuesta para calcular el coeficiente de correlación de Pearson (Herrera et al., 2004) entre estos valores y si las respuestas eran o no correctas.

- **Niveles de relevancia dados para cada respuesta por los evaluadores humanos.** En este caso se puede optar entre la consideración binaria tradicional de corrección o considerar múltiples valores de relevancia de la respuesta (Kato et al., 2005).

- **Cantidad de información relevante contenida en la respuesta.** Las respuestas a preguntas factuales (preguntas que esperan una respuesta corta, como por ejemplo el nombre de una persona) deben de cubrir completamente la información solicitada. Sin embargo, la evaluación de preguntas más complejas debe dar valores parciales a las respuestas que cubran de forma parcial la necesidad de información. En estos casos son comunes las medidas basadas en *nuggets* y pirámides (Dang et al., 2006; Lin and Demner-Fushman, 2006).

A continuación se describen los métodos de evaluación que se usan más habitualmente en función del tipo de pregunta.

2.3.3. Evaluación de Preguntas Factuales

2.3.3.1. Mean Reciprocal Rank

Las preguntas de tipo factual preguntan acerca del nombre de una persona, un lugar, el día en el que algo ha ocurrido, etc. Al principio de las evaluaciones de sistemas de QA se utilizaba la medida de evaluación *Mean Reciprocal Rank* (MRR) cuando para una misma pregunta bastaba con obtener una sola respuesta válida, pero se solicitaban varias respuestas ordenadas por orden de relevancia (Fukumoto et al., 2004a; Voorhees and Tice, 1999).

Cuando se utiliza esta medida, cada pregunta respondida recibe un valor igual al inverso del ranking de la primera respuesta correcta a dicha pregunta, o 0 en caso de que ninguna de las respuestas sea correcta. Es decir, si la primera respuesta a la pregunta es correcta se otorga 1, si la primera fuese incorrecta pero la segunda correcta se otorgaría $\frac{1}{2}$ y así sucesivamente. A este valor por pregunta se le denomina *Reciprocal Rank* (RR). La medida RR se puede ver de la siguiente manera: en el caso de respuestas devueltas en un ranking, el número de comprobaciones necesarias para obtener una respuesta correcta es uno cuando la primera respuesta del ranking es correcta, dos cuando lo es la segunda y no la primera, y así sucesivamente dado que las comprobaciones se realizan según el orden dado en el ranking. Es decir, el número de comprobaciones realizadas, que puede verse como un coste, es igual a la posición más alta en el ranking de las respuestas correctas y RR es el recíproco de dicho coste. Para un conjunto de preguntas se calcula la media de los valores de RR de cada pregunta, valor que se denomina *MRR*.

2.3.3.2. Mean Reciprocal Cost

Similar a MRR pero cuando se devuelven varias respuestas sin que formen un ranking entre ellas existe la medida *Mean Reciprocal Cost* (MRC) (Kato et al., 2005). El coste de comprobar las respuestas en este escenario dada una lista de *m* elementos con *n* respuestas correctas (*c(n,m)*) se define de la siguiente manera: primero se selecciona aleatoriamente un elemento de la lista y se comprueba. Si este elemento es correcto el coste es 1. Si no es correcto, se repite el proceso con los demás *m-1* elementos, lo que tendría un coste de *c(n,m-1)*. La probabilidad de no obtener la respuesta correcta a la primera es $\frac{m-n}{m}$, obteniéndose la siguiente fórmula de recurrencia: $c(n,m) = 1+c(n,m-1)*\frac{m-n}{m}$. Resolviendo esta fórmula se obtiene $c(n,m) = \frac{m+1}{n+1}$. El valor otorgado a cada pregunta es el inverso de este coste y se denomina *Reciprocal Cost* (RC). Finalmente, se calcula *MRC* como la media de *RC* sobre todo el conjunto de preguntas.

2.3.3.3. Accuracy

Como para los sistemas de QA se requieren resultados muy precisos, las campañas de evaluación tendieron a restringir el número de respuestas devueltas a una sola y usar *accuracy* (proporción de preguntas respondidas correctamente en lugar de *MRR*) como medida de evaluación (Magnini et al., 2004; Voorhees, 2003). *Accuracy* es una medida más simple, más fácil de entender y que recompensa a sistemas con un comportamiento más preciso.

2.3.3.4. Permitir la Posibilidad de no Responder

La primera ocasión en la cuál los sistemas participantes tuvieron la oportunidad de dejar una pregunta sin contestar fue en el TREC 2001 (Voorhees, 2001a). En dicha edición se usaron como medidas de evaluación secundarias (la principal fue *MRR*) los siguientes valores:

- proporción de preguntas respondidas
- proporción de preguntas respondidas que fueron respondidas correctamente

Sin embargo, no se propuso ninguna medida como combinación de estos dos valores, los cuáles son difíciles de interpretar por separado y hacen difícil la comparación entre sistemas. Además, en el TREC 2001 hubo pocos participantes que decidiesen dejar algunas preguntas sin responder y por tanto no se generaron datos suficientes como para realizar un análisis adecuado de los resultados.

2.3.3.5. Confidence Weigted Score

La opción de dejar preguntas sin responder se desechó en la siguiente edición del TREC (TREC 2002), proponiéndose otra medida (Voorhees, 2002). En esta edición se debía dar exactamente una respuesta por pregunta y todas las respuestas devueltas debían ser ordenadas de acuerdo a la confianza del sistema en la corrección de las mismas. Dadas estas condiciones, los sistemas fueron evaluados usando la medida *Confidence Weigted Score* (CWS) (Fórmula (2.16), donde *n* es el numero de preguntas y *C(i)* es el numero de respuestas correctas hasta la posición *i* del ranking).

$$CWS = \frac{1}{n}\sum_{i=1}^{n}\frac{C(i)}{i} \tag{2.16}$$

La formulación de CWS está inspirada en la definición de la precisión media (en inglés *Average Precision*, AP) sobre el ranking para una sola consulta en IR, cuya formulación se muestra en la Ecuación (2.17), donde *R* es el número de resultados relevantes, *r* es una posición en el ranking devuelto por el sistema e *I(i)* una función que devuelve 1 si la respuesta *i-ésima* es correcta y 0 en caso contrario. Si se adapta *AP* al caso en el cuál solo se solicita una respuesta por pregunta, *R*

es igual a *n* (número de preguntas) en *CWS*. Sin embargo, en *AP* se incrementa el valor en una posición del ranking solo cuando hay un resultado relevante en dicha posición (ya que se usa la función *I(i)*), mientras que en *CWS* todas las posiciones del ranking añaden valor sin importar si hay o no un resultado relevante en dicha posición. De este modo, *CWS* da más valor a algunas preguntas que a otras. En concreto, las preguntas cuyas respuestas están colocadas al principio del ranking son las que contribuyen en mayor medida al valor final, mientras que las preguntas con respuestas al final del ranking practicamente no aportan valor.

$$AP = \frac{1}{R} \sum_{r} I(r) \frac{C(r)}{r} \tag{2.17}$$

Aunque *CWS* fue diseñada con el fin de mejorar la capacidad de los sistemas a la hora de emitir juicios sobre la confianza en sus respuestas, su uso como medida de evaluación en QA fue discutido y se descartó en las siguientes ediciones del TREC en favor del uso de *accuracy*, debido a que *accuracy* se centra más en evaluar la capacidad de los sistemas encontrando respuestas correctas. Desde entonces, las evaluaciones del TREC dejaron de desarrollar medidas de evaluación enfocadas a promover un mayor conocimiento de los sistemas sobre sus respuestas.

2.3.3.6. K y K1

Los trabajos para evaluar la seguridad de un sistema de QA en la corrección de sus respuestas continuaron en el CLEF 2004 (Herrera et al., 2004), donde los sistemas de QA debían de devolver junto con cada respuesta un valor entre 0 y 1 para indicar la confianza del sistema sobre la corrección de dicha respuesta (1 representa la máxima confianza y 0 que no se confía en la corrección de la respuesta).

En el CLEF 2004 se propuso el uso del coeficiente de correlación Pearson entre el valor de confianza de cada respuesta y si la respuesta era o no correcta (utilizando el valor 1 si la respuesta era correcta y 0 en caso contrario). Debido a que el valor obtenido no suponía una medida del rendimiento de un sistema de QA, los autores propusieron el uso de dos nuevas medidas llamadas *K* y *K1*. El objetivo de estas medidas es premiar a los sistemas que otorgan de manera acertada sus valores de confianza: valores altos a las respuestas correctas y valores bajos a las respuestas incorrectas. Las medidas propuestas estaban basadas en una función de utilidad que valía *-1* si la respuesta era incorrecta y *1* si la respuesta era correcta. El valor de esta función de utilidad se multiplicaba por el valor de confianza de cada respuesta. La formulación de *K1* se puede ver en la Ecuación (2.18), mientras que la medida *K* es una variación de *K1* para cuando se permite devolver más de una respuesta por pregunta. La formulación de *K* se puede ver en la Ecuación (2.19), donde *R(i)* es el número total de respuestas correctas y distintas que se conocen para la pregunta *i*; *respondidas(i)* es el número de respuestas dadas por el sistema a la pregunta *i*; y *eval(r)* tiene tres posibles valores:

$$eval(r) = \begin{cases} 1 \;\; si\, la\, respuesta\, r\, es\, correcta \\ 0 \;\; si\, la\, respuesta\, r\, es\, repetida \\ -1 \;\; si\, la\, respuesta\, r\, es\, incorrecta \end{cases}$$

$$K1 = \frac{\sum_{i \epsilon \{correctas\}} val\_confianza(i) - \sum_{i \epsilon \{incorrectas\}} val\_confianza(i)}{\#preguntas} \tag{2.18}$$

$$K = \frac{1}{\#preguntas} \sum_{i \epsilon \{preguntas\}} \frac{\sum_{r \epsilon respuestas(i)} val\_confianza(r) * eval(r)}{max\{R(i), respondidas(i)\}} \tag{2.19}$$

El valor final de *K1* es difícil de interpretar: un valor positivo no tiene por qué significar tener más respuestas correctas que incorrectas, sino que la suma de los valores de confianza de las respuestas correctas es superior a la suma de los valores de las respuestas incorrectas. Puede que esta dificultad de interpretación sea la razón por la cuál no se le ha dado a esta medida mucha importancia en el CLEF.

2.3.3.7. Medidas que usan Diversos Grados de Relevancia

Por otro lado, en la serie de conferencias NTCIR se han preocupado de otorgar a las respuestas juicios que indiquen más que un simple valor binario acerca de si la respuesta es correcta o incorrecta. Para ello, los evaluadores del NTCIR-5 otorgaban a cada respuesta varios valores del rango [0,1] para indicar la calidad de la misma (Kato et al., 2005). Los aspectos que se tienen en cuenta para indicar la calidad de las respuestas son:

- Por un lado, la calidad de la expresión de la respuesta hace relación a la granularidad de la propia respuesta. Para ciertas preguntas que por ejemplo esperan como respuesta una fecha, es más correcto responder *3 de enero de 2000* que simplemente *2000*.
- Por otro lado, la respuesta puede ser correcta en un documento por un error, y sin embargo ser incorrecta en el conjunto global de todos los documentos que contiene la colección. Por este motivo se mide la calidad de la información dada respecto a la colección donde se buscan las respuestas.
- Se considera también importante fomentar que no se devuelvan respuestas repetidas, penalizando los casos en los cuáles una misma respuesta es devuelta más de una vez para una misma pregunta.

2.3.4. Evaluación de Preguntas de Tipo Lista

En las preguntas de tipo lista se espera como respuesta una sucesión de elementos que cumplan un determinado criterio solicitado en la pregunta. Al evaluar este tipo de preguntas es importante comprobar tanto que las respuestas dadas son correctas, como que se ha devuelto el mayor número de elementos correctos.

Por tanto, al realizar la evaluación se calcula para cada pregunta la media armónica *F* (Fórmula (2.22)) entre la *precisión* (proporción de respuestas dadas que son correctas) (Fórmula (2.20)) y la *cobertura* (proporción de respuestas correctas encontradas) (Fórmula (2.21)). Para un conjunto de preguntas se calcula la media de los valores de *F* obtenidos para cada pregunta (Voorhees, 2003; Fukumoto et al., 2004a).

$$precisión = \frac{\#respuestas\ devueltas\ correctas}{\#respuestas\ devueltas} \tag{2.20}$$

$$cobertura = \frac{\#respuestas\ devueltas\ correctas}{\#total\ respuestas\ correctas\ en\ colección} \tag{2.21}$$

$$F = \frac{2 * cobertura * precisión}{cobertura + precisión} \tag{2.22}$$

2.3.5. Evaluación de Preguntas Complejas

En preguntas más complejas que las de tipo factual en las que se requieren respuestas más largas y que contengan información importante acerca de personas, organizaciones o descripciones de objetos, el mero concepto binario de correcto o incorrecto parece insuficiente para evaluar una respuesta.

Es por ello que tanto en el TREC desde su edición de 2003 (Voorhees, 2003) como en el NTCIR a partir de su edición de 2008 (Mitamura et al., 2008) se ha empezado a evaluar este tipo de respuestas atendiendo a las piezas de información (a las cuáles se les denomina *nuggets*) presentes en las respuestas. Para ello, se define un *nugget* como un hecho para el cuál el evaluador puede tomar una decisión binaria sobre si una respuesta contiene o no a dicho *nugget*.

Para comenzar el proceso de evaluación de estas preguntas se crea una lista con los *nuggets* respuesta de una pregunta. Para crear esta lista se utilizan las respuestas generadas por todos los sistemas participantes en la evaluación más las que obtuvo el experto humano que creó la pregunta. A continuación el evaluador decide cuáles de estos *nuggets* se consideran vitales en el sentido de que tienen que aparecer en una respuesta para que dicha respuesta sea dada por buena. En función de esta lista se anotan los *nuggets* de las respuestas devueltas por los sistemas.

A la hora de evaluar estas preguntas los sistemas no son penalizados por recuperar *nuggets* no vitales. Para ello, el valor de *cobertura* de *nuggets* se calcula solo sobre los *nuggets* vitales. Por otro lado, a la hora de calcular la *precisión* se presenta el problema de no conocer el número exacto de *nuggets* recuperado en una

respuesta, por lo que se usa como aproximación la longitud de la respuesta para estimar el número de *nuggets* de cada respuesta.

A la hora de combinar la *cobertura* con la *precisión* se calcula su media armónica (*medida F*) dando más importancia a la *cobertura* sobre la *precisión* (por ejemplo, en el TREC 2003 se le dio 5 veces más importancia a la *cobertura* que a la *precisión*, mientras que en las siguientes evaluaciones del TREC y en el NTCIR se le dio 3 veces más importancia).

La evaluación de este tipo de preguntas se diseñó de este modo para hacer a la evaluación dependiente solo del contenido de la respuesta y no de su estructura particular.

2.3.6. Combinación de Resultados

En evaluaciones con distintos tipos de pregunta, donde cada tipo es evaluado de una manera diferente, surge el problema de cómo combinar los resultados obtenidos por los diversos métodos de evaluación en un solo valor que resuma el rendimiento de un sistema. Este problema ha sido tratado más profundamente en el TREC que en el CLEF, puesto que en este último foro todos los tipos de pregunta se evalúan igual usando *accuracy*. Por otro lado, en el NTCIR los distintos tipos de pregunta se evalúan como parte de tareas distintas, por lo que no se realiza combinación alguna de resultados.

La primera combinación de resultados se realizó en el TREC 2003 (Voorhees, 2003), donde el valor final fue la suma ponderada de los valores obtenidos en preguntas de tipo factual, de tipo lista y preguntas complejas. Tanto en la edición de 2003 como en la de 2004 la mitad del valor final correspondía a las preguntas factuales, mientras que la otra mitad correspondía a las de tipo lista y a las complejas en partes iguales, recibiendo más peso las preguntas de tipo factual debido a que la mayoría de las preguntas formuladas eran de este tipo (Voorhees, 2004). Sin embargo, desde la edición de 2006 cada tipo de pregunta tiene el mismo peso en el valor de evaluación final (Dang et al., 2006).

2.4. Evaluación de Validación de Respuestas

Hay autores que descomponen a los sistemas de QA en dos etapas (Tonoike et al., 2004): una primera que se dedica a recolectar respuestas candidatas y una segunda que consiste en la validación de cada una de las respuestas obtenidas. La primera de dichas etapas ha sido ampliamente estudiada mientras que la referente a la validación ha comenzado a recibir atención por parte de la comunidad científica recientemente.

Básicamente, la tarea de Validación de Respuestas (en inglés Answer Validation, AV) consiste en decidir si las respuestas producidas por un sistema de QA son o no correctas. Más concretamente, un sistema de AV recibe una *Pregunta* y una *Respuesta* y devuelve un valor indicando si la *Respuesta* es o no correcta y

en qué grado. Cuando un sistema de AV considera que una respuesta es correcta, entonces se dice que valida la respuesta. Por otro lado, cuando se considera que la respuesta es incorrecta, entonces se dice que el sistema rechaza la respuesta. De este modo, un sistema de AV se puede utilizar para eliminar respuestas erróneas de entre las respuestas candidatas a una pregunta, así como para crear un ranking de las respuestas generadas por un sistema de QA (Magnini et al., 2002a).

El desarrollo de técnicas automáticas para llevar a cabo la validación de respuestas es de gran interés para el desarrollo de sistemas de QA, ya que se espera que permita mejorar los resultados actuales disminuyendo la cantidad de respuestas incorrectas generadas. Por otro lado, el desarrollo de sistemas de AV permitiría también la evaluación automática de sistemas de QA, lo cuál ayudaría a facilitar y hacer más rápido el desarrollo de sistemas de QA (Breck et al., 2000). Esto se debe a que un deseo de los desarrolladores de sistemas de QA es poder modificar y evaluar sus sistemas de forma frecuente. Sin embargo el procedimiento para evaluar manualmente sistemas de QA sobre grandes conjuntos de datos es largo y costoso, por lo que el hecho de poder contar con procedimientos automáticos de evaluación sería de gran ayuda y permitiría reducir el coste asociado (tanto en tiempo como en dinero) a la evaluación. Además, al contar con métodos de evaluación automáticos sería posible el desarrollo de sistemas de QA basados en enfoques de generación y prueba. De este modo, un sistema de QA podría reconfigurarse automáticamente mientras busca respuestas a una pregunta hasta que encuentre una que considere correcta (Magnini et al., 2002a).

2.4.1. Validación vs. Selección de Respuestas

Los sistemas de AV pueden desempeñar diversas tareas dentro de los sistemas de QA, siendo las dos principales funciones la validación y la selección de respuestas. En los siguientes subapartados se describe en qué consiste cada una de estas dos tareas.

2.4.1.1. Validación

Cuando un sistema de AV realiza tareas de validación dentro del contexto de un sistema de QA, su tarea consiste en dado un conjunto de respuestas a una pregunta eliminar las respuestas que considera incorrectas. El objetivo es por tanto reducir la cantidad de respuestas incorrectas a la salida del sistema de QA. Es decir, al sistema de AV se le puede ver como un filtro que debe de eliminar las respuestas incorrectas. Realizando validación se pretende mejorar la precisión del sistema de QA, lo cuál conllevaría una mejora en resultados en cuanto a las medidas de evaluación *MRR* y *MRC* (descritas en la Sección 2.3.3 de la página 52).

Este proceso de validación se puede aplicar en dos fases distintas según Harabagiu and Hickl (2006) dentro de la arquitectura clásica de un sistema de QA vista en la Sección 2.3.1 (página 47):

- Por un lado se puede aplicar la validación a la lista de respuestas candidatas generadas para cada pregunta con el objetivo de devolver solamente respuestas correctas al usuario. El emplazamiento de un módulo de AV en esta fase haría que la arquitectura final sea como la mostrada en la Figura 2.6 (página 60). De este modo, tras el uso del sistema de AV se podrían tener a la salida las mismas respuesta que antes (si se considera que todas las respuestas candidatas son correctas) o una cantidad menor (si algunas respuestas candidatas han sido consideradas como incorrectas).

- Un módulo de AV se puede situar también a la salida de la fase de selección de pasajes con el fin de validar los pasajes relevantes a una pregunta dada. En este caso la lista de pasajes relevantes sería filtrada por el módulo de AV. El hecho de realizar un filtrado de pasajes relevantes provoca que se reduzca el número de pasajes a tratar en la fase de extracción de la respuesta. Dado que en la fase de extracción se suele realizar un procesamiento más complejo que en otras fases, la validación de pasajes contribuye a reducir el tiempo de respuesta del sistema de QA. En este caso la arquitectura resultante sería la mostrada en la Figura 2.7 (página 61).

Evidentemente se puede aplicar validación simultaneamete en ambas fases quedando una arquitectura como la mostrada en la Figura 2.8 (página 62).

2.4.1.2. Selección

Un módulo de AV que realiza selección se utiliza en un sistema de QA que tiene que devolver una única respuesta por pregunta. En este escenario el módulo de AV recibe todas las respuestas candidatas a una determinada pregunta y elige de entre ellas la que considera que tiene mayores opciones de ser correcta. De este modo la respuesta elegida se convierte en la respuesta del sistema a la pregunta de entrada, dando lugar a la arquitectura mostrada en la Figura 2.9[16] (página 62). Por tanto, el objetivo de un módulo de AV dentro de este escenario es maximizar la precisión del sistema de QA devolviendo una respuesta por pregunta (lo cuál se evalúa en QA por medio de *accuracy* como se indicó en la Sección 2.3.3.3 de la página 53).

Pero además, los módulos de AV también se pueden utilizar para llevar a cabo la selección en sistemas multi-flujo de QA. Un sistema multi-flujo (en inglés multi-stream) de QA está compuesto por varios sistemas individuales de QA que reciben las mismas preguntas y generan respuestas a dichas preguntas como se muestra en la Figura 1.2 (página 28). El objetivo de este tipo de sistemas es mejorar los resultados obtenidos por un único sistema de QA mediante la combinación de resultados.

[16] En esta Figura se ha decidido omitir las distintas fases de un sistema de QA puesto que no son relevantes

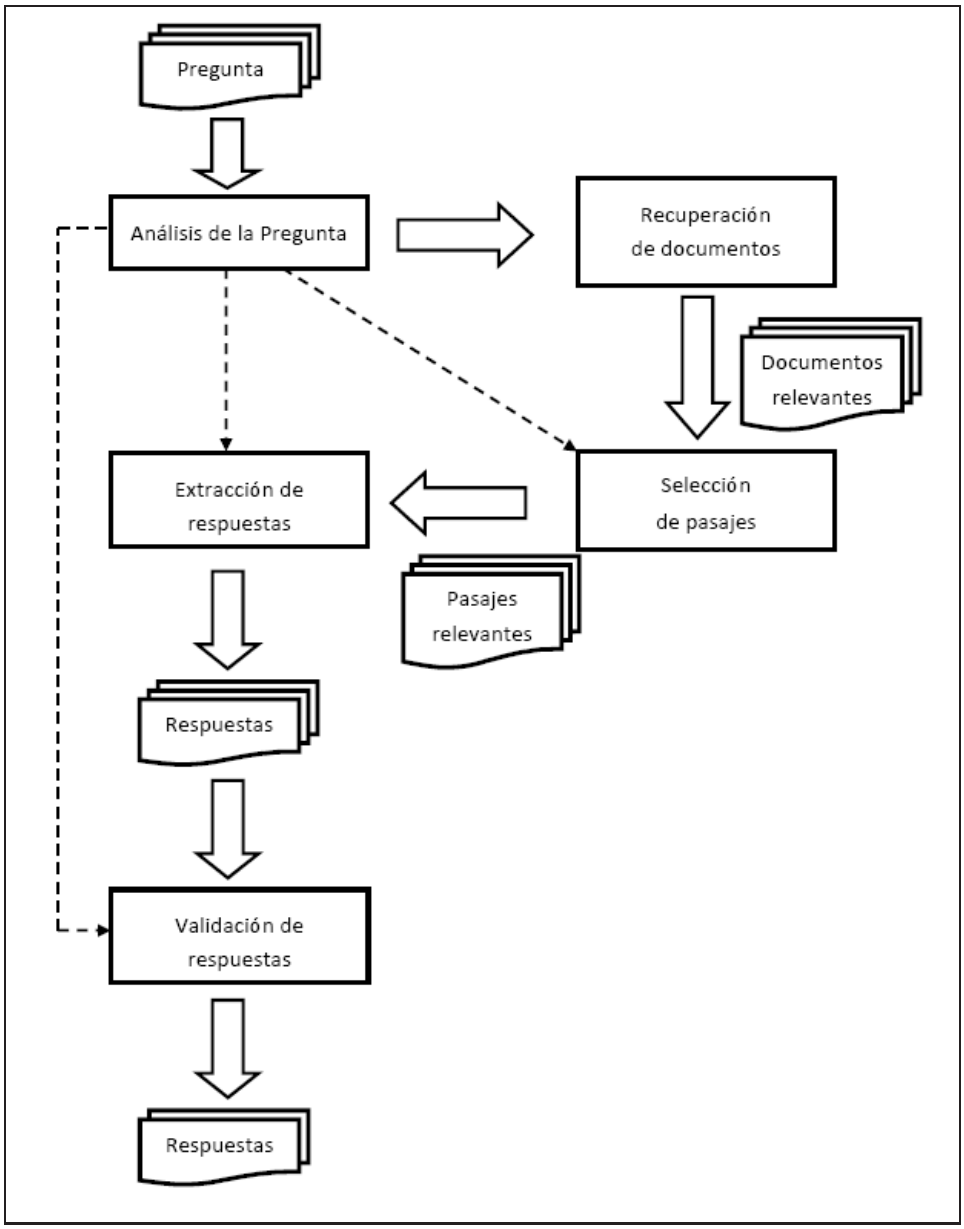

Figura 2.6: Arquitectura de un sistema de QA donde se aplica un módulo de Validación de Respuestas a las respuestas candidatas

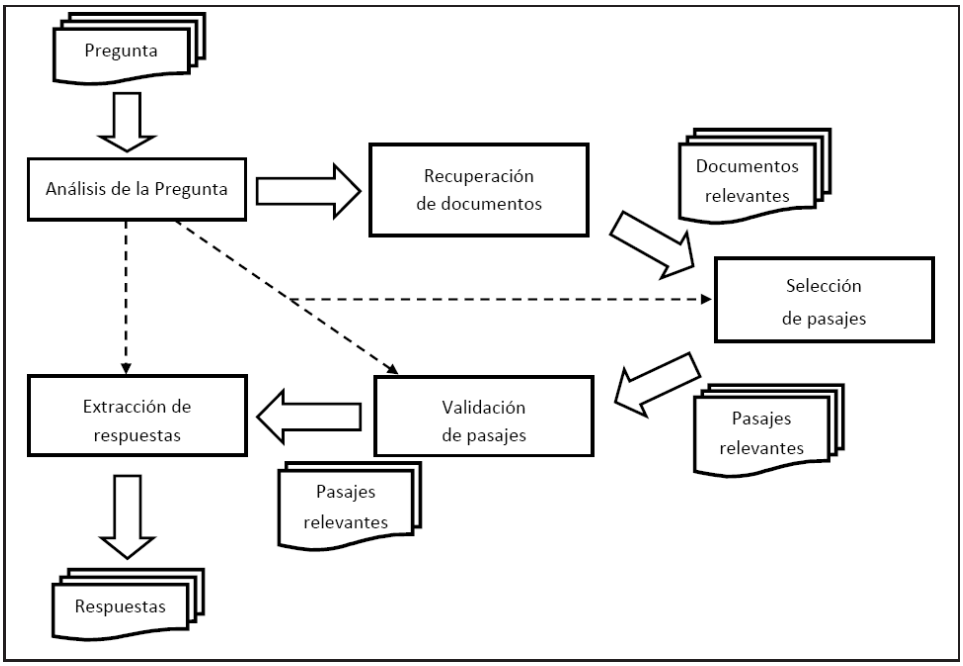

Figura 2.7: Arquitectura de un sistema de QA donde se aplica un módulo de Validación de Respuestas para filtrar pasajes relevantes

Esta idea de combinación entre sistemas ha sido utilizada en otras tareas de lenguaje natural como etiquetado morfosintáctico (Brill and Wu, 1998), desambiguación del sentido de las palabras (Pedersen, 2000) y análisis sintáctico (Henderson and Brill, 1999). Además, el concepto de sistemas multi-flujo de QA es similar al de la combinación de clasificadores en aprendizaje automático (Dietterich, 1997).

Sin embargo, en los sistemas multi-flujo de QA surge el problema de qué criterio seguir para seleccionar la respuesta final de entre todas las aportadas por los distintos sistemas individuales de QA (Jijkoun and de Rijke, 2004). Una posibilidad la representa el uso de un sistema de AV para realizar dicha selección de forma semejante a cómo se utiliza para realizar la selección en un sistema individual de QA. De este modo se tendría un sistema como el mostrado en la Figura 1.2 (página 28), donde la selección sería llevada a cabo por un módulo de AV.

Por último, también se puede pensar en utilizar dentro de un sistema de QA un módulo de AV que realice tanto validación como selección. Es decir, un módulo que filtre las respuestas y de entre las que considere correctas seleccione una. Evidentemente este procedimiento se puede llevar a cabo por un único sistema de AV que realice las dos funciones o por dos sistemas de AV, uno de los cuáles ejecutaría primero la validación mientras que el otro realizaría a continuación la selección.

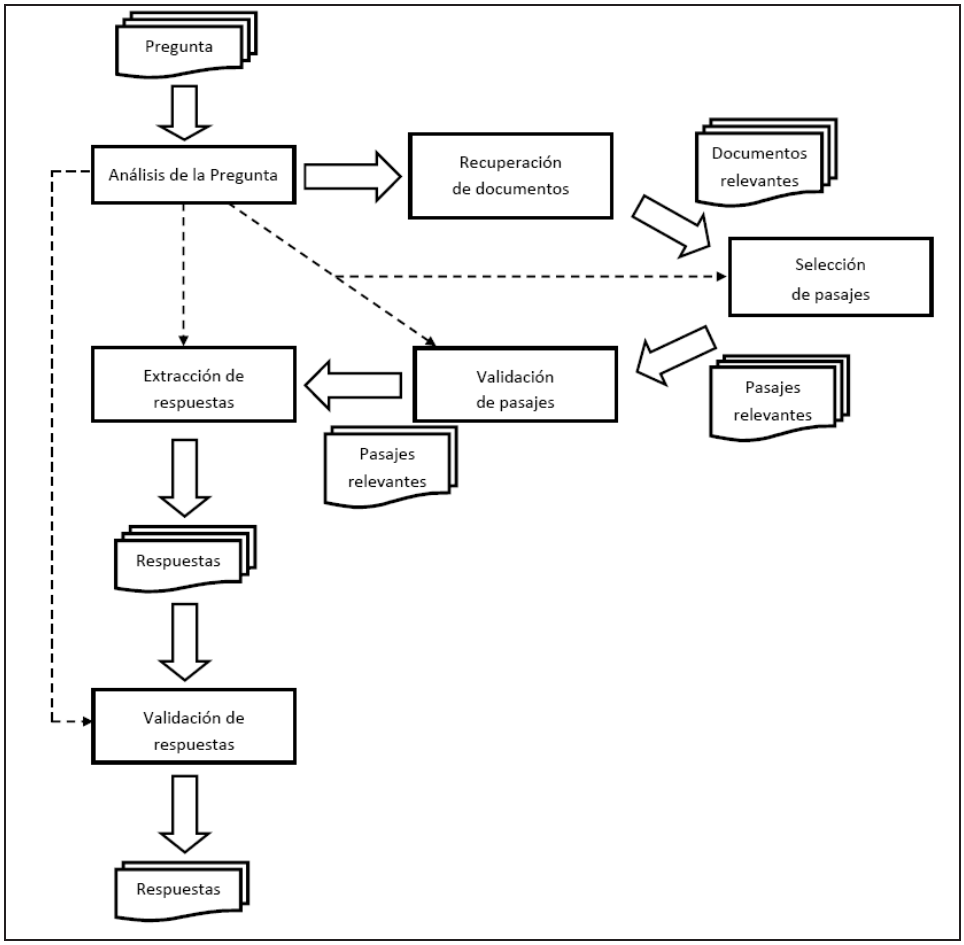

Figura 2.8: Arquitectura de un sistema de QA donde se incorpora un módulo de Validación de Respuestas para validar tanto párrafos relevantes como respuestas candidatas

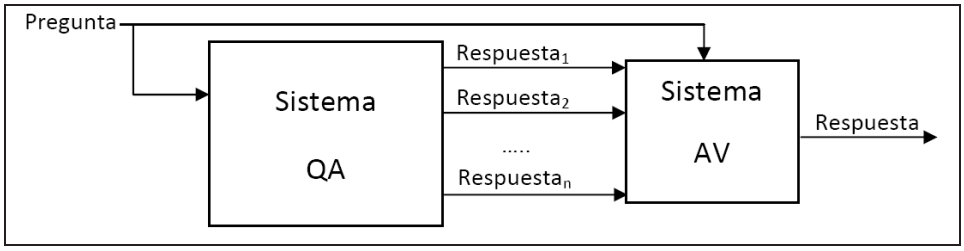

Figura 2.9: Arquitectura de un sistema de QA donde se incorpora un módulo de Validación de Respuestas para realizar la selección de la respuesta final de entre un conjunto de respuestas candidatas

2.4.2. Características de la Evaluación

En esta sección se repasan las evaluaciones de sistemas de AV que se han realizado antes de la propuesta de este trabajo. Estas evaluaciones fueron llevadas a cabo por las mismas personas que realizaban la propuesta de un sistema de AV. Es decir, en el mismo trabajo se realizaba la propuesta de un sistema de AV y se llevaba a cabo una pequeña evaluación del mismo. De este modo, la evaluación estaba enfocada a evaluar las características del sistema propuesto sin realizar comparaciones con otros enfoques.

Una de las primeras evaluaciones de este tipo de sistemas fue la realizada por Breck et al. (2000) en inglés. Dado que el objetivo de su trabajo era desarrollar métodos automáticos de evaluación de sistemas de QA (cómo se mostrará en la sección 2.4.3 de la página 64), la evaluación se centró en medir la correlación entre los resultados de evaluación obtenidos utilizando juicios humanos y los resultados obtenidos utilizando su sistema. Además, los autores midieron también el porcentaje de respuestas para las cuáles el juicio emitido por su sistema coincidía con el de los evaluadores humanos. Es decir, midieron el *accuracy* de su sistema clasificando las respuestas en correctas e incorrectas. Para realizar la evaluación, los autores hicieron uso de las preguntas (un total de 198), respuestas (un total de 37927) y juicios humanos del TREC-8 (Voorhees and Tice, 1999), creando de forma manual para cada pregunta una respuesta correcta para así poder utilizar su sistema (ya que su método se basa en comparar esta respuesta con la que se tiene que validar).

Otra evaluación a tener en cuenta es la realizada por Magnini et al. (2002a) en inglés. Para realizar dicha evaluación se construyó una colección de 492 preguntas obtenidas de la tarea de QA del TREC 2001, tomando para cada pregunta tres respuestas correctas y tres respuestas incorrectas a partir de los runs enviados por los participantes. En total se obtuvieron 2726 pares pregunta-respuesta[17].

Para medir el rendimiento de su sistema sobre la colección creada, los autores hicieron uso de tres medidas de evaluación distintas que no fueron combinadas entre si:

- **Precisión:** proporción de pares pregunta-respuesta considerados como correctos y que en realidad eran correctos.
- **Cobertura:** proporción de pares pregunta-respuesta correctos de la colección que se lograron detectar.
- **Tasa de éxito:** los autores denominaros así a la proporción de pares pregunta-respuesta para los cuáles un sistema daba el mismo juicio que los evaluadores humanos (correcto o incorrecto). En realidad esta medida es otro nombre para el *accuracy* tradicional.

Además de los resultados de sus sistemas, los autores aportaron los resultados de un sistema “baseline” que hacía uso de los documentos suministrados por la

[17] Para algunas preguntas no se pudo disponer de tres respuestas correctas

organización del TREC para cada pregunta. Estos documentos son un total de 1000 relevantes por cada pregunta que proporcionaba la organización del TREC para facilitar la participación de aquellos grupos que no podían hacer uso de un sistema de IR. Para crear el sistema "baseline", una respuesta se consideraba como correcta en caso de que estuviese contenida en alguno de los 10 primeros documentos del ranking de los 1000 documentos para esa pregunta. En caso contrario la respuesta se consideraba incorrecta.

Esta evaluación se realizó solamente sobre los sistemas desarrollados por los autores (descritos en la sección 2.4.3 de la página 64) y no se evaluó la posible mejora de resultados que podría conseguir un sistema de QA que utilizase un módulo de AV. Además, la colección creada no fue utilizada por otros investigadores y no se tiene conocimiento de que esté disponible.

Otra evaluación de este tipo de sistemas es la realizada por Tonoike et al. (2004) para poner a prueba su sistema en japonés. Esta evaluación se realizó sobre el conjunto de preguntas y respuestas del juego de mesa *¿Quién quiere ser millonario?*. Este juego consiste en una serie de preguntas para cada una de las cuales hay cuatro posibles respuestas y solo una es correcta. Tras realizar un análisis sobre las preguntas, los autores concluyeron que estas eran similares (tanto en tamaño como en naturaleza) a las de evaluaciones internacionales de QA, en concreto aquellas del NTCIR4 QAC2 (Fukumoto et al., 2004b) y TREC 2003 (Voorhees, 2003). En total, se utilizaron 906 preguntas para realizar la evaluación.

Los autores utilizaron como medidas de evaluación *precisión* y *cobertura*, mientras que como "baselines" se emplearon versiones simplificadas de las propuestas realizadas por los autores (los enfoques propuestos por los autores se muestran en la sección 2.4.3 de la página 64).

En resumen, las evaluaciones de sistemas de AV previas a la propuesta de este trabajo fueron realizadas para medir el rendimiento de sistemas concretos y no se realizaron comparaciones con otros sistemas. Además, estas evaluaciones se centraron en medir la efectividad de la validación realizada, pero en ningún caso en estudiar el impacto que puede suponer el utilizar módulos de AV dentro de un sistema de QA, ni en evaluar sistemas de AV que realizan selección de respuestas.

2.4.3. Algunos Sistemas Previos a esta Propuesta

En esta sección se revisan las aproximaciones realizadas por distintos sistemas de AV desarrollados antes de la propuesta de este trabajo. Las aproximaciones estudiadas se pueden agrupar principalmente en dos enfoques: los métodos basados en redundancias que tratan de aprovechar entre otras cosas la gran cantidad de información que hay disponible en la Web; y los métodos basados en un profundo análisis de la pregunta y las respuestas. Se realiza esta revisión puesto que es relevante para la propuesta realizada en este trabajo.

2.4.3.1. Métodos basados en Redundancia

Uno de los primeros trabajos realizados en AV fue el descrito en Breck et al. (2000), cuyo objetivo era hacer más rápida la evaluación de sistemas de QA. Este método calculaba la proporción de términos de respuestas obtenidas por expertos humanos que aparecían en las respuestas generadas por el sistema de QA a evaluar. Para que dicho cálculo no se viera afectado por formulaciones distintas entre los expertos humanos y los sistemas a evaluar, se eliminaban palabras vacías y se utilizaban las raíces de las palabras (más conocidas por su nombre en inglés, *stems*). Sin embargo, este método no tiene en cuenta a la pregunta y presenta como principal inconveniente que el proceso de validación se reduce a preguntas para las cuáles ya se tienen generadas respuestas correctas, por lo que utilizarlo sobre nuevas preguntas tiene asociado el coste de generar a mano respuestas para las nuevas preguntas.

Otros métodos basados en redundancia aplican un simple esquema de voto según el cuál se considera que la respuesta que más veces aparece entre las candidatas es la que tiene mayor probabilidad de ser correcta (Brill et al., 2001). Este enfoque presenta varios inconvenientes, como que por ejemplo no siempre se cumple que la respuesta más frecuente es la correcta. Además, las respuestas cortas suelen verse favorecidas en estos casos. Por ejemplo, la respuesta *1944* tendería a tener un mayor número de apariciones que *6 de junio de 1944* para responder a la pregunta *¿Qué día se realizó el Desembarco de Normandía?*, ya que una está contenida en la otra y eso contaría como una repetición[18].

El resto de métodos se basan principalmente en aprovechar la información contenida en la Web asumiendo que las respuestas a un gran número de preguntas se encuentran disponibles en varias fuentes. Estos métodos confían en descubrir lo relacionadas que están las respuestas con las preguntas midiendo la frecuencia con la que ambas aparecen juntas en diversas fuentes. La hipótesis en la que se basan estos métodos es que dicho número de apariciones se puede considerar como una característica importante para tomar la decisión sobre si validar o no una respuesta.

Estos métodos asumen que para realizar con garantías el proceso de validación es necesario poder contar con una base de conocimiento que sea lo suficientemente grande como para contener una gran parte del conocimiento humano. También es deseable que dicha base de conocimiento contenga la suficiente redundancia como para poder contener diferentes formulaciones de los hechos que la conforman, lo que hace más fácil realizar consultas sobre ella. Otra característica deseable es que dicha base de conocimiento pueda cambiar dinámicamente permitiendo reflejar en cada momento el estado actual del conocimiento humano. Teniendo en cuenta que el desarrollo de bases de conocimiento de gran tamaño es muy costoso tanto en tiempo como en recursos, es fácil darse cuenta de la dimensión del problema.

[18] En este ejemplo se podría desechar la respuesta *1944* al comprobar que no es un día, ya que se está preguntando por un día. Sin embargo, dejando a un lado este tratamiento de tipos (que no era llevado a cabo por Brill et al. (2001)), este ejemplo sirve para mostrar algunos de los defectos de este método.

Todos estos motivos son los que conducen al uso de la Web como base de conocimiento en estos enfoques. Sin embargo, hay que tener en cuenta que en la Web el conocimiento está almacenado en texto no estructurado y que además el acceso a los motores de búsqueda puede ser a veces demasiado lento si, por ejemplo, se quiere realizar una validación rápida (como podría ser el caso de un sistema de QA en tiempo real).

Una vez se asume a la Web como base de conocimiento, el modo de funcionar de estos métodos consiste en extraer primero una serie de palabras clave tanto de las preguntas como de las respuestas. Para obtener estas palabras clave se eliminan las palabras vacías, añadiendo en algunos casos variaciones morfológicas de las palabras o sinónimos extraídos de WordNet (Magnini et al., 2002a). Una vez se tienen las palabras clave, se utiliza el número de documentos en la Web que contienen a estas palabras clave como una medida de asociación entre las preguntas y las respuestas, tomándose la decisión de validación en función del número de documentos recuperados (Tonoike et al., 2004).

Otros métodos, sin embargo, combinan las palabras clave de las preguntas y las respuestas para crear patrones de validación que se lanzan a un motor de búsqueda. Por ejemplo, dada la pregunta *¿Cuál es la capital de Francia?* y la respuesta *París*, un posible patrón sería *[París <texto> capital <texto> Francia]*. La decisión final sobre si validar o no la respuesta se toma considerando el número de documentos recuperados con dicho patrón (Magnini et al., 2002b).

Por otro lado, la mayoría de trabajos sobre selección de respuestas en sistemas multi-flujo han utilizado esquemas de voto que seleccionan la respuesta con mayor número de apariciones (Brill et al., 2001; Burger et al., 2002), incluyendo en algunos casos información acerca del rendimiento individual que obtuvo cada sistema de QA en el pasado (Jijkoun and de Rijke, 2004). Otras propuestas han basado su decisión en el valor de confianza que otorga cada sistema individual de QA a su respuesta, realizando una ponderación entre los valores otorgados por los distintos sistemas (Chu-Carroll et al., 2003).

2.4.3.2. Métodos basados en Análisis Textual

El otro enfoque utilizado para realizar la tarea de AV considera que es necesario llevar a cabo un mayor análisis de las relaciones semánticas existentes entre respuesta y pregunta basándose, si es necesario, en conocimiento semántico.

Antes de la propuesta de validación que se realiza en este trabajo se desarrollaron métodos en esta línea principalmente por parte del Language Computer Corporation[19] (LCC), para incorporarlos a sus sistemas de QA. Estos sistemas estaban enfocados a la tarea de QA del TREC en la cuál había que devolver fragmentos de texto que contuviesen la respuesta correcta.

Uno de estos métodos consiste en validar una respuesta si se puede encontrar una explicación a dicha respuesta. Para realizar este proceso se pueden utilizar

[19] http://www.languagecomputer.com/

patrones léxico-sintácticos o interpretaciones de aposiciones. A este procedimiento le llamaron abducción de respuestas (Harabagiu and Maiorano, 1999).

Otro método en esta línea consiste en transformar primero a forma lógica tanto la pregunta como el fragmento de texto que constituye la respuesta, para a continuación realizar el proceso de validación haciendo uso de un demostrador lógico que contiene conocimiento del mundo, extraído de WordNet, en forma de axiomas (Moldovan et al., 2002). El principal problema que presenta este enfoque es su coste computacional, ya que el proceso de demostración lógica es lento.

2.5. Evaluación de Implicación Textual

La tarea de Reconocimiento de Implicación Textual (en inglés Recognising Textual Entailment, RTE) consiste en decidir dados dos textos en lenguaje natural si el significado de un texto implica al significado del otro texto (llamado hipótesis) (Dagan et al., 2005). Dicho de otro modo, se trata de averiguar si el significado de la hipótesis se puede obtener a partir del significado del texto. Sean por ejemplo los dos textos siguientes:

1. De acuerdo con la Enciclopedia Británica, Indonesia es la mayor nación-archipiélago del mundo, con 13.670 islas.

2. Indonesia está formada por 13.670 islas.

En dicho ejemplo es evidente que el significado del segundo texto se puede deducir a partir del primero. En este caso se dice que el primer texto implica al segundo.

Dado que un fenómeno fundamental del lenguaje natural es la variabilidad de las expresiones semánticas, donde el mismo significado se puede expresar de diversas formas, la tarea de RTE representa un intento de unificar los esfuerzos para capturar el mayor número de inferencias semánticas necesarias por distintas aplicaciones. El deseo es que la investigación en este campo lleve al desarrollo de motores de implicación que puedan ser usados como un módulo independiente en muchas aplicaciones de forma similar al uso actual de los analizadores o tokenizadores, de tal modo que la noción de implicación textual sea independiente del tipo de tarea al que se aplique.

Son muchas las aplicaciones que se beneficiarían de tener un módulo de RTE. A continuación se muestran algunos ejemplos de estas tareas:

- **Sistemas de QA:** la respuesta dada por un sistema de QA tiene que ser implicada por el texto que soporta la veracidad de la respuesta. Los sistemas de RTE podrían ayudar a validar las respuestas candidatas para que se pueda realizar una mejor selección de la respuesta final. De hecho, algunos trabajos recientes han mostrado cómo el uso de sistemas de RTE puede mejorar el rendimiento de sistemas de QA(Harabagiu and Hickl, 2006; Rodrigo et al., 2009).

- **Resumen automático (en inglés Automatic Summarization, SUM):** un fragmento que contiene información redundante dentro de un resumen es candidato a ser eliminado si hay otro fragmento que lo implica. Por tanto, los sistemas de RTE servirían para eliminar los fragmentos redundantes de un resumen.

- **Sistemas de IR:** en esta tarea los documentos recuperados deberían implicar a la consulta que se ha lanzado para obtener dichos documentos. Los sistemas de RTE ayudarían a los de IR a verificar que los documentos recuperados son relevantes para la consulta que se lanzó.

- **Sistemas de IE:** las relaciones semánticas entre palabras, como por ejemplo las relaciones de meronimia, que se extraen de un determinado documento deben de ser implicadas por dicho documento. Los sistemas de RTE servirían para comprobar que las relaciones extraídas son efectivamente correctas.

2.5.1. Características de la Evaluación

Las colecciones de evaluación que se utilizan en RTE están formadas generalmente por un conjunto de pares *Texto-Hipótesis (T-H)* donde para cada par hay que indicar si el texto implica o no a la hipótesis. El *Texto* está formado generalmente por una o dos frases mientras que la *Hipótesis* consta de una frase corta. Un ejemplo de un par T-H (obtenido a partir de la colección de desarrollo del RTE-3 Challenge) se puede ver en la Figura 2.10 (página 68).

Texto: `Between March and June, scientific observers say, up to 300,000 seals are killed. In Canada, seal-hunting means jobs, but opponents say it is vicious and endangers the species, also threatened by global warming.`

Hipótesis: `Hunting endangers seal species.`

Figura 2.10: Par Texto-Hipótesis de ejemplo.

Las colecciones de evaluación de sistemas de RTE se suelen crear de forma manual, donde un conjunto de expertos humanos toma la decisión acerca de qué valor otorgar a cada par (si hay o no implicación). En caso de que haya discrepancias entre los expertos, una solución que se puede adoptar es tomar el valor más votado. Sin embargo, hay casos en los cuáles se ha decidido incorporar a las colecciones de evaluación solamente aquellos pares sobre los cuáles están de acuerdo todos los evaluadores con el fin de evitar cualquier tipo de desacuerdo. Esta es la solución adoptada en los RTE Challenges.

También hay casos en los cuáles se han creado colecciones de evaluación de forma semiautomática (un ejemplo se propone en el Capítulo 3 de la página 89 de

este trabajo) o de forma totalmente automática. Un ejemplo de colección generada automáticamente lo representa la colección desarrollada por MITRE durante el primer RTE Challenge, la cuál se describe más ampliamente en el apartado 3.2 (página 93).

La fuente a partir de la cuál se generan las colecciones puede variar en función de los objetivos de la evaluación. Si se desea tratar de cubrir el mayor número posible de escenarios (como suele pasar en los RTE Challenges que se describen en el apartado 2.5.2 de la página 70), son varias las fuentes de datos que se utilizan. Sin embargo, si solo se desea evaluar a los sistemas de RTE sobre un determinado escenario, entonces se restringe la fuente de donde se obtiene la colección. Un ejemplo de este último caso se desarrolla en el Capítulo 5 de este trabajo (página 125), donde se crean colecciones enfocadas a evaluar sistemas de Validación de Respuestas basados en RTE.

Una vez se tienen las colecciones, la evaluación de un sistema se realiza comparando la salida del sistema para cada par de la colección, con el valor otorgado por los expertos humanos. Generalmente se utilizan dos medidas de evaluación: *accuracy* y *confidence weigted score* (o su variante *average precision*). A continuación se describen estas medidas.

2.5.1.1. Accuracy

Una forma de realizar la evaluación de sistemas de RTE consiste en considerar a dichos sistemas como clasificadores, de modo que se evalúa su capacidad para decidir si hay o no implicación. En este caso la principal medida de evaluación es *accuracy* (Fórmula (2.23)), la cuál representa la proporción de predicciones realizadas correctamente.

$$CWS = \frac{\#pares\ correctamente\ clasificados}{\#pares\ colección} \tag{2.23}$$

En caso de tener colecciones de evaluación que estén balanceados en términos de pares positivos y negativos, un sistema que indicase que hay implicación en todos los pares (o un sistema que considerase que no hay implicación en ningún par) lograría un valor de *accuracy* del 50 %, lo cuál representa un “baseline” natural.

2.5.1.2. Confidence Weigted Score

Cuando se desea evaluar la capacidad de los sistemas otorgando valores de confianza a sus predicciones, una de las medidas de evaluación que se utilizan en RTE es *Confidence Weigthed Score* (CWS) (Fórmula (2.24)).

$$CWS = \frac{1}{n}\sum_{i=1}^{n}\frac{\#predicciones - correctas - hasta - ranking - i}{i} \tag{2.24}$$

Para poder hacer uso de esta medida, el sistema a evaluar ha de ordenar su salida en forma de ranking. Este ranking está formado de tal manera que el sistema coloca en las primeras posiciones los pares T-H para los cuáles está más seguro de la decisión tomada, y en las últimas posiciones los pares T-H para los cuáles la confianza en su respuesta es menor.

CWS sirve en RTE para recompensar a los sistemas que otorgan valores altos de confianza cuando sus predicciones son correctas y valores bajos de confianza cuando sus predicciones son erróneas. La diferencia de su uso respecto a la evaluación en QA (que se vio en la Sección 2.3 de la página 47) radica en que en QA se premia solo la detección de respuestas correctas, mientras que en RTE se consideran clasificaciones correctas, es decir, que se detecte correctamente si en un par hay implicación o no.

2.5.1.3. Average Precision

Una alternativa al uso de *CWS* para evaluar el ranking otorgado por los sistemas de RTE es el uso de la precisión media (en inglés *Average Precision*, AP). Como ya se mencionó en la sección 2.3.3.5 (página 53), *CWS* está inspirada en *AP*. Sin embargo, *AP* evalúa la habilidad de los sistemas para ordenar los pares T-H de acuerdo a la confianza que se tiene en la implicación de los mismos, en lugar de en la clasificación realizada (lo que se evalúa con *CWS*). Es decir, *AP* evalúa la habilidad de situar primero aquellos pares en los que hay implicación y en último lugar los pares en los que no hay implicación. Por este motivo, la formulación de *AP* es distinta de la de *CWS*.

$$AP = \frac{1}{R}\sum_{i=1}^{n}\frac{E(i) * \#predicciones - correctas - hasta - ranking - i}{i} \tag{2.25}$$

La formulación de AP se muestra en la Fórmula (2.25), donde se puede observar que la principal diferencia con CWS (Fórmula (2.24) de la página 69) es la inclusión de la función E(*i*). E(*i*) vale 1 si el par *i* está marcado en la colección de test como un ejemplo positivo de implicación, mientras que E(*i*) vale 0 en caso contrario. Además, el valor final sobre todos los pares se divide entre *R*, el cuál representa el número total de pares con implicación que hay en la colección de evaluación. Estas modificaciones respecto a *CWS* tienen como consecuencia principal que sólo se añade valor al resultado final en las posiciones del ranking donde se encuentra un par con implicación, mientras que en *CWS* se añade valor en todas las posiciones del ranking.

2.5.2. Recognising Textual Entailment (RTE) Challenges

A pesar del interés que tiene el campo de la implicación textual, no ha sido hasta fechas recientes cuando se han empezado a desarrollar foros de evaluación

Cuadro 2.2: Número de pares en cada edición del RTE Challenge.

Año	Cjto. Desarrollo	Cjto. Test
2005	567	800
2006	800	800
2007	799	800
2008	-	1000
2009	600	500

en los que poder comparar distintos enfoques bajo las mismas condiciones. El foro de evaluación de sistemas de RTE que más éxito ha tenido ha sido el RTE Challenge, cuya primera edición se celebró en el año 2005 con el objetivo de proporcionar una primera oportunidad para presentar y comparar diferentes aproximaciones para modelar y reconocer la implicación textual (Dagan et al., 2005). Las siguientes ediciones de este foro se han venido realizando anualmente, siendo la última edición hasta el momento la de 2009 (aunque ya está planificada una nueva edición para finales de 2010). La organización de este foro estuvo en primer lugar a cargo de la red de excelencia PASCAL (Pattern Analysis, Statistical Modelling and Computational Learning)[20], pasando a estar a cargo del National Institute of Standards and Technology (NIST)[21] en su cuarta edición (la de 2008) como una tarea dentro del TAC.

A la primera edición se presentaron 16 sistemas de grupos de todo el mundo y fue tal el éxito de la tarea, que se celebraron nuevas ediciones en 2006 (Bar-Haim et al., 2006), con 23 participantes, y en 2007 y 2008 con 26 participantes en cada una de las dos ediciones (Giampiccolo et al., 2007, 2008).

Para realizar la evaluación de los sistemas participantes, en cada edición se crea un conjunto de pares T-H que se dividen en un conjunto de desarrollo (para permitir que los participantes pongan a punto sus sistemas) y un conjunto de test sobre el cuál se evalúa de forma comparativa el rendimiento de los sistemas. En el Cuadro 2.2 (página 71) se puede ver el número de pares disponibles en cada edición para el desarrollo y la evaluación de los sistemas participantes. En dicho Cuadro se puede observar como en todas las ediciones se proporcionó una colección de desarrollo excepto en la de 2008.

En general, cada colección está subdividida en 4 subconjuntos (7 subconjuntos en la edición 2005 y 3 en la de 2009), cada uno de los cuáles se corresponde con ejemplos extraídos de diversas aplicaciones de Procesamiento del Lenguaje Natural. Dentro de cada subconjunto se trata de balancear el número de ejemplos positivos y negativos. El objetivo principal de crear distintos subconjuntos consiste en evaluar el comportamiento de los sistemas de RTE sobre distintos fenómenos que pueden aparecer en distintas tareas típicas del Lenguaje Natural. En algunos

[20] http://pascallin.ecs.soton.ac.uk/

[21] http://www.nist.gov/

casos, los ejemplos se toman de recursos externos, como conjuntos de datos disponibles públicamente o la salida de sistemas reales, y en otros casos se obtienen directamente a partir de la Web. Los distintos tipos de aplicaciones que se tienen en cuenta y la forma en la cuál se obtienen los pares se muestran a continuación (entre paréntesis se indica el acrónimo que se utiliza para cada tipo de tarea):

- **Recuperación de Información (IR):** las hipótesis se generan a partir de consultas a sistemas de IR. Dichas consultas expresan determinadas relaciones semánticas, mientras que los textos se obtienen a partir de los documentos recuperados por un motor de búsqueda para dichas consultas.

- **Búsqueda de Respuestas (QA):** estos pares se crean haciendo uso de un sistema real de QA al cual se le lanzan varias preguntas, recogiéndose tanto las respuestas que genera el sistema como los textos que devuelve para soportar la veracidad de sus respuestas. Como texto del par se toma un fragmento de uno de los textos soporte, mientras que para crear la hipótesis se pasa a forma afirmativa la pregunta y a continuación se inserta en ella la respuesta dada por el sistema. De este modo, si la respuesta es correcta y soportada por el texto, el texto debe de implicar a la hipótesis.

- **Extracción de Información (IE):** los anotadores eligen como textos de los pares frases que son candidatas a contener relaciones semánticas comúnmente utilizadas en tareas de IE. Como hipótesis se eligen formulaciones sencillas de las relaciones.

- **Resumen Automático (SUM):** para crear los pares de este tipo, los anotadores reciben clusters de documentos sobre un mismo tema así como el resumen obtenido por un sistema automático para cada uno de estos clusters. A continuación los anotadores escogen frases que tengan entre si un gran solapamiento léxico. Para crear un par positivo se simplifica la hipótesis eliminando partes de una frase hasta que ésta sea totalmente implicada por el texto seleccionado para el par. Los ejemplos negativos se crean simplificando las hipótesis del mismo modo pero sin que al final haya una relación de implicación.

En cuanto a los objetivos específicos de la evaluación de cada campaña, en este trabajo se ha decidido establecer una división agrupando distintas ediciones de la tarea para poder explicar de forma más clara los objetivos y los retos planteados en la evaluación.

2.5.2.1. Los Tres Primeros RTE Challenges

El planteamiento de la tarea fue bastante homogéneo durante las tres primeras ediciones del RTE Challenge. El objetivo principal era ofrecer un foro para evaluar los distintos enfoques para realizar RTE y conseguir que la comunidad interesada en dicha tarea fuese creciendo. Es por ello que apenas se introdujeron

modificaciones reseñables entre las distintas ediciones. Este hecho hizo que durante las primeras ediciones las colecciones de evaluación estuviesen formadas por textos generalmente cortos que no contenían demasiada información de correferencia, con el fin de que los participantes se centrasen en mayor medida en investigar métodos para realizar el proceso de inferencia.

2.5.2.2. RTE Challenges 4 y 5

Las siguientes ediciones de los RTE Challenges supusieron un paso adelante en cuanto a la evaluación de sistemas de RTE al plantear la tarea de manera más realista. La innovación más notable fue la posibilidad de que el juicio de cada par pudiese tener 3 posibilidades. En concreto, en los casos donde no había implicación había que diferenciar las situaciones en las cuáles el contenido de la hipótesis contradecía el contenido del texto, de los casos en los cuáles no había suficiente información para decidir que el texto implicaba a la hipótesis (pero tampoco se podía inferir que el texto contradijese a la hipótesis).

La inclusión de esta mayor distinción en los pares sin implicación no influyó en la cantidad de pares con implicación de cada colección, la cuál se siguió manteniendo en el 50 % del total. La influencia tuvo lugar en la distribución de los pares sin implicación, una parte de los cuáles fue anotada como contradicción (utilizando el juicio *CONTRADICTION*), mientras que el resto de pares fueron anotados como que la relación de implicación era desconocida (utilizando el juicio *UNKNOWN*). En el caso particular del RTE-4, la proporción de pares con contradicción fue del 15 % del total, mientras que la de pares con implicación desconocida fue del 35 % del total.

Dentro del contexto de la evaluación se mantiene como medida principal de evaluación a *accuracy*, pero en este caso sobre tres juicios distintos. En caso de querer realizar una evaluación con solo dos juicios (implicación y no implicación), los pares con contradicción y relación desconocida se agrupan en un solo conjunto (pares con no implicación). Por otro lado, para realizar la evaluación de los sistemas que devuelven sus respuestas en una lista ordenada, también se agrupan los pares con contradicción y relación desconocida de implicación en un sólo conjunto debido a que la precisión media (descrita en la sección 2.5.1 de la página 68) sólo puede ser aplicada sobre una anotación binaria.

Otra de las modificaciones introducidas en estas dos ediciones con el fin de hacer más realista la tarea consistió en aumentar la longitud de los textos de los pares, manteniendo la longitud de las hipótesis. El objetivo principal de esta modificación es realizar un acercamiento a la evaluación de casos reales en los cuáles se requiere realizar análisis del discurso.

2.5.2.3. Tarea Piloto del RTE-5

A pesar de que en la edición de 2009 se eliminaron los pares procedentes de sistemas de resumen automático, este tipo de sistemas se tuvieron en cuenta a la

hora de definir una tarea piloto en dicha edición. La tarea piloto propuesta dentro del RTE-5 consiste en encontrar todas las oraciones de un conjunto de documentos que impliquen a una hipótesis dada. La relación con la tarea de resumen automático viene dada por el hecho de que la hipótesis de la que se parte se toma a partir de un conjunto de resúmenes creados a mano a partir de los documentos donde hay que buscar oraciones. Los objetivos principales de esta tarea piloto eran:

- Crear una colección de evaluación que refleje la distribución natural de la implicación textual en una colección de documentos.
- Estudiar los problemas que pueden surgir a la hora de aplicar sistemas de RTE en entornos reales.
- Analizar el impacto potencial de los sistemas de RTE en una aplicación con datos reales, en este caso resumen automático. En cierto sentido este objetivo es similar al de la propuesta realizada en este trabajo (Capítulo 5 de la página 125), la cuál está centrada en los sistemas de QA.

Las características planteadas en esta tarea piloto hacen que la evaluación sea distinta a la realizada anteriormente en RTE. En este escenario se desea evaluar tanto la capacidad de un sistema para encontrar oraciones que implican a una hipótesis dada, como la efectividad con la cuál se realiza esta detección. Es por ello que para realizar la evaluación se mide la *precisión* (proporción de oraciones recuperadas que efectivamente implican a la hipótesis dada, como muestra la Fórmula (2.26)) y la *cobertura* (proporción de oraciones que implican a la hipótesis que han sido detectadas, como muestra la Fórmula (2.27)). Estas dos medidas se combinan en una sola haciendo uso de la media armónica *F* (Fórmula (2.11) de la página 44). Además, se calcula tanto una macro-media[22] (más conocida por su nombre en inglés, *macro-average*), como una micro-media[23] (más conocida por su nombre en inglés, *micro-average*).

$$precisión = \frac{\#frases\_recuperadas\_que\_implican\_a\_H}{\#frases\_recuperadas} \tag{2.26}$$

$$cobertura = \frac{\#frases\_recuperadas\_que\_implican\_a\_H}{\#total\_frases\_que\_implican\_a\_H} \tag{2.27}$$

2.5.3. Algunos Sistemas Existentes de RTE

En esta sección se pretende realizar una visión general de los enfoques desarrollados para la tarea de RTE. El motivo por el cuál se realiza en este trabajo una

[22] Se calcula *F* por cada documento y luego se calcula la media de todas las *F*

[23] Se calcula la *precisión* y la *cobertura* teniendo en cuenta todos los documentos y posteriormente se calcula *F* sobre dichos valores de *precisión* y *cobertura*

visión para esta tarea y no para otras como QA reside en que este tipo de sistemas es el que más se asemeja a los de Validación de Respuestas, los cuáles van a ser tratados a partir del Capítulo 3 (página 89). Esta visión se basa principalmente en lo observado a lo largo de las evaluaciones del RTE Challenge, que es donde han sido desarrollados la mayor parte de los sistemas actuales.

El principal criterio utilizado por los sistemas para tomar la decisión de implicación ha sido la similitud entre texto e hipótesis y la cobertura de la hipótesis por parte del texto (en métodos léxicos y sintácticos). Por otro lado, ha habido sistemas que han explorado la posibilidad de inferir la hipótesis a partir del texto (en el enfoque lógico).

El tipo más básico de inferencia utilizado mide el grado de solapamiento entre las palabras del texto y las de la hipótesis incluyendo en algunos casos un procesamiento más complejo como stemming, lematización, etiquetado morfosintáctico o midiendo el solapamiento de secuencias de palabras entre texto e hipótesis. Los tratamientos a un mayor nivel de inferencia léxica tienen también en cuenta relaciones entre palabras que pueden reflejar implicación y que se pueden obtener a partir de WordNet[24] o recursos adquiridos automáticamente.

En el siguiente nivel se establecen los sistemas que miden el grado de solapamiento entre las estructuras sintácticas del texto y la hipótesis basándose en algún criterio de distancia, como por ejemplo algoritmos de distancia de edición de árboles. A lo largo de las distintas evaluaciones del RTE Challenge se ha observado un incremento hacia la consolidación general de estos métodos basados en la estructura sintáctica del texto y la hipótesis.

En un nivel superior está el modelo semántico, en el cuál las representaciones de los pares T-H son árboles de constituyentes etiquetados con roles semánticos. A la hora de decidir si hay implicación en este modelo se consideran, en general, las coincidencias entre los conjuntos de atributos y la estructura de los argumentos a nivel de roles semánticos En algunos casos se realiza también un tratamiento más cuidadoso de determinados fenómenos semánticos. Por ejemplo, en Tatu and Moldovan (2007) se realizó un análisis sofisticado de las entidades nombradas distinguiendo para las entidades nombradas referentes a personas entre el nombre y el apellido de la persona. Además, a partir del RTE-3 se empezó a incluir también algunas formas de extracción de relaciones.

En el caso de los métodos lógicos los pares T-H se transforman en proposiciones o predicados con el fin de comprobar si hay implicación lógica. Dado que no siempre es posible realizar la demostración completa de implicación, hay ocasiones en las cuáles se van relajando las condiciones, por ejemplo eliminando predicados, y una vez terminada la demostración se utiliza un determinado valor umbral en función de las relajaciones realizadas para decidir si hay o no implicación.

En cuanto al uso de recursos, las bases de datos léxicas (en su mayoría WordNet y DIRT[25]) han sido utilizadas ampliamente, mientras que otros tipos de recursos

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

[25] http://www.aclweb.org/aclwiki/index.php?title=DIRT_Paraphrase_Collection

como FrameNet[26], VerbNet[27] y PropBank[28] también han sido tenidos en cuenta aunque en menor medida.

Además, ha habido sistemas que se han centrado en la detección de la no implicación buscando casos de falta de concordancia entre el texto y la hipótesis siguiendo las observaciones de Vanderwende and Dolan (2005), que sugieren que a veces es más fácil detectar la no implicación que los casos donde sí existe implicación.

Los resultados obtenidos por los sistemas actuales se mueven en torno al 49 % y el 80 % de *accuracy*, encontrándose la mayoría de los sistemas entre el 59 % y el 66 %. Las razones principales para este bajo rendimiento son el tamaño relativamente pequeño de las colecciones disponibles para entrenamiento y la falta de conocimiento lingüístico y del mundo. De hecho, los sistemas que mejor han tratado estas limitaciones son en gran parte los sistemas que mejores resultados han obtenido. Por ejemplo, Hickl et al. (2006) utilizó en el RTE-2 un corpus de implicación de gran tamaño obtenido automáticamente a partir de la Web y que contribuyó a mejorar en un 10 % sus resultados, mientras que Tatu et al. (2006) desarrolló un sistema basado en inferencia lógica que hacía uso de conocimiento lingüístico y del mundo obtenido a partir de varias fuentes.

Por otro lado y con el propósito de que los métodos empleados en RTE siguieran siendo efectivos a medida que los textos fueran más largos (problema que a partir del RTE-4 se tiene cada vez más en consideración para hacer más realista a la tarea), Hickl and Bensley (2007) consideraron que era necesario emplear técnicas que fueran capaces de enumerar el conjunto de proposiciones que se pueden inferir de cada texto y cada hipótesis. Para ello, su sistema se basa en la extracción de un conjunto de enunciados obtenidos a partir de la hipótesis y del texto y que son más sencillos que éstos. A continuación el sistema trata de identificar los enunciados del texto que hacen ciertos a los de la hipótesis. Una vez han sido extraídos todos los enunciados de los pares T-H, el trabajo se reduce a identificar los enunciados del texto que con mayor probabilidad soportan la veracidad de los enunciados de la hipótesis. Con este enfoque se consiguen unos resultados en torno al 80 % de *accuracy* sin necesidad de utilizar recursos adicionales de entrenamiento.

2.6. Confianza en la Evaluación

A la hora de evaluar sistemas se pueden utilizar distintas métricas de evaluación en función de qué comportamiento se desea juzgar. De este modo se puede tener que para evaluar una misma tarea existe más de una medida de evaluación, cada una de las cuáles se encarga de medir un aspecto distinto. Este ha sido el caso en IR, dado que el comportamiento de los sistemas de IR es suficientemente complejo como para ser resumido en un único valor, a lo largo de las distintas evaluaciones de IR se han propuesto distintas medidas de evaluación, teniendo que por ejemplo

[26] http://framenet.icsi.berkeley.edu/

[27] http://verbs.colorado.edu/ mpalmer/projects/verbnet.html

[28] http://verbs.colorado.edu/ mpalmer/projects/ace.html

el software de evaluación utilizado en el TREC devuelve hasta 85 valores distintos basándose en 20 medidas de evaluación distintas (Buckley, 2004). No se puede decir que haya una medida que sea mejor que otra, sino que hay medidas que son más adecuadas que otras para un determinado escenario o para evaluar un determinado comportamiento.

A la hora de utilizar medidas de evaluación no es solamente interesante conocer qué se está evaluando, sino también la confianza que se puede depositar en los resultados obtenidos, y por tanto en las conclusiones asociadas a esos resultados, y si dichos resultados son susceptibles de repetirse en futuras evaluaciones, sobre todo a la hora de decidir si un sistema es mejor que otro.

En esta sección se pretende realizar un repaso a distintos métodos para conocer la confianza que se puede establecer sobre los resultados de evaluación en IR. Este estudio se ha centrado en IR debido a que aunque los objetivos de esta tesis están más centrados en medidas de evaluación de QA, es poco el trabajo que hay en esta línea y el poco que hay está basado en métodos propuestos en IR.

En la comunidad de IR es ampliamente conocido que para que haya éxito a la hora de realizar una evaluación es importante contar con los siguientes tres componentes fundamentales (Hull, 1993):

1. Una colección de datos que sea adecuada.

2. Una medida o un conjunto de medidas que refleje la calidad de la búsqueda realizada.

3. Una medida de la significancia de las diferencias obtenidas entre sistemas.

En el contexto de IR, significancia se refiere a comprobar si la diferencia observada en los resultados de dos sistemas para un conjunto dado de “topics” representa una diferencia en el rendimiento de ambos sistemas sobre toda la población de “topics” (Zobel, 1998).

En un evaluación de sistemas de IR los resultados obtenidos con una determinada medida sólo indican efectividad para un conjunto concreto de “topics” sobre una colección particular, utilizándose estos resultados para decidir si un sistema es mejor que otro sobre dicho conjunto de “topics”. Sin embargo hay ocasiones en las cuáles al interpretar los resultados se comete el error de asumir que si dos sistemas tienen un rendimiento similar entonces dicha diferencia no es significativa, pero que si esa diferencia supera un determinado porcentaje, entonces si lo es. Dichas suposiciones pueden ser falsas (Savoy, 1997).

Si al realizar una evaluación se omite el estudio de significancia o la diferencia de rendimiento es pequeña, la conclusión de que un sistema es mejor que otro no es fiable. El término fiable hace referencia en este contexto a determinar si en el caso de que el experimento se repitiese sobre otro conjunto de datos, que en cierto modo sea similar al utilizado, los resultados serían parecidos a los obtenidos (Jensen et al., 2007). Además, está claro que un mejor entendimiento de las propiedades de

las medidas de evaluación empleadas permite al investigador obtener mayores y mejores conclusiones de los resultados de un experimento (Hull, 1993).

Los métodos utilizados en IR para comprobar la confianza en los resultados se pueden dividir en dos grandes grupos en función de la metodología que se utiliza:

- Métodos basados en test estadísticos
- Métodos basados en test empíricos

A continuación se describe en qué consiste cada uno de estos métodos.

2.6.1. Métodos basados en Tests Estadísticos

Los test estadísticos utilizan técnicas estadísticas para estudiar si una determinada afirmación es confirmada o rechazada por los datos de una muestra. La ventaja que presenta este tipo de métodos es que permiten detectar mejoras significativas de rendimiento incluso en los casos en los cuáles las mejoras en cuanto a valores numéricos son pequeñas (Smucker et al., 2007).

Un test estadístico permite evaluar la confianza en la afirmación de que una diferencia en los resultados es significativa. Por tanto, el objetivo es comprobar si los resultados son realmente significativos o si por el contrario se deben a la casualidad y es poco probable que vuelvan a reproducirse (Sanderson and Zobel, 2005; Webber et al., 2008; Savoy, 1997).

Es importante hacer notar que hay que distinguir entre significancia estadística y significancia práctica. Por un lado, la significancia estadística hace referencia a rechazar la hipótesis de que dos resultados son iguales basándose en un test cuantitativo. Sin embargo, la significancia práctica se podría definir como la diferencia en rendimiento que es perceptible por el usuario, la cuál es a veces más difícil de medir (Hull, 1993).

El método que se suele utilizar para comprobar la significancia estadística es el contraste de hipótesis (también denominado test de hipótesis o prueba de significancia), el cuál es una metodología de inferencia estadística que se utiliza para juzgar si una propiedad que se supone cumple una población estadística es compatible con lo observado en una muestra de dicha población. En el caso de IR la propiedad sería que *el sistema A es mejor que el sistema B*; la población sería el conjunto de todos los "topics" y colecciones posibles que existen en el mundo; y la muestra que se utiliza para el estudio se corresponde con los "topics" y colecciones utilizados en la evaluación. Por tanto, mediante el contraste de hipótesis se trata de averiguar si los resultados obtenidos sobre una determinada colección de datos y "topics" son significativos y se pueden esperar los mismos resultados sobre otra colección y conjunto de "topics" distinto.

Cuando se realiza un test de contraste de hipótesis sobre resultados de IR se parte de la suposición de que los sistemas de IR que se van a comparar son equivalentes en términos de rendimiento. A esta suposición se le denomina hipótesis nula H_0, la cuál tratará de rechazar el test. Para ello se calcula un valor al que se

denomina *p-valor*, el cuál representa *el mínimo valor de significancia al cuál puede rechazarse* H_0.

Al principio de realizar el experimento de contraste de hipótesis ha de fijarse un nivel de significancia α, de tal modo que si el *p-valor* es menor que α se rechaza H_0 y se acepta la hipótesis de que los resultados de los dos sistemas son significativamente distintos. Generalmente se suele utilizar 0.05 para el valor de α, mientras que un valor menor implica tener mayor confianza a la hora de rechazar H_0 y concluir que los métodos son distintos (Keen, 1992). Análogamente, un valor mayor indicaría tener una menor confianza en la conclusión obtenida.

El hecho de que el resultado del test indique que la diferencia observada no es significativa no quiere decir que no haya una diferencia real entre los sistemas comparados, sino que el test no ha sido capaz de detectarla. En este caso no se puede asegurar la relevancia estadística del experimento realizado.

Hay varias categorías de test estadísticos:

- **Paramétricos:** son los métodos más comunes y hacen un determinado número de suposiciones acerca de la distribución de los datos. Su uso en IR es cuestionable debido a que la mayoría de este tipo de test están basados en distribuciones que no siguen las medidas utilizadas en IR (Savoy, 1997).
- **No paramétricos:** estos métodos no requieren especificar una distribución teórica de los datos, además de permitir contrastar hipótesis de valores que se calculen sin utilizar la media aritmética. Es decir, estos métodos se pueden aplicar sobre los resultados de medidas de evaluación que no utilicen la media aritmética (como aquellas medidas que utilizan por ejemplo la media armónica, como puede ser *F*). Sin embargo, estos test no son tan poderosos como los paramétricos.

En función del número de resultados que se comparan entre si, se puede establecer un división de los métodos de contraste de hipótesis: comparaciones realizadas entre dos sistemas y comparaciones realizadas entre más de dos sistemas. En los siguientes apartados se mencionan los métodos más comunes para cada tipo de comparación.

2.6.1.1. Comparaciones entre dos Métodos

En este tipo de contrastes de hipótesis se comparan los resultados de dos sistemas de IR con el fin de tratar de decidir si un sistema produce resultados significativamente mejores que los del otro sistema. Dado que en IR los resultados se calculan primero por "topic" para luego calcular una media sobre el conjunto de todos los "topics", a la hora de comparar dos sistemas se tiene que por cada "topic" hay un par de valores que pueden ser comparados (un valor por cada sistema). Los métodos estadísticos analizan la diferencia entre los valores de cada "topic" para concluir si la diferencia al utilizar todos los "topics" es o no significativa.

El método más utilizado en estos casos es el test-t pareado. Este test compara la Test-t pareado

magnitud de las diferencias por "topics" entre los sistemas, con la variación entre dichas diferencias. Si la diferencia media es grande en comparación con su error estándar, entonces se considera que los métodos son significativamente diferentes. Sin embargo, la principal crítica que recibe este test es que asume que los resultados siguen una distribución normal, lo cuál no tiene por qué cumplirse siempre en IR (van Rijsbergen, 1979). A pesar de este hecho, ha habido estudios en los cuáles se ha visto que mientras que los resultados podrían no seguir una distribución normal, el test-t es relativamente robusto a muchas violaciones de normalidad. Solamente en casos con un gran falta de simetría se podría comprometer la validez del test, pero estos casos pueden ser fácilmente comprobados mediante una representación gráfica de los datos (Hull, 1993).

Hay dos alternativas no paramétricas: el test de los rangos de Wilcoxon y el test de los signos.

Test de los rangos de Wilcoxon

En el test de los rangos de Wilcoxon se sustituye cada diferencia (la obtenida entre resultados de un mismo "topic") por el ranking de su valor absoluto y se multiplican por el signo de la diferencia. A continuación se compara la suma de los rankings de cada sistema con el valor esperado bajo la suposición de que los dos sistemas son iguales.

Test de los signos

El test de los signos se fija únicamente en el signo de las diferencias, sin tener en cuenta sus magnitudes. Si uno de los dos sistemas obtiene mejores resultados que el otro con mayor frecuencia de lo que en media se podría esperar, entonces se considera que hay una gran evidencia de que ese sistema es mejor.

En Smucker et al. (2007) se realizó un estudio según el cuál el test de los rangos de Wilcoxon y el de los signos discrepan entre ellos y con otros métodos en algunas ocasiones. Además, en el mismo estudio se observó que había casos en los cuáles se podía fallar a la hora de detectar que había significancia, por lo que estos métodos no se consideran muy fiables.

Métodos basados en bootstrapping

Otro tipo de métodos estadísticos para comparar entre si los resultados de dos sistemas son los basados en "bootstrapping". Este tipo de métodos no requieren hacer suposiciones sobre la distribución de las medidas de evaluación debido a que las distribuciones se estiman haciendo reemplazamiento sobre los datos usados para el estudio (Jensen et al., 2007; Cormack and Lynam, 2006). De hecho, este tipo de test presenta la ventaja de que permite estudiar cualquier tipo de medida y no solo aquellas basadas en la media aritmética (Sakai, 2006a,b), lo cuál no puede realizarse con otros test estadísticos como por ejemplo el test-t.

Son varios los estudios que sugieren que es preferible utilizar un test basado en "bootstrapping" a un test-t pareado (Sakai, 2006a; Smucker et al., 2007). Además, mediante el método basado en "bootstrapping" es posible estimar la diferencia en resultados requerida para obtener un determinado nivel de significancia.

2.6.1.2. Comparaciones entre más de dos Métodos

El enfoque tradicional que se suele seguir para comparar más de dos sistemas es el análisis de la varianza (en inglés Analysis of Variance, ANOVA). La principal

ventaja que presenta este método en lugar de realizar test estadísticos sobre todos las combinaciones de pares de sistemas es que permite reducir el trabajo a realizar (Hull, 1993).

En este método primero se realiza un test para comprobar si hay alguna diferencia entre sistemas. En caso de detectarse alguna diferencia, se realizan comparaciones posteriores para determinar qué sistemas son significativamente distintos.

Cuando se utilizan métodos no paramétricos de ANOVA, en general estos métodos remplazan los datos por rankings basados en el rendimiento dentro de cada "topic". A pesar de que la utilización de rankings reduce el poder del test, presenta la ventaja de que es necesario realizar menos suposiciones sobre la distribución de los datos.

2.6.2. Métodos basados en Test Empíricos

El otro enfoque utilizado para estudiar la confianza que se puede depositar en los resultados lo constituyen los métodos basados en test empíricos, los cuáles tienen en común la utilización de los resultados de un gran número de sistemas para poder obtener una regla general acerca de la diferencia entre resultados que se necesita para poder concluir que un sistema es mejor que otro. A continuación se describen dos métodos de este tipo que se han utilizado en IR.

2.6.2.1. Estabilidad y Poder de Discriminación

Uno de los principales métodos empíricos es el descrito en Buckley and Voorhees (2000), donde se define un método para calcular empíricamente la tasa de error asociada a la conclusión *el sistema A es mejor que el sistema B* para una determinada medida de evaluación. La tasa de error sirve para medir la estabilidad de una medida de evaluación de tal manera que cuanto menor sea la tasa de error, se considera que la medida de evaluación es más estable.

Los resultados obtenidos mediante este método no pretenden concluir que las medidas con una alta tasa de error no deban de ser utilizadas. Cada medida de evaluación evalúa un determinado comportamiento y su elección se debe de realizar en función de los objetivos que se plantean a la hora de realizar la evaluación. Los resultados obtenidos mediante este método han de verse de tal manera que para tener la misma certeza de qué sistema es mejor cuando se utiliza una medida con una alta tasa de error, es necesario utilizar más datos de evaluación o tener mayores diferencias entre los valores obtenidos que cuando se utiliza una medida con una tasa de error menor.

Para comprender mejor el método, a continuación se describen los datos que se utilizaron en la propuesta del mismo, los cuáles fueron tomados a partir del TREC-8 Query Track (Buckley and Walz, 1999): Se define un conjunto de consultas como una colección con 50 consultas (una consulta por "topic"). Para el experimento se contaba con 21 conjuntos de consultas distintas de tal manera que en cada conjunto había consultas para los mismos 50 "topics", siendo dichas consultas distintas entre

conjuntos. En la tarea participaron 9 sistemas, cada uno de los cuáles devolvió 1000 documentos por cada una de las 1050 consultas (21 versiones de los 50 "topics"). Además, se contaba con la lista de documentos relevantes para cada uno de los 50 "topics".

El método de estabilidad sirve para cuantificar la tasa de error asociada a la decisión de que un sistema de IR es mejor que otro basándose en un experimento con un determinado número de "topics", una medida de evaluación determinada y un cierto valor utilizado para decidir si los resultados de los dos sistemas son diferentes o equivalentes. A este valor se le denomina umbral de equivalencia (en inglés se le denomina *fuzziness value* y dada su función hemos decidido utilizar dicha traducción). Este umbral de equivalencia representa la diferencia en porcentaje entre los resultados de dos sistemas de tal modo que si la diferencia es menor que este umbral, se considera que ambos resultados son equivalentes y hay un empate entre los sistemas. Por ejemplo, si el umbral de equivalencia se fija al 10 %, los resultados que estén el uno dentro del otro en dicho margen del 10 % serán considerados como equivalentes.

Para realizar el experimento, por cada conjunto de consultas se calcula la media de una medida de evaluación (por ejemplo *precisión*, *p@1*, *average precision*, etc) sobre dicho conjunto para cada uno de los 9 sistemas. A continuación se comparan los resultados de cada par de sistemas sobre dicho conjunto para ver si el primer sistema es mejor, peor o igual que el segundo en función del umbral de equivalencia escogido. Este proceso se puede repetir un número determinado de veces utilizando en cada ocasión un conjunto de consultas distinto.

Por ejemplo, al utilizar los datos del TREC-8 Query Track se pueden realizar entre cada par de sistemas 21 comparaciones (una por cada uno de los 21 conjuntos de consultas), realizándose en total 756 comparaciones (21 conjuntos por 36 parejas de sistemas). Con el fin de aumentar el número de comparaciones realizadas durante el experimento para así tener datos más fiables, los autores combinaron los conjuntos de consultas aleatoriamente, generando nuevos conjuntos de consultas disjuntos entre si.

Para calcular la tasa de error se utiliza la intuición de que al comparar dos sistemas el mejor sistema es aquél que gana en mayor número de comparaciones. Bajo este punto de vista, hay un error cada vez que gana el considerado como peor sistema. De este modo se puede definir la tasa de error como el número total de errores que ha habido durante todo el experimento (veces que ha ganado el peor sistema), dividido del número total de comparaciones realizadas. De acuerdo con este procedimiento, la fórmula de la tasa de error para una determinada medida *M* se define en la Ecuación (2.28), donde: S es el conjunto de sistemas que se utilizan en el experimento, $|x > y|$ representa el número de veces que el sistema *x* es mejor que el sistema *y*, $|y > x|$ representa el número de veces que el sistema *y* es mejor que el sistema *x*, mientras que $|x == y|$ representa el número de veces que se considera que empatan los sistemas *x* e *y*. Todos estos valores son los que se obtienen al realizar las comparaciones sobre los resultados obtenidos haciendo uso de la medida *M*.

$$Tasa\ de\ error_M = \frac{\sum_{x,y \epsilon S} min(|x > y|, |y > x|)}{\sum_{x,y \epsilon S} (|x > y| + |y > x| + |x == y|)} \tag{2.28}$$

Por otro lado, observando el número de empates que hay entre sistemas se puede calcular la proporción de empates para una determinada medida *M* mediante la Fórmula (2.29). Esta proporción de empates es una manera de ver el poder de discriminación de la medida, de modo que con que mayor proporción de empates tenga una medida, menor poder de discriminación tiene. Tener una medida con bajo poder de discriminación tiene como consecuencia que para poder concluir que un sistema es mejor que otro, es necesario tener una mayor diferencia entre resultados que en el caso de utilizarse una medida más discriminatoria.

$$Proporción\ de\ empates_M = \frac{\sum_{x,y \epsilon S} |x == y|}{\sum_{x,y \epsilon S} (|x > y| + |y > x| + |x == y|)} \tag{2.29}$$

Siguiendo el procedimiento descrito anteriormente, el experimento se realizó utilizando distintas medidas de evaluación de IR. Al realizar experimentos variando el tamaño de los conjuntos de "topics" se observó que conforme se aumentaba el número de "topics" disminuía la tasa de error, lo cuál es consecuente con la metodología utilizada para evaluar sistemas de IR (Voorhees, 2001b). También se corroboró la intuición de que mayores valores para el umbral de equivalencia provocan una disminución de la tasa de error, aunque también se disminuye el poder de discriminación. Por tanto, al utilizar mayores umbrales de equivalencia para decidir qué sistema es mejor aumenta la confianza en los resultados. Sin embargo, el coste asociado es tener menor capacidad de discriminación entre los distintos sistemas.

Al comparar distintas medidas de IR se observó que las medidas que hacían uso de poca información, como por ejemplo *p@1*, eran poco estables, mientras que *MAP*, que mide el área bajo la curva *cobertura-precisión*, era más estable.

Este método ha sido aplicado tanto sobre otras medidas de IR (Sakai, 2007b), como sobre medidas de QA (Sakai, 2007a), y aunque no está basado en métodos estadísticos, los resultados son parecidos a los obtenidos utilizando métodos estadísticos basados en "bootstrapping" (Sakai, 2006a).

Una de las utilidades de este método reside en que permite aportar para una determinada colección su tasa de error, lo cuál puede servir a los investigadores para interpretar sus resultados sobre esa colección.

2.6.2.2. Método Swap

Otro método empírico lo constituye el trabajo desarrollado en Voorhees and Buckley (2002), el cuál tiene como objetivo obtener empíricamente una relación entre el número de "topics" utilizados en un experimento, la diferencia observada en rendimiento entre dos sistemas y la confianza que se puede depositar en la afirmación de que un sistema es mejor que otro. Para ello, este método propone calcular la tasa de error en función del número de "topics" utilizados y la diferencia en resultados.

La motivación para diseñar este método surgió a partir de la observación de que el comportamiento de los sistemas de IR varía en gran medida entre los distintos "topics" de una colección de evaluación, motivo por el cuál se recomienda tener un número suficiente de "topics" sobre el que realizar la evaluación si se desean obtener resultados significativos. La tasa de error obtenida mediante este método sirve para cuantificar la probabilidad de que al usar un conjunto distinto de "topics" del mismo tamaño se obtenga una conclusión distinta sobre qué sistema es mejor. Es decir, este método se basa en calcular la probabilidad de que la decisión de que un sistema de IR es mejor que otro cambiará si se utiliza un conjunto de "topics" distinto.

Para estimar la probabilidad de que haya un cambio se realiza un gran número de comparaciones entre sistemas utilizando distintos conjuntos de "topics" (de igual tamaño y disjuntos entre si) y se cuenta el número de veces que hay cambios en la decisión tomada (acerca de qué sistema es el mejor de los dos comparados). Repitiendo este proceso un número elevado de veces se puede estimar la probabilidad de que haya un cambio en la decisión de qué sistema es mejor cuando se utiliza una determinada medida de evaluación para comparar dos sistemas cualesquiera.

El primer paso para calcular la tasa de error según este método consiste en definir 21 ranuras (en inglés se utiliza el término *bin*) para llevar control de las diferencias entre los resultados de un par de sistemas de acuerdo a una determinada métrica de evaluación. Estas ranuras funcionan del siguiente modo: la primera ranura representa las diferencias entre resultados menores de 0.01; la segunda ranura representa las diferencias mayores o iguales que 0.01 y menores que 0.02; las siguientes ranuras se van incrementando en 0.01, de tal modo que la última ranura representa las diferencias mayores o iguales a 0.2. Por ejemplo, si la diferencia entre los resultados de dos sistemas utilizando *MAP* es de 0.005, esta diferencia estaría representada en la primera ranura.

Una vez definidas las ranuras, para cada par de sistemas a comparar se utilizan dos conjuntos de "topics" disjuntos y se evalúa sobre ellos a los dos sistemas. Si el resultado acerca de qué sistema es mejor es distinto al utilizar los dos conjuntos, se considera que hay una discrepancia (*swap* en inglés, de ahí el nombre del método) y se incrementa un contador que lleva la cuenta de las discrepancias que hay para la ranura correspondiente a la diferencia en resultados entre el par de sistemas comparados. Siguiendo el ejemplo anterior (donde la diferencia entre dos sistemas era de 0.005), se incrementaría el contador de discrepancias de la primera ranura.

Dicho procedimiento se repite con todos los pares de sistemas disponibles, realizando por cada par un número de comparaciones determinado de antemano sobre distintos conjuntos de "topics". La Figura 2.11 (página 85) muestra el algoritmo que se sigue para llevar a cabo este procedimiento dada una determinada medida *M* para calcular la tasa de error de dicha medida para cada diferencia, es decir, para cada ranura.

```
por cada par de sistemas x, y ϵ S
  por cada ejecución de 1 hasta N
    seleccionar dos conjuntos de topics A, B ϵ T de tamaño z disjuntos entre si
    d_M(A) = M(x, A) − M(y, A);
    d_M(B) = M(x, B) − M(y, B);
    incrementar el contador_ranura correspondiente a d_M(A);
    si d_M(A) * d_M(B) < 0
      incrementar contador_discrepancias de ranura correspondiente a d_M(A);
por cada ranura r
  tasa_error (r) = contador_discrepancias (r) / contador_ranura (r);
```

Figura 2.11: Algoritmo para calcular la tasa de error para una medida *M* de acuerdo al método de swap dado un conjunto de sistemas *S*, un número de ejecuciones N, un conjunto de "topics" T y un tamaño z para cada conjunto de "topics".

La tasa de error obtenida mediante este procedimiento se interpreta como se muestra en el siguiente ejemplo: se parte de que se comparan dos sistemas cuya diferencia en resultados es *d*, y para la diferencia *d* la tasa de error para un determinado número de "topics" es del 9 % (es decir, la tasa de error para esa ranura es del 9 %). Al repetir la comparación entre ambos sistemas sobre 100 conjuntos distintos de "topics" del mismo tamaño que el utilizado para el experimento original se espera que sobre 91 conjuntos sea mejor uno de los dos sistemas, mientras que el otro será mejor en los 9 conjuntos restantes.

El experimento que realizaron los autores se llevó a cabo utilizando la medida de evaluación *MAP* (descrita en la sección 2.1.3.2 de la página 39) sobre los resultados de sistemas del TREC-3 al TREC-10, cada uno de los cuáles trabajaba sobre 50 "topics", a excepción del TREC-4 que utilizaba 49. Al igual que con los experimentos basados en estabilidad, se observó que al aumentar el número de "topics" la tasa de error disminuía. También se comprobó que la tasa de error disminuía al aumentar la diferencia entre sistemas, lo cuál coincide con lo observado en el método de estabilidad.

Este método ha sido aplicado para comprobar la confianza en la evaluación desarrollada en QA en el TREC-2002 (Voorhees, 2002) y el TREC-2003 (Voorhees, 2003). Además, al igual que el método para calcular la estabilidad, este método no está basado en un test de significancia estadística pero los resultados son consistentes con los obtenidos utilizando métodos estadísticos basados en "boots-

trapping" (Sakai, 2006a).

Una extensión a este método es la realizada por Tetsuya Sakai para medidas de IR (Sakai, 2007b) y medidas de QA (Sakai, 2007a), utilizando en ambos casos este método para calcular la sensibilidad de distintas medidas de evaluación. La sensibilidad en este contexto hace referencia al poder de discriminación de una medida, de modo que con más sensible sea la medida más útil será ésta para discriminar entre sistemas ya que habrá menos empates.

Para medir la sensibilidad se fija primero un determinado nivel de confianza sobre el que calcularla, de modo que si el nivel de confianza utilizado es del 95 % (se acepta 5 % de error como máximo), se toma la primera ranura (empezando por las que representan diferencias más pequeñas) para la cuál el error es menor del 5 %. Entonces, calculando el porcentaje de comparaciones totales realizadas en el experimento que cumplen dicha diferencia (diferencias iguales o mayores) se obtiene la sensibilidad de la medida. La interpretación de este procedimiento es que cuanto más sistemas cumplen dicha diferencia, mejor discrimina la medida ya que habrá menos empates.

2.7. Recapitulación

En este capítulo se han descrito los procedimientos más comunes para realizar la evaluación de varias tareas típicas del Procesamiento del Lenguaje Natural como son la Recuperación de Información, la Extracción de Información y la Búsqueda de Respuestas, y otras más nuevas como la Validación de Respuestas y el Reconocimiento de Implicación Textual.

Al describir la evaluación de los sistemas de Búsqueda de Respuestas se ha podido observar que aunque se han realizado varios intentos para evaluar la capacidad de los sistemas de QA emitiendo juicios acerca de la validez de sus respuestas, este tipo de evaluaciones no ha tenido un gran éxito y en las evaluaciones actuales sólo cuentan los aciertos de los sistemas y no se penalizan los fallos. De este modo, la validación queda relegada a un segundo plano: es mejor dar una respuesta errónea que no responder. Dado que la definición de la tarea de evaluación marca profundamente las arquitecturas de los sistemas participantes, la comunidad ha relegado la validación a un segundo plano y no se han desarrollado en gran medida este tipo de tecnologías.

Sin embargo, hay estudios según los cuáles cabe esperar que la incorporación de una fase de validación al procesamiento de los sistemas de QA pueda conseguir mejorar los resultados actuales (Magnini et al., 2002b). De hecho, una de las motivaciones para el desarrollo de sistemas de implicación textual es que su uso podría servir para mejorar los resultados en Búsqueda de Respuestas al realizarse una etapa de validación basada en implicación textual.

En el capítulo se ha podido observar que hasta la fecha, los investigadores no han prestado demasiada atención a la evaluación de sistemas de Validación de Respuestas. De hecho, las pocas evaluaciones que se han realizado han estado enfo-

cadas a evaluar sistemas concretos y no se han realizado comparaciones con otras aproximaciones, ni el estudio de la mejora que podría suponer el uso de módulos de Validación en los sistemas de Búsqueda de Respuestas. Además, todas estas evaluaciones se han centrado en sistemas que realizan validación, sin atender a los sistemas que realizan selección de respuestas. Parece claro que la definición de un marco de evaluación de sistemas de AV que permita comparar distintos enfoques entre si y estudiar la mejora que podría suponer la incorporación de módulos de AV en QA, serviría para fomentar el desarrollo de sistemas de AV, su incorporación en QA y la mejora de resultados en QA.

Capítulo 3

Propuesta de Modelo de Validación de Respuestas

En este capítulo se propone un modelo para realizar la Validación de Respuestas basándose en el Reconocimiento de Implicación Textual. La motivación para realizar esta propuesta surge del estudio realizado sobre los métodos empleados para realizar Validación de Respuestas antes de este trabajo (los cuáles fueron descritos en la sección 2.4.3 de la página 64), los cuáles no realizaban, en general, un análisis profundo de la respuesta y la pregunta en su contexto.

En este capítulo se realiza también un estudio sobre la viabilidad de la propuesta realizada. Para realizar este estudio se construyó una colección que permite evaluar el enfoque propuesto, además de suponer una motivación para desarrollar la metodología de evaluación desarrollada en el Capítulo 5 (página 125), la cual se define partiendo del modelo desarrollado en este capítulo.

3.1. Validación de Respuestas como Problema de Implicación Textual

En esta sección se realiza la propuesta de modelar la Validación de Respuestas como un problema de Implicación Textual. Para ello, primero se describe la motivación para realizar dicha propuesta, para a continuación describir la propuesta realizada y las subtareas que surgen al modelar la Validación de Respuestas como un problema de RTE: la generación automática de hipótesis y la decisión final sobre el valor de implicación, y por tanto de validación.

3.1.1. Motivación

Como se ha estudiado en la sección 2.4.3 (página 64), son muchos los métodos que para realizar la Validación de Respuestas se basan en buscar evidencias de corrección de las respuestas en fuentes externas. La más usual de estas fuentes es la Web debido a la gran cantidad de información redundante que contiene. Estos

métodos han demostrado obtener buenos resultados y que con su uso se podría mejorar el rendimiento de los sistemas de QA (Magnini et al., 2002a).

Sin embargo, en este trabajo se considera que la validación de una respuesta ha de realizarse dentro de su contexto, en el sentido de que las evidencias que sirvan para validar la respuesta han de ser obtenidas de la fuente de la cuál se ha obtenido la respuesta y no de una fuente de datos externa. Es cierto que puede que sea necesario el uso de conocimiento externo para realizar la validación, pero dicho conocimiento ha de servir como apoyo o complemento para el proceso de validación y no como evidencia de corrección de la respuesta. En el modelo que se propone en este capítulo se parte de que la decisión de validación ha de tomarse considerando el contenido del documento que contiene a la respuesta, pudiéndose utilizar también el contenido de una fuente de conocimiento externa, pero en ningún caso haciendo uso únicamente de la fuente de conocimiento externa [1].

Además, los métodos basados en redundancia realizan un análisis poco profundo de la respuesta, relacionando a la respuesta con la pregunta en términos de relevancia. Estos métodos no se han preocupado de tratar los fenómenos semánticos que han de existir para determinar si la respuesta es o no correcta (Harabagiu and Hickl, 2006). En esta tesis se considera que la aplicación de métodos que sean capaces de realizar un análisis más profundo de la pregunta y la respuesta podría suponer una mejora en el rendimiento de los sistemas de AV, lo cuál llevaría asociado la mejora de resultados en QA. Dado que los sistemas de RTE sí tratan de realizar esta aproximación a la semántica entre textos, en esta tesis se decidió proponer un modelo de AV basado en RTE.

3.1.2. Modelo de Validación de Respuestas

En la propuesta realizada en esta tesis, un sistema de AV recibe como entrada una *Pregunta*, una *Respuesta* y un *Texto Soporte* que supuestamente sirve como justificación de por qué la *Respuesta* dada es correcta. Si el sistema de AV considera que la respuesta es correcta, ésta se devuelve como salida del sistema. En caso contrario se indica que la respuesta es incorrecta, lo cuál puede servir para indicar al sistema de QA que debe de generar una nueva respuesta.

Esta propuesta se basa en la idea descrita en Dagan et al. (2005) según la cuál cuando un sistema de QA devuelve una *Respuesta* y un *Texto Soporte* para dicha *Respuesta*, un sistema de RTE daría por válida la *Respuesta* si el *Texto Soporte* implica a una hipótesis creada como combinación de la *Respuesta* junto con la formulación afirmativa de la *Pregunta*. La Figura 3.1 (página 91) muestra la arquitectura de un sistema de AV que usa RTE dentro del contexto de un sistema de QA. Como se puede ver en la Figura, dada una pregunta el sistema de QA genera una respuesta y un texto soporte. A continuación el módulo de AV construye una hipótesis a partir de la pregunta y la respuesta candidata y utiliza un sistema de

[1] Este uso de fuentes de conocimiento externas es similar al propuesto en RTE, donde se pide inferir el significado de la hipótesis a partir del texto pudiendo utilizarse conocimiento adicional, pero en ningún caso el conocimiento por si solo puede implicar a la hipótesis (Dagan et al., 2005)

RTE para decidir si la hipótesis está implicada o no por el texto soporte. Si hay implicación entonces se considera que la respuesta es correcta y se devuelve ésta como salida. En caso contrario se le indica al sistema de QA que la respuesta que generó era incorrecta, lo que puede servir para que el sistema de QA genere otra respuesta.

De acuerdo a la arquitectura de la Figura 3.1, los sistemas de AV basados en RTE deben de tener en cuenta el problema adicional de la generación automática de hipótesis como combinación de preguntas y respuestas, el cuál es un nuevo subproblema en el contexto de un sistema de QA.

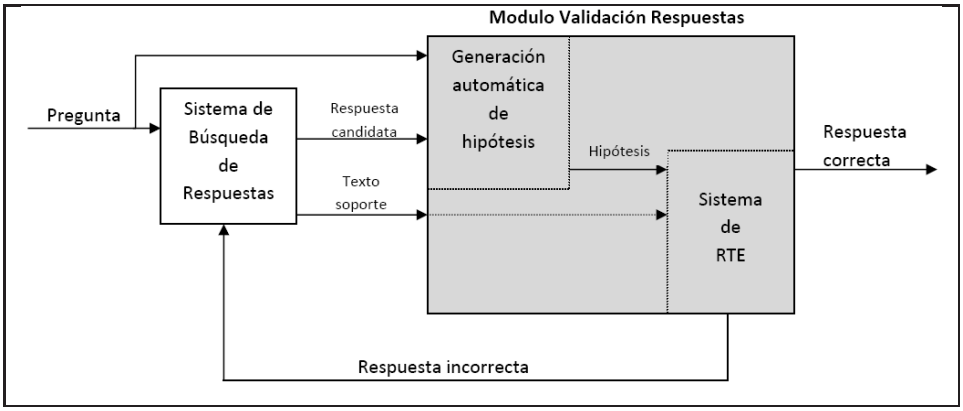

Figura 3.1: Contexto de un sistema de Validación de Respuestas basado en RTE dentro de un sistema de Búsqueda de Respuestas.

3.1.3. Generación Automática de Hipótesis

Como se ha indicado previamente, al utilizar RTE para tomar la decisión de validación, se hace necesario tener la capacidad de generar hipótesis combinando preguntas y respuestas, tarea que no se tenía antes dentro del contexto de los sistemas de QA. Desde un punto de vista teórico la principal dificultad radica en asegurarse de que las hipótesis generadas a partir de respuestas correctas serán validadas por el módulo de RTE. Es decir, que se cumpla la propuesta de Dagan et al. (2005) de que la respuesta será válida si la combinación de ésta con la forma afirmativa de la pregunta es implicada por el texto soporte.

Los enfoques para generar hipótesis como combinación de preguntas y respuestas pueden ser diversos. Se puede partir desde la mera concatenación de ambos elementos, hasta la creación de hipótesis más elaboradas mediante el uso de patrones de pregunta. En este caso, por cada tipo de pregunta se tendría un patrón que indica cómo hay que combinar la respuesta y la pregunta para crear una hipótesis de una calidad razonable.

Otro enfoque puede ser el de crear una hipótesis en forma lógica. En este caso la hipótesis podría crearse como combinación de las formas lógicas de la pregunta y la respuesta, sin necesidad de crear previamente una hipótesis textual.

En cualquier caso, el método elegido para construir la hipótesis depende en gran medida del tipo de procesamiento que se vaya a realizar para tomar la decisión de implicación. Por ejemplo, en el caso de utilizar demostradores lógicos es evidente que la hipótesis se tiene que construir en forma lógica.

3.1.4. Decisión de Implicación

Una vez se tiene un par texto-hipótesis (hipótesis construida a partir de una pregunta y una respuesta), el último paso de un sistema de AV basado en RTE consiste en tomar la decisión de si el texto implica o no a la hipótesis. O lo que es lo mismo, si la respuesta es o no correcta de acuerdo al texto soporte. En definitiva, la decisión se modela como un problema de clasificación binaria: decidir si una respuesta pertenece a la clase *correcta* o a la clase *incorrecta*.

El hecho de plantear la decisión final como un problema de clasificación binaria hace que se abra un amplio rango de posibilidades para tomar dicha decisión, ya que se pueden usar desde métodos basados en aprendizaje automático hasta métodos basados en demostradores lógicos. Todos estos métodos tienen sus propias particularidades y permiten el uso de un amplio rango de recursos, representaciones y métodos distintos.

Según Harabagiu et al. (2003), la validación de respuestas basada en RTE se puede tratar de cuatro maneras distintas:

1. Obtener información lingüística del par texto-hipótesis y utilizar esta información en un clasificador que decida si hay o no implicación.

2. Evaluar la probabilidad de que haya implicación entre el texto y la hipótesis.

3. Representar el conocimiento de la hipótesis y el texto en un lenguaje que pueda ser asociado a un mecanismo de inferencia y aplicar dicho mecanismo de inferencia para tomar la decisión final.

4. Utilizar la definición clásica de implicación en Inteligencia Artificial y construir modelos del mundo en los cuáles tanto la hipótesis como el texto sean ciertos. A continuación comprobar si el modelo de la hipótesis está contenido en el modelo del texto.

Cada uno de estos métodos tiene sus particularidades y problemas asociados. En el caso de los enfoques lógicos (los cuáles se encuadran en el tercer tipo), se seguiría la intuición de Groenendijk (1999) y Lewis (1988) de que la relación que hay entre una pregunta y una respuesta correcta a dicha pregunta puede ser modelada en términos de implicación lógica. En este tipo de métodos podría necesitarse conocimiento del mundo, lo cuál presenta el problema adicional de cómo adquirir y representar este conocimiento. Al hacer uso de conocimiento externo hay que tener en cuenta que ha de ser el texto el que soporte la veracidad de la respuesta y no el conocimiento adicional, es decir, que la hipótesis sea implicada por el

texto más el conocimiento adicional y no solo por dicho conocimiento. Además, otro problema que tienen los enfoques lógicos es que suelen tener asociado un alto coste computacional.

3.2. Estudio de Viabilidad

Una vez realizada la propuesta de sistemas de AV basados en RTE, en esta sección se procede a estudiar la viabilidad de este enfoque. Para realizar este estudio se construyó un corpus de pares texto-hipótesis similar a la colección de evaluación del primer RTE Challenge (Dagan et al., 2005), con la peculiaridad de que todos los pares fueron construidos a partir de la salida de sistemas reales de QA.

El principal motivo por el que se decidió construir este corpus en vez de hacer uso de los pares de los RTE Challenges obtenidos a partir de sistemas de QA, radica en que los pares de los RTE Challenges fueron seleccionados, filtrados y adaptados manualmente. Es decir, aunque provienen de sistemas reales de QA, dichos pares no reflejan fielmente los fenómenos a los cuáles debe de enfrentarse un sistema real de AV debido a la manipulación a la cuál han sido sometidos.

Otra colección que hubiera podido servir de inspiración es la construida por MITRE para su participación en el primer RTE Challenge (Burger and Ferro, 2005). En dicha participación MITRE decidió abordar la tarea de RTE realizando alineamiento estadístico, para lo cuál necesitaron una colección de entrenamiento mayor que la suministrada por la organización. Para construir su colección de entrenamiento, MITRE se basó en la observación de que en la mayoría de los pares de la tarea de RTE donde hay implicación la hipótesis es una paráfrasis de un subconjunto del texto correspondiente. Basándose en dicha observación construyeron su colección de entrenamiento a partir de un conjunto de noticias de periódico, partiendo de la idea de que en estas noticias suele ser frecuente que el titular sea una paráfrasis del primer párrafo de la noticia. A continuación realizaron un procesamiento semiautomático para seleccionar los 100.000 pares más prometedores, de entre los cuáles se estimaba que el 75 % contenían una relación de implicación.

MITRE obtuvo uno de los mejores resultados del primer RTE Challenge haciendo uso de este corpus. Aunque este corpus es útil para entrenar sistemas estadísticos de RTE, carece de juicios humanos para cada par y no tiene relación con la tarea de QA, por lo cuál no es útil para los propósitos de este trabajo.

La colección que se realizó para comprobar la viabilidad de la propuesta fue desarrollada en castellano y se le denominó SPARTE (SPAnish RTE). Para su construcción se partió de las preguntas, respuestas y juicios humanos utilizados en la tarea de QA del CLEF 2003, 2004 y 2005 (Magnini et al., 2003, 2004; Herrera et al., 2004; Vallin et al., 2005). El Cuadro 3.1 (página 94) muestra por cada edición del QA@CLEF el número de preguntas, respuestas y runs de los que se partió para el desarrollo de la colección, mostrándose también el número de pares pregunta-respuesta (una pregunta y una respuesta dada a dicha pregunta) disponibles al comienzo del desarrollo de la colección.

Cuadro 3.1: Número de preguntas y respuestas de las que se partió para el desarrollo de la colección sobre la cuál realizar el estudio de viabilidad.

Edición	**#preguntas**	**#respuestas por pregunta**	**#runs**	**#pares pregunta-respuesta**
2003	200	3	2	1200
2004	200	1	8	1600
piloto 2004	100	1	1	100
2005	200	1	18	3598
				6498

3.2.1. Construyendo las Hipótesis

Dado que la tarea de RTE está definida entre pares texto-hipótesis, el primer paso para crear la colección consistió en poder generar hipótesis a partir de las preguntas y respuestas de partida.

A la hora de decidir el mecanismo para construir las hipótesis se partió del método empleado en los RTE Challenges para crear las hipótesis de los pares relativos a QA. Según dicho método, cada hipótesis es una combinación de la respuesta junto con la forma afirmativa de la pregunta a la que pretende responder. Para este estudio se decidió transformar manualmente cada pregunta a forma afirmativa, obteniendo por cada pregunta lo que se denomina un *patrón de hipótesis*. Por ejemplo, para la pregunta (1) se obtendría el patrón (2), donde *</answer>* indica el lugar donde se insertará automáticamente cada respuesta dada a esa pregunta con el fin de construir las distintas hipótesis asociadas a dicha pregunta. Por ejemplo, dada la respuesta *Zagreb* a la pregunta (1), se obtendría la hipótesis *La capital de Croacia es Zagreb*. De este modo el proceso de generación de hipótesis es semiautomático.

(1) *¿Cuál es la capital de Croacia?*

(2) *La capital de Croacia es </answer>*

Tras realizar la instanciación automática de los patrones se eliminaron las hipótesis repetidas así como las que se generaron a partir de respuestas NIL. El motivo para eliminar las hipótesis generadas a partir de respuestas NIL fue que éstas pueden ser o no correctas, pero en cualquier caso NIL significa que no hay respuesta, por lo que no hay respuesta que validar.

3.2.2. Extrayendo el Texto Soporte

En el apartado 3.1.2 (página 90) se asume que los sistemas de QA generan junto con cada respuesta un texto soporte. Sin embargo, en las evaluaciones de QA donde se generaron los recursos que se utilizaron para el presente estudio (CLEF

2003, 2004 y 2005), los sistemas participantes tenían que indicar el identificador del documento soporte en lugar de un fragmento de texto soporte. Dado que para la formulación propuesta de la tarea de AV como un problema de RTE es necesario contar con dichos fragmentos, se procedió a identificar los textos soporte automáticamente mediante un procesamiento sencillo que consiguió identificar los fragmentos correctos en un 81 % de los casos. Los pasos principales que se realizaron fueron:

1. Eliminar las palabras vacías de la pregunta.
2. Dividir en oraciones el documento soporte.
3. Tomar la oración que contiene la respuesta y el mayor número de palabras de la pregunta.
4. Si no hay ninguna oración que contenga a la respuesta, tomar la frase que contenga mayor número de palabras de la respuesta.
5. Tomar las siguientes oraciones del documento hasta completar 500 caracteres (tamaño máximo que se permitiría posteriormente para los textos soporte en la tarea de QA del CLEF 2006 (Magnini et al., 2007)).

3.2.3. Determinando el Valor de Implicación

Con el objetivo de comprobar la validez del enfoque propuesto y dado que los juicios humanos sobre las respuestas de los sistemas de QA no son binarios, fue necesario realizar una transformación automática de cada posible juicio de QA a valores binarios de implicación (TRUE o FALSE)[2] de la siguiente manera:

- **Respuesta correcta (R):** en QA este juicio indica que la respuesta a la pregunta es correcta y está soportada por el texto soporte. Por tanto el texto implica a la hipótesis formada por la pregunta y la respuesta y el valor del par es *TRUE*.
- **Respuesta no soportada (U):** en este caso aunque la respuesta podría ser considerada como correcta, el texto soporte no permite afirmar que la respuesta sea correcta. Por tanto el valor de este par debe de ser *FALSE*.
- **Respuesta inexacta (X):** en QA este juicio significa que el texto devuelto como respuesta contiene una respuesta correcta, pero dicho fragmento de texto es demasiado largo o demasiado corto, siendo un caso difícil de evaluar. No se tiene información adicional para decidir si la respuesta contiene demasiada información o si, por el contrario, la respuesta es demasiado corta como para ser considerada correcta. Por tanto no hay un criterio claro para

[2] Se eligió la nomenclatura TRUE/FALSE puesto que era la utilizada en las colecciones del RTE-1 y esta colección estaba inspirada en ellas

decidir si el texto implica o no a la hipótesis sin un análisis humano de cada caso. Dado que hay pocos ejemplos de este tipo, se decidió otorgar a estos pares el valor *UNKNOWN* y descartarlos para la evaluación de la propuesta.

- **Respuesta incorrecta (W):** en este caso la respuesta a la pregunta no es correcta. Aunque se podría dar el caso de que la respuesta estuviese en el texto soporte, la reformulación de la pregunta y la respuesta como una afirmación (la hipótesis) impide la implicación entre el texto y la hipótesis, siendo el valor final del par *FALSE*.

3.2.4. Colección Resultante

En total se generaron 2962 pares texto-hipótesis a partir de 635 preguntas (una media de 4.66 pares por pregunta). El Cuadro 3.2 (página 96) muestra la división en pares con implicación y no implicación. El primer detalle que se observa en el Cuadro es que se obtienen más pares sin implicación que con implicación (un 77 % de pares sin implicación) a diferencia de los RTE Challenge, donde las colecciones están balanceadas (50 % de pares con implicación y 50 % de pares sin implicación). Sin embargo esta colección refleja la salida real de los sistemas actuales de QA, por lo que cabe pensar que un sistema de AV se encontrará con más respuestas incorrectas que correctas. Por tanto estos resultados muestran que la evaluación de sistemas de AV debe de tener en cuenta este factor (como se describirá en el Capítulo 4 de la página 105), centrándose en la detección de respuestas correctas (pares con implicación).

Cuadro 3.2: Número de pares Texto-Hipótesis en SPARTE

	Número	**Porcentaje**
Pares TRUE	695	23 %
Pares FALSE	2267	77 %
Total	2962	

En las Figuras 3.2 (página 97) y 3.3 (página 98) se muestran fragmentos de la colección resultante. Como se puede ver en ambas Figuras, el formato es similar al utilizado en las colecciones del RTE-1 (Dagan et al., 2005).

3.2.5. Evaluación de la Propuesta

Con el objetivo de comprobar la validez de la propuesta se realizó una evaluación de la colección construida. Puesto que como se indicó más arriba la evaluación de sistemas de AV debe centrarse en la detección de respuestas correctas (aspecto sobre el cuál se profundizará en el Capítulo 4 de la página 105), a la hora de comprobar la validez de la propuesta sobre la colección construida se tuvo en cuenta dicho aspecto. Es por ello que se revisó el 100 % de los pares con implicación (695

```
<pair id="1" value="TRUE" task="QA">
   <q>
      ¿Cuál es la capital de Croacia?
   </q>
   <t doc="EFE19940127-14481">"Estoy muy orgulloso de
      ser el primer político yugoslavo que visitará Zagreb
      desde que comenzó la crisis" (en 1991 con el
      estallido de la guerra en Croacia), subrayó Simic,
      quien informó de que Granic viajará a Belgrado entre
      dos y tres semanas después de su visita a la capital
      de Croacia.</t>
   <h>
      La capital de Croacia es Zagreb
   </h>
</pair>
...
<pair id="614" value="TRUE" task="QA">
   <q>
      ¿Qué torneo ganó Andrei Medvedev?
   </q>
   <t doc="EFE19940424-13985">Montecarlo, 24 abr (EFE).-
      El ucraniano Andrei Medvedev destronó al español
      Sergio Bruguera, que había ganado el torneo de
      Montecarlo en 1991 y 1993, al vencerle hoy en la
      final, por 7-5, 6-1 y 6-3. </t>
   <h>
      Andrei Medvedev ganó el torneo de Montecarlo
   </h>
</pair>
```

Figura 3.2: Fragmento de la colección SPARTE con pares donde hay implicación. Dentro de cada hipótesis está marcada la respuesta que la generó.

```
<pair id="26" value="FALSE" task="QA">
    <q>
       ¿Qué país ganó la Copa Davis?
    </q>
    <t doc="EFE19940406-02726">"Para cualquier jugador lo
       más importante es defender a su país en Copa Davis,
       eso nos gusta a todos. Roland Garros me gusta por el
       ambiente que rodea a la competición </t>
    <h>
        Roland Garros ganó la Copa Davis
    </h>
</pair>
...
<pair id="58"value="FALSE" task="QA">
    <q>
       ¿Quién era conocido como el "Zorro del Desierto"?
    </q>
    <t doc="EFE19940205-02731">Se trata del arbitraje
       pedido para el valle austral conocido como Laguna
       del Desierto que tiene una extensión de 523
       kilómetros cuadrados </t>
    <h>
        Laguna del Desierto era conocido como el
       "Zorro del Desierto"
    </h>
</pair>
```

Figura 3.3: Fragmento de la colección SPARTE donde hay pares sin implicación. Dentro de cada hipótesis está marcada la respuesta que la generó.

pares), mientras que de los pares sin implicación se revisó el 5 % (113 pares). El Cuadro 3.3 (página 99) muestra los errores encontrados en la colección generada y en qué porcentaje se detectó cada error. Las principales conclusiones que se obtuvieron al revisar la colección fueron las siguientes:

Cuadro 3.3: Errores encontrados en SPARTE. (i): % pares revisados; (ii): % Errores producidos por juicios incorrectamente realizados en QA; (iii): % Errores producidos por la reformulación de AV en términos de RTE; (iv): % Errores producidos por la extracción automática del texto soporte; (v): % Total de errores

	(i)	**(ii)**	**(iii)**	**(iv)**	**(v)**
Pares TRUE	100 %	6 %	2 %	13 %	21 %
Pares FALSE	5 %	4 %	2 %	-	6 %

- **Errores en el juicio del evaluador de QA.** En algunas ocasiones se observó que algunas respuestas estaban mal evaluadas de origen, con lo que la transformación a valores de implicación daba lugar a pares mal etiquetados. Aunque se esperaba que dichos errores estuviesen dentro del rango de desacuerdo entre anotadores (alrededor del 2 % según Vallin et al. (2005)), fueron algo más numerosos (6 % en los pares TRUE). Dichos errores se debieron principalmente a respuestas no soportadas por el documento indicado como soporte pero que habían sido evaluadas como correctas. Este hecho dio lugar a pares etiquetados como TRUE en los cuáles no había una relación de implicación.

- **Errores debidos a la reformulación de la tarea de AV en términos de RTE.** Se detectaron errores de este tipo en un 2 % de los pares al haber respuestas incorrectas que generaban pares con implicación, o respuestas correctas que generaban pares donde no había implicación. Aunque es una proporción muy pequeña (similar al desacuerdo entre anotadores en QA), hay que prestar atención a estos errores puesto que afectan a la definición de la propuesta. Respecto a respuestas correctas que generaron pares sin implicación, estos casos fueron principalmente el resultado de los cambios en sintaxis y semántica que introdujeron respuestas con más información de la adicional. Es decir, respuestas que deberían de haber sido evaluadas como inexactas, por lo que una evaluación más estricta eliminaría este problema. Un ejemplo de estos casos se puede ver en la Figura 3.4 (página 101). En esta Figura está resaltada para cada hipótesis la respuesta que dio lugar a dicha hipótesis. En cuanto a respuestas incorrectas que generaron pares con implicación, la Figura 3.5 (página 102) muestra algunos ejemplos de errores encontrados a este respecto. En los ejemplos de la Figura se puede comprobar cómo la respuesta devuelta a la pregunta no se puede considerar como

correcta, pero sin embargo la hipótesis a la que da lugar está implicada por el texto soporte.

- **Errores en la identificación automática del texto soporte.** Puesto que por cada par el texto tuvo que ser extraído automáticamente a partir del documento soporte de cada respuesta, se decidió revisar en los pares estudiados el resultado de dicha extracción automática. Se encontró que en el 100 % de los casos bastaba con una oración del documento soporte para soportar la respuesta y por tanto implicar a la hipótesis. También se detectó que el proceso automático para extraer las oraciones había fallado en un 13 % de los pares con implicación (hay que tener en cuenta que en los pares sin implicación no se puede obtener ningún texto soporte correcto, puesto que no hay ninguna respuesta correcta que soportar), dando en dichos casos un fragmento que realmente no soportaba la veracidad de la respuesta. Sin embargo estos errores carecen de importancia puesto que la extracción automática se realizó solamente para el estudio, ya que el modelo asume que para cada respuesta se devuelve un fragmento de texto como soporte y el juicio que se otorga a la respuesta tiene en cuenta dicho fragmento de texto.

Tras la construcción de la colección y su estudio se comprobó que a pesar de los errores menores que se encontraron el enfoque propuesto era válido. De hecho, algunos de los problemas detectados se debieron a fallos en los juicios humanos que se tomaron como referencia o a la extracción automática de textos soporte que se realizó. Teniendo en cuenta que dichos errores no se van a producir en el modelo de validación propuesto (dado que los juicios humanos son sólo una referencia y los fragmentos de texto serán los devueltos por los sistemas y sobre los que se tiene que comprobar la veracidad de la respuesta), la conclusión general fue que la propuesta suponía un enfoque apropiado y que con ella sería posible mejorar los resultados en AV respecto a los métodos que realizan un análisis menos profundo de las preguntas y las respuestas.

3.3. Recapitulación

La mayoría de los métodos planteados para realizar AV antes de la propuesta de este trabajo se han basado en medir la relevancia de una respuesta respecto a una pregunta en términos de conteo de redundancias, sin realizarse un análisis de las relaciones semánticas existentes entre pregunta y respuesta. Debido a que la tarea de RTE analiza las relaciones semánticas entre dos fragmentos de texto (llamados texto e hipótesis), en este capítulo se ha propuesto un modelo de AV basado en RTE. De este modo, este modelo propone que los sistemas de AV hagan uso de métodos más complejos que los utilizados anteriormente.

Para comprobar la viabilidad del modelo se decidió construir una colección de pares texto-hipótesis, típicos de una tarea de RTE, orientados a la Validación de

```
<pair id="167" value="TRUE" task="QA">
    <q>
       ¿Qué iglesia aprobó los nuevos cánones para la
       ordenación de mujeres?
    </q>
    <t doc="EFE19941202-00867">El Sínodo de la Iglesia
       Anglicana aprueba en Londres los nuevos cánones
       para la ordenación de mujeres </t>
    <h>
       La iglesia Sínodo de la Iglesia Anglicana aprobó los
       nuevos cánones para la ordenación de mujeres
    </h>
</pair>
...
<pair id="9" value="TRUE" task="QA">
    <q>
       ¿Cuál es el nombre de pila del juez Borsellino?
    </q>
    <t doc="EFE19940718-10595">El juez Paolo Borsellino
       fue asesinado junto con su escolta dos meses después
       de que murieran, su colega Giovanni Falcone, su
       esposa Francesca Morbillo y la escolta</t>
    <h>
       Paolo Borsellino es el nombre de pila del
       juez Borsellino
    </h>
</pair>
```

Figura 3.4: Ejemplos de respuestas correctas (que deberían de haber sido inexactas) y que tras la reformulación propuesta se convierten en pares sin implicación. Dentro de cada hipótesis está marcada la respuesta que la generó.

```
<pair id="705" value="FALSE" task="QA">
   <q>
      ¿Dónde comenzaron las excavaciones británicas para la
      construcción del Eurotúnel?
   </q>
   <t doc="EFE19940503-01053">La razón de que en Kent
      (sureste de Inglaterra), precisamente por donde el
      "Eurotúnel" abre su boca en suelo británico, los
      campanarios ... </t>
   <h>
      En Inglaterra comenzaron las excavaciones británicas
      para la construcción del Eurotúnel
   </h>
</pair>
...
<pair id="1037" value="FALSE" task="QA">
   <q>
      ¿Qué submarino chocó con un buque en el Canal de la
      Mancha el 16 de febrero de 1995?
   </q>
   <t doc="EFE19950125-14055">Londres, 25 ene (EFE).- La
      fragata británica Battleaxe y el submarino alemán
      U-14 chocaron hoy, miércoles, en el Canal de la
      Mancha</t>
   <h>
      el submarino alemán chocó con un buque en el Canal
      de la Mancha el 16 de febrero de 1995
   </h>
</pair>
```

Figura 3.5: Ejemplos de respuestas incorrectas que tras la reformulación propuesta se convierten en pares con implicación. Dentro de cada hipótesis está marcada la respuesta que la generó.

Respuestas y realizar el estudio sobre dicha colección. Con este propósito se examinaron los pares que se generaron y se observó la validez del enfoque propuesto (a pesar de haber ciertos errores, la mayoría de los cuáles se debían a factores externos a la propuesta realizada, como por ejemplo la utilización de juicios humanos mal realizados).

La validez de esta propuesta supone un primer paso para el desarrollo de la metodología de evaluación de sistemas de AV realizada en el capítulo 5 (página 125), la cuál parte del modelo definido en este capítulo.

Capítulo 4

Medidas de Evaluación en Validación de Respuestas

Este capítulo se centra en nuevas métricas para evaluar sistemas de Validación de Respuestas. La motivación surge del estudio realizado sobre las evaluaciones realizadas previamente sobre sistemas de AV (descritas en la sección 2.4.2 de la página 63), al detectar, por una parte, que no se trató el impacto que puede suponer la utilización de módulos de AV dentro de un sistema de QA, y que tampoco se habían evaluado los sistemas de AV que realizan selección de respuestas. Es por ello que en este capítulo se proponen una serie de medidas con el objetivo de cubrir estos dos aspectos.

En este capítulo se diferencian dos tipos de evaluación de acuerdo con las dos funciones que puede desarrollar un sistema de AV dentro de uno de QA (funciones que se describieron en la sección 2.4.1 de la página 58): validación y selección. A lo largo del capítulo se proponen y analizan distintas medidas y "baselines" para evaluar cada una de estas dos funciones.

Finalmente, se realiza un estudio comparando las medidas propuestas para la evaluación de sistemas de AV que realizan validación con otras medidas utilizadas en aprendizaje automático, que podrían ser utilizadas también para evaluar sistemas de AV. En el estudio se analiza la confianza que se puede depositar en las medidas comparadas, así como su adecuación a la tarea de AV, observándose la idoneidad de las medidas propuestas en este capítulo.

4.1. Medidas para Evaluar la Validación de Respuestas

La función de un sistema de AV que realiza validación es eliminar las respuestas incorrectas de entre las respuestas candidatas generadas para una determinada pregunta. La decisión acerca de la corrección de las respuestas es un problema de clasificación binaria donde las respuestas se clasifican como correctas o incorrectas. Esta tarea de clasificación binaria tiene la matriz de confusión mostrada en el Cuadro 4.1 de la página 106 (las fórmulas del resto del capítulo serán dadas en

términos de dicha matriz).

Cuadro 4.1: Matriz de contingencia para la validación de respuestas.

	Respuesta correcta	**Respuesta incorrecta**
Respuesta validada	n_{cv}	n_{iv}
Respuesta rechazada	n_{cr}	n_{ir}

En aprendizaje automático se utiliza *accuracy* (Fórmula (4.1)) como método principal para evaluar el rendimiento de sistemas de clasificación binaria. Mediante el uso de *accuracy*, la detección de respuestas correctas e incorrectas es recompensada en la misma proporción. Sin embargo, la tarea de AV tiene una característica que hay que tener en cuenta a la hora de realizar la evaluación: los sistemas de QA producen como salida una cantidad distinta de respuestas correctas e incorrectas, dando lugar a colecciones de evaluación no balanceadas (como se pudo comprobar en la sección 3.2.4 de la página 96). Por ejemplo, a partir de la salida de sistemas reales de QA se podría generar una colección de evaluación donde más del 80 % de las respuestas serían incorrectas (de hecho, constituye un ejemplo real de las colecciones que se verán en la sección 5.4 de la página 134). Un sencillo sistema de AV que decidiese clasificar como incorrectas a todas las preguntas de esta colección obtendría un valor alto de *accuracy* (más del 80 %), pero sin embargo no sería realmente útil para realizar validación. Esta situación se debe a que el uso de *accuracy* asume que la distribución de las clases es constante y relativamente balanceada, lo cuál no ocurre siempre en AV.

$$accuracy = \frac{n_{cv} + n_{ir}}{n_{cv} + n_{iv} + n_{cr} + n_{ir}} \tag{4.1}$$

Teniendo en cuenta la naturaleza no balanceada de las colecciones, se puede considerar que un sistema de AV que realiza validación se centra en detectar las respuestas correctas de entre un conjunto de respuestas candidatas. Por este motivo, se propone centrar la evaluación en la detección de respuestas correctas, midiendo la *precisión* y la *cobertura* sobre las respuestas correctas:

- **Precisión:** proporción de respuestas validadas que son realmente correctas (ver Fórmula (4.2)). Esta medida se centra en evaluar la habilidad del sistema prediciendo que una respuesta es correcta, premiando a los sistemas que sólo validan respuestas correctas.

- **Cobertura:** proporción de respuestas correctas de la colección que han sido validadas por el sistema (ver Fórmula (4.3)). Esta medida evalúa la capacidad de un sistema para detectar todas las respuestas correctas de entre las candidatas que recibe sin tener en cuenta la precisión con la cuál se realiza esta detección. De este modo, los sistemas que detectan todas las respuestas correctas recibirán el máximo valor de *cobertura*.

$$precisión = \frac{n_{cv}}{n_{cv} + n_{iv}} \tag{4.2}$$

$$cobertura = \frac{n_{cv}}{n_{cv} + n_{cr}} \tag{4.3}$$

Uno de los inconvenientes que tienen *precisión* y *cobertura* es que suelen variar de forma inversa entre si. Es decir, cuando se logran incrementar los resultados de una, normalmente se disminuyen los de la otra. Es por ello que se suelen combinar en una única medida que resume el resultado de ambas. Como se indicó en la sección 2.1.2 (página 36), la medida más utilizada para combinar *precisión* y *cobertura* es la *medida F* (Fórmula (2.6) de la página 38). En dicha Fórmula valores de β menores que 1 dan más importancia a la *precisión* sobre la *cobertura*, mientras que el comportamiento contrario se consigue con valores de β mayores que 1. La elección del valor de β a utilizar depende de los objetivos de la evaluación y es por ello que, al igual que generalmente en IR, en este trabajo se decidió utilizar el valor β = 1 para así otorgar la misma importancia a la *precisión* y a la *cobertura*, resultando en la Fórmula (2.7) de la página 38.

Por otro lado, al hacer uso de *precisión*, *cobertura* y *F* sobre una colección de evaluación formada por un conjunto de preguntas y respuestas se pueden seguir dos enfoques distintos:

1. Aplicar las medidas sobre el conjunto total de respuestas. Es decir, calcular *precisión* y *cobertura* sobre todas las respuestas de la colección y a continuación calcular *F*. A este enfoque se le denomina micro-media (más conocido por su nombre en inglés, *micro-average*).

2. Calcular *precisión*, *cobertura* y *F* por cada conjunto de respuestas pertenecientes a una misma pregunta y a continuación calcular la media de *F* sobre todas las preguntas. A este enfoque se le denomina macro-media (más conocido por su nombre en inglés, *macro-average*).

Cada uno de estos enfoques tiene sus ventajas y sus inconvenientes y la decisión sobre cuál de los dos elegir depende de los objetivos de la evaluación. Si se quiere evaluar el rendimiento de los sistemas de AV validando respuestas por cada pregunta, o lo que es lo mismo, evaluar la detección de respuestas correctas por pregunta, entonces se debe seguir el enfoque número 2. Sin embargo, este enfoque presenta los siguientes inconvenientes:

- En un escenario real puede haber preguntas que no tienen ninguna respuesta correcta ya sea por una incorrecta formulación de la pregunta (como por ejemplo *¿Quién es el portero de la selección española de baloncesto?*), o por no encontrarse la solución en la fuente en la cuál se realiza la búsqueda. Al evaluar sistemas de AV sobre respuestas a estas preguntas no tiene sentido hablar de *la proporción de respuestas correctas detectadas*, ya que no existe

ninguna respuesta correcta que pueda ser detectada. Esto significa que no se puede calcular la *cobertura* en estas preguntas y como consecuencia tampoco se puede calcular el valor de *F* sobre estas preguntas.

- Por otro lado, el número de respuestas por pregunta que valida un sistema de AV es variable. Esto significa que hay preguntas para las cuáles un sistema de AV no valida ninguna respuesta (o lo que es lo mismo, el sistema rechaza todas las respuestas a esa pregunta). En dichas preguntas no es razonable calcular *la proporción de respuestas dadas como correctas por el sistema*, es decir, la *precisión*, ya que ninguna respuesta ha sido dada como correcta. Por este motivo, tampoco se puede calcular *F* en estas preguntas.

Dados los inconvenientes citados, se considera que el segundo enfoque es más problemático de aplicar y menos informativo para los objetivos de este trabajo que el primer enfoque (ya que la media final no sería sobre todas las preguntas al no poderse calcular *F* en algunas de ellas). Por este motivo se propone realizar el cálculo de *precisión*, *cobertura* y *F* sobre el conjunto total de respuestas (calculo de micro-average).

Hay que tener en cuenta que los valores obtenidos a partir de estas medidas (*precisión*, *cobertura* y *F*) dependen del número de respuestas correctas contenidas en una colección. Por tanto, los resultados obtenidos sobre colecciones con distribuciones distintas de respuestas correctas e incorrectas no se pueden comparar entre si. Sin embargo, se puede realizar la comparación con los siguientes "baselines":

- Un sistema que valida todas las respuestas y que por tanto siempre obtiene el valor 1 de cobertura.

- Un sistema que valida aleatoriamente la mitad de las respuestas.

El comportamiento del "baseline" que valida todas las respuestas se correspondería con el de un sistema de QA que no realiza una etapa de validación. Es decir, el comportamiento sería como el de un sistema de QA que devuelve todas las respuestas candidatas sin comprobar su validez. Por tanto, el hecho de que un sistema de AV obtuviese mejores resultados que este "baseline" sobre la misma colección indicaría que el sistema de AV propuesto es capaz de mejorar los resultados de un sistema de QA que no realiza validación (y cuya salida se corresponde con la colección de evaluación), lo cual supondría una motivación para el uso de este sistema de AV dentro de uno de QA.

4.2. Medidas para Evaluar la Selección de Respuestas

Un sistema de AV que realiza selección tiene como objetivo seleccionar para cada pregunta una respuesta correcta de entre el conjunto de respuestas candidatas

obtenidas para la pregunta (como ya se indicó en la sección 2.4.1.2 de la página 59). De este modo, por cada pregunta hay dos posibles comportamientos:

- Sólo se selecciona una respuesta.
- Todas las respuestas son consideradas incorrectas y no se selecciona ninguna de ellas, dejando sin responder a la pregunta.

Este comportamiento es comparable al de un sistema de QA que como máximo genera una respuesta por pregunta, ya que para cada pregunta no hay más de una respuesta. La matriz de contingencia asociada a este comportamiento se muestra en el Cuadro 4.2 (página 109). Además, para las fórmulas de las medidas de evaluación propuestas en esta sección hay que tener en cuenta que el número total de preguntas es $n = n_{ca} + n_{wa} + n_{ws} + n_{wr} + n_{cr}$

Cuadro 4.2: Matriz de contingencia para la selección de respuestas

	Preguntas CON respuesta correcta	**Preguntas SIN respuesta correcta**
Preguntas respondidas correctamente (una respuesta seleccionada)	n_{ca}	-
Preguntas respondidas incorrectamente	n_{wa}	n_{ws}
Preguntas sin responder (se rechazan todas las respuestas)	n_{wr}	n_{cr}

En este contexto se puede evaluar tanto la capacidad del sistema realizando selección, como su capacidad detectando preguntas sin ninguna respuesta correcta. Además, también es interesante estimar el rendimiento que se obtendría si se considera que se van a pedir más respuestas al sistema de QA en caso de que se detecten preguntas sin respuestas correctas. Los siguientes subapartados se centran en estudiar la evaluación de cada uno de estos tres aspectos.

4.2.1. Evaluando la Selección Correcta

La medida que se propone en esta tesis para evaluar la selección de respuestas es *qa_accuracy* (Fórmula (4.4)), la cuál mide la proporción de respuestas correctamente seleccionadas. Esta proporción es una medida comparable al *accuracy* utilizado en QA, lo cuál permite comparar los resultados de un sistema tradicional de QA con los que se obtendrían al incorporar un módulo de AV que realizase selección tanto en un sistema individual de QA, como en uno multi-flujo. Por ejemplo, se podrían comparar los resultados de sistemas individuales de QA (evaluados

con *accuracy*) con los de un sistema de AV que realizase selección de entre el conjunto de respuestas generadas por dichos sistemas de QA (evaluado utilizando *qa_accuracy*).

qa_accuracy tiene un límite superior dado por la proporción de preguntas que tienen al menos una respuesta candidata correcta. Este límite superior corresponde a una selección perfecta de las respuestas candidatas. Al valor normalizado de *qa_accuracy* respecto a este límite superior se le denomina en esta tesis *normalized_qa_accuracy* (Fórmula (4.5)).

Además de con este límite superior, los resultados obtenidos con *qa_accuracy* se pueden comparar con el siguiente "baseline": un sistema que considera correctas al 100 % de las respuestas candidatas y por cada pregunta selecciona aleatoriamente una respuesta. Este "baseline" puede ser visto como la proporción media de respuestas correctas por cada pregunta y en esta tesis se le denomina *random_qa_accuracy* (Fórmula (4.6)).

$$qa\_accuracy = \frac{n_{ca}}{n} \tag{4.4}$$

$$normalized\_qa\_accuracy = \frac{n_{ca}}{n_{ca} + n_{wa} + n_{wr}} \tag{4.5}$$

$$random\_qa\_accuracy = \frac{1}{n} \sum_{q \epsilon preguntas} \frac{\#respuestas\ correctas\ de\ (q)}{\#respuestas\ de(q)} \tag{4.6}$$

4.2.2. Evaluando la Detección de Preguntas sin Respuestas Correctas

El principal inconveniente de *qa_accuracy* es que sólo premia la habilidad de un sistema de AV seleccionando respuestas correctas y no su capacidad para detectar que todas las respuestas a una determinada pregunta son incorrectas. El motivo para premiar este comportamiento tiene su fundamento en el hecho de que una posible mejora en los resultados de QA se podría obtener teniendo en cuenta esta capacidad. Esto se debe a que en el caso de detectar las preguntas sin respuestas correctas, un sistema de AV podría solicitar a los sistemas de QA nuevas respuestas y abrir así la posibilidad de obtener respuestas correctas para estas preguntas.

Cuando un sistema de AV considera que no hay ninguna respuesta correcta para una pregunta se dice que el sistema de AV rechaza contestar a la pregunta. Para evaluar la precisión de un sistema de AV rechazando preguntas, en esta tesis se propone el uso de *qa_rej_accuracy* (Fórmula (4.7)). *qa_rej_accuracy* cuantifica la proporción de preguntas que han sido rechazadas correctamente sobre el total de preguntas. Es decir, la proporción de preguntas para las cuáles no había ninguna respuesta correcta y el sistema de AV lo ha detectado.

$$qa\_rej\_accuracy = \frac{n_{cr}}{n} \tag{4.7}$$

4.2.3. Evaluando el Rendimiento Potencial

Usando *qa_accuracy* y *qa_rej_accuracy* se define *qa_accuracy_max* (Fórmula (4.8)). *qa_accuracy_max* puede ser vista como el tradicional *accuracy* usado en clasificación binaria (porcentaje de ejemplos positivos y negativos detectados correctamente), ya que recoge tanto las preguntas donde se selecciona una respuesta correcta (análogo a la detección de ejemplos positivos en clasificación), como las preguntas donde se detecta que no hay respuestas correctas (análogo a la detección de ejemplos negativos en clasificación).

Sin embargo, en *qa_accuracy_max* se da demasiado valor a las preguntas rechazadas correctamente. De hecho, la interpretación dentro del contexto de QA sería que una vez detectadas las preguntas sin respuestas correctas, todas estas preguntas serán respondidas correctamente en un segundo ciclo. Asegurar que se encontrarán respuestas correctas para todas estas preguntas es una estimación demasiado optimista.

Para estimar la cantidad de preguntas correctamente rechazadas que se podrían responder correctamente en un segundo ciclo, en esta tesis se ha decidido utilizar el valor observado de *qa_accuracy*. Es decir, la precisión observada anteriormente seleccionando respuestas correctas. La interpretación de esta estimación dentro del contexto de QA sería que la preguntas correctamente rechazadas se responderán correctamente en la proporción dada por *qa_accuracy* (que representa el acierto que se tuvo anteriormente seleccionando respuestas correctas). Al hacer uso de esta estimación se obtiene *estimated_qa_performance* (Fórmula (4.9)). Esta medida es una estimación del rendimiento potencial que se puede obtener en QA utilizando un módulo de AV que realiza selección, y que además es capaz de detectar preguntas sin ninguna respuesta correcta. El límite superior de este rendimiento potencial sería el indicado por *qa_accuracy_max*.

$$qa\_accuracy\_max = qa\_accuracy + qa\_rej\_accuracy \tag{4.8}$$

$$estimated\_qa\_performance = qa\_accuracy + qa\_rej\_accuracy * qa\_accuracy \tag{4.9}$$

4.3. Evaluación de las Medidas Propuestas

Como se mencionó en la sección 4.1 (página 105), en clasificación binaria con colecciones no balanceadas, lo cuál ocurre en AV, no es recomendable utilizar *accuracy* como medida de evaluación. La propuesta que se realizó entonces fue medir la *precisión*, *cobertura* y su media armónica (*medida F*) sobre las respuestas correctas. Sin embargo, una de las principales críticas que reciben las evaluaciones realizadas con *precisión*, *cobertura* y *F* es que los valores obtenidos varían al cambiar la distribución de las clases (proporción de ejemplos positivos y negativos),

incluso aunque no haya cambios significativos en el rendimiento del sistema a evaluar (Provost and Fawcett, 2001). Este es uno de los argumentos utilizados para realizar la evaluación sobre conjuntos no balanceados haciendo uso del análisis del espacio Receiver Operating Characteristic (ROC), cuyo análisis puede ser resumido en un valor escalar denominado área bajo la curva (en inglés Area Under the Curve, *AUC*).

En esta sección se realiza un estudio comparativo entre *AUC* y *F* para evaluar sistemas de AV que realizan validación. Para ello se comparan ambas medidas respecto a diversos aspectos. En concreto, el trabajo realizado en esta sección se centra en:

- Comparar *F* y *AUC* para evaluar la validación de respuestas y tratar de determinar las situaciones en las cuáles es más adecuado utilizar cada una de estas medidas.
- Comparar la confianza que se puede depositar en los resultados obtenidos mediante *F* y *AUC*. Para ello, este estudio se centra en la estabilidad y el poder de discriminación de ambas medidas.

Este estudio aporta un mayor entendimiento sobre las dos opciones para evaluar la validación de respuestas, de modo que se pueda elegir la medida que se considere más apropiada. Para ello, primero se exponen los fundamentos del análisis ROC (los detalles de *precisión*, *cobertura* y *F* fueron expuestos en la sección 4.1 de la página 105). A continuación se comparan ambas medidas de acuerdo a su poder de discriminación, su estabilidad y su adecuación a la evaluación de la AV. Finalmente y a partir de lo observado en los experimentos, se discute brevemente sobre el uso de la *medida F*.

4.3.1. Análisis ROC

El análisis ROC (Receiver Operating Characteristic) es una metodología comúnmente utilizada en diagnóstico médico y que se usa también para evaluar clasificadores en inteligencia artificial (Beck and Shultz, 1986; Friedman and Wyatt, 1997). En problemas de clasificación binaria el espacio ROC es una representación en dos dimensiones con la proporción de ejemplos positivos detectados (en inglés true positive rate, abreviado *tp rate* y mostrado en la Fórmula (4.10) utilizando la nomenclatura de la matriz de confusión del Cuadro 4.1 de la página 106) en el eje Y, y la proporción de falsos positivos (en inglés false positive rate, abreviado *fp rate* y mostrado en la Fórmula (4.11)) en el eje X. A *tp rate* se le conoce también como sensibilidad dentro del contexto del análisis ROC y además se puede ver que *tp rate* representa el mismo valor que *cobertura* (Fórmula (4.3) de la página 107). De este modo, cada matriz de confusión genera un punto en el espacio ROC.

$$tp\,rate = \frac{n_{cv}}{n_{cv} + n_{cr}} \tag{4.10}$$

$$fp\,rate = \frac{n_{iv}}{n_{iv} + n_{ir}} \tag{4.11}$$

Son varios los puntos a destacar en el espacio ROC (ver Figura 4.1 de la página 113). El punto (0,0) representa a un clasificador que predice todas las instancias como negativas, mientras que el punto (1,1) corresponde a un clasificador que predice todas las instancias como positivas. Por otro lado, el punto (0,1) representa una clasificación perfecta mientras que el punto (1,0) corresponde al clasificador que falla al realizar todas sus predicciones. Por último, la línea diagonal *y=x* representa la estrategia de clasificar aleatoriamente, por lo que se puede decir que cualquier clasificador que se encuentre en dicha diagonal no tiene información acerca del problema de clasificación abordado. Por tanto, cualquier clasificador que aparezca por debajo de dicha diagonal tiene un rendimiento peor que la clasificación aleatoria. Lo deseable es tener siempre un clasificador que se encuentre por encima de dicha diagonal.

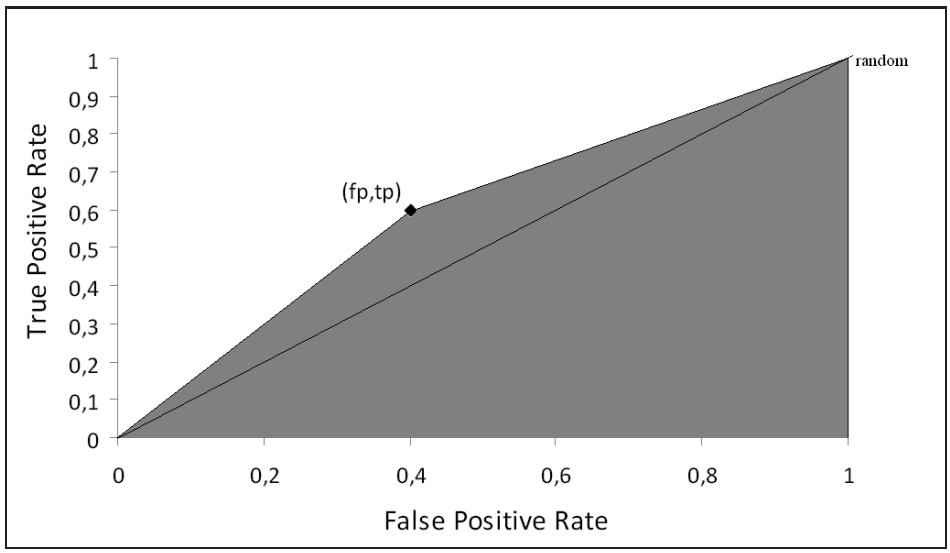

Figura 4.1: Área bajo la curva (AUC) del punto (0.4 , 0.6)

Además de esta representación gráfica, a partir del concepto de curva ROC se puede obtener un valor escalar que sirve como indicador del rendimiento de un clasificador. La curva ROC de un clasificador está formada por una secuencia de puntos ROC del clasificador, incluyendo los puntos (0,0) y (1,1), conectados por segmentos. Las curvas ROC tienen la propiedad de no verse afectadas por cambios en la distribución de las clases, lo cuál es una ventaja para evaluar sistemas de AV.

El método usado para generar la secuencia de puntos ROC depende del clasificador. Por ejemplo, en algunos clasificadores se obtiene la secuencia de puntos ROC variando el valor umbral utilizado para decidir cuándo clasificar una instancia como positiva o como negativa. En caso de que no haya ningún método para generar la secuencia de puntos, un clasificador puede formar una curva ROC conectan-

do el único punto ROC que genera, junto con los puntos (0,0) y (1,1) (Drummond and Holte, 2004). Este es el método que se propone en este trabajo para construir curvas ROC que sirvan para evaluar a sistemas de AV (vistos como clasificadores binarios). La Figura 4.1 (página 113) muestra un ejemplo de una curva ROC siguiendo este enfoque.

Dada una curva ROC se puede utilizar el área bajo la curva (en inglés Area Under the ROC Curve, *AUC*) como indicador del rendimiento del clasificador (Bradley, 1997; Hanley and McNeil, 1982). Dado que *AUC* es una porción del área del cuadrado unidad, su valor está comprendido entre 0 y 1. Como la estrategia de clasificar aleatoriamente produce una diagonal entre los puntos (0,0) y (1,1), a la cuál le corresponde un área de 0.5, es deseable tener clasificadores cuyo valor de *AUC* sea superior a 0.5.

Los siguientes apartados muestran una comparación de *AUC* con *F* para estudiar su adecuación a la evaluación de la validación de respuestas.

4.3.2. Datos utilizados para el Análisis

Para realizar la comparación de las dos medidas de evaluación se hizo uso de los datos del Answer Validation Exercise[1] (AVE) 2008 del CLEF (los cuáles están descritos con más detalle en la sección 5.4 de la página 134).

Estas colecciones fueron creadas a partir de la salida real de sistemas de QA y están enfocadas a la evaluación de sistemas de AV. Para ello, las colecciones contienen un conjunto de pares *{Respuesta, Texto Soporte}* agrupados por *Pregunta*. Los sistemas participantes en la tarea tenían que tener en cuenta cada *Pregunta* y clasificar cada uno de los pares *{Respuesta, Texto Soporte}* de dicha *Pregunta* como correcto o incorrecto. Es decir, validar o rechazar cada *Respuesta* a una *Pregunta*.

El número de respuestas por idioma así como la distribución en correctas o incorrectas depende de la salida de los sistemas de QA a partir de los cuáles se generan las colecciones. En los Cuadros 5.10 (página 141) y 5.11 (página 141) se puede ver la cantidad de preguntas y respuestas (con su distribución en correctas e incorrectas) por idioma en las colecciones de evaluación del AVE 2008. El número de runs participantes por cada idioma se muestra en el Cuadro 5.18 (página 147).

Para realizar el estudio, se decidió utilizar los datos de inglés debido a que fue el idioma que contó con el mayor número de runs participantes y con la segunda colección de evaluación de mayor tamaño (el tamaño de las colecciones se mide en términos del número de respuestas). Dado que los métodos que se utilizan en este capítulo para comprobar la confianza que se puede depositar en las medidas se basan en realizar un gran número de comparaciones entre distintos runs, se decidió utilizar los datos de inglés debido a que un mayor número de runs permite realizar más comparaciones y tener una mayor confianza en los resultados obtenidos.

[1] http://nlp.uned.es/clef-qa/ave/

4.3.3. Poder de Discriminación y Estabilidad

Para comparar la estabilidad y el poder de discriminación de *F* y *AUC* se utilizó el método de Buckley and Voorhees (2000), el cuál fue descrito en la sección 2.6.2 (página 81). Hay que recordar que la estabilidad hace referencia al error asociado a la conclusión *el sistema X es mejor que el sistema Y*, de modo que cuanto más estable es una medida menor es el error.

Además, el método de Buckley and Voorhees (2000) permite calcular también el poder de discriminación de una medida, de modo que cuanto más discriminativa es la medida menos empates habrá entre sistemas y menor será la diferencia de resultados requerida para concluir qué sistema es mejor.

Para realizar los experimentos de esta sección se utilizó el algoritmo de la Figura 4.2 (página 115) para obtener los datos necesarios para calcular la tasa de error (Fórmula (2.28) de la página 83). Esta tasa se utilizó para comprobar la estabilidad (a menor tasa de error más estable es una medida de evaluación). Los datos obtenidos por el algoritmo permiten calcular también la proporción de empates (Fórmula (2.29) de la página 83), que sirve para evaluar el poder de discriminación de una medida (cuanto menor es la proporción de empates mayor es el poder de discriminación).

```
por cada par de runs x,y ϵ S
  por cada ejecución desde 1 hasta 500
    Q_i = seleccionar aleatoriamente una subcol de tamaño c a partir de Q;
    margen = umbral * max (M(x,Q_i),M(y,Q_i));
    si |M(x,Q_i) - M(y,Q_i)| < |margen| entonces
      |x == y| + +;
    sino
      si |M(x,Q_i) > M(y,Q_i)| entonces
        |x > y| + +;
      sino
        |y > x| + +;
```

Figura 4.2: Algoritmo para realizar el cálculo de $|x > y|$, $|y > x|$ y $|x == y|$ para calcular la tasa de error y la proporción de empates de una determinada medida de evaluación M de acuerdo con el método de estabilidad de Buckley and Voorhees (2000)

En el algoritmo de la Figura 4.2, *S* representa un conjunto de runs, mientras que *x* e *y* son runs pertenecientes a *S*. La colección de evaluación (conjunto de preguntas y respuestas) de la que se parte se representa por *Q*, y al umbral de equivalencia (diferencia en porcentaje entre los resultados de dos sistemas de modo que si la diferencia es menor que este umbral se considera que ambos resultados son equivalentes y hay un empate entre los sistemas) se le denomina *umbral*. Finalmente, M(x,Q_i) representa el resultado del run *x* sobre la subcolección Q_i de acuerdo con

la medida de evaluación M.

En cuanto a la elección del umbral de equivalencia, está claro que el hecho de utilizar valores altos implica tener valores más bajos de la tasa de error. Es por ello que para realizar un estudio más completo se decidió variar el umbral de equivalencia desde 0.01 hasta 0.10 (siguiendo el trabajo realizado por Sakai (2007b)), dibujando para cada medida de evaluación una curva *proporción de empates - tasa de error*.

La Figura 4.3 (página 116) muestra las curvas *proporción de empates - tasa de error* de *F* y *AUC* calculadas utilizando las colecciones de evaluación y los 8 runs participantes en inglés en el AVE 2008 con c=500 respuestas[2]. En la Figura 4.3 se puede ver cómo la tasa de error de ambas medidas disminuye al aumentar la proporción de empates, lo cuál corresponde a incrementos en el valor del umbral de equivalencia. Además, ambas medidas obtienen una tasa de error por debajo del 5 %, lo cuál indica que son medidas muy estables (el 5 % es uno de los valores de referencia más utilizados para comprobar la confianza en unos resultados, tanto en los métodos de estabilidad (Sakai, 2007b), como en los de significancia estadística (Keen, 1992)).

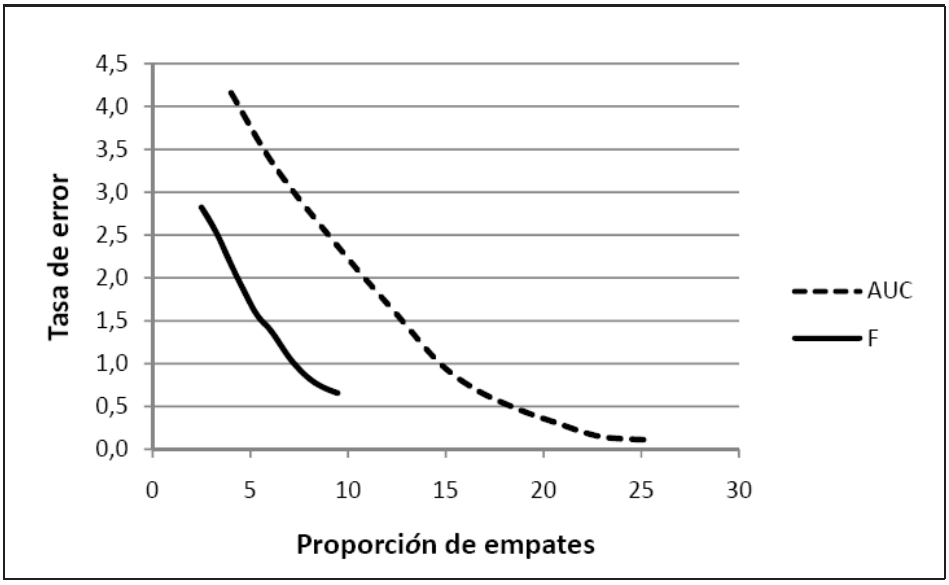

Figura 4.3: Curvas *proporción de empates* / *tasa de error* para *F* y *AUC* utilizando los runs y colecciones del AVE 2008 con c=500

Se pueden dar dos interpretaciones distintas de las curvas mostradas en la Figura 4.3:

1. Si se fija a un determinado valor la proporción de empates, la tasa de error que se obtiene para *F* es menor que la de *AUC*. Por ejemplo, si la proporción

[2]La decisión de utilizar este tamaño para las subcolecciones de los experimentos se debe a mantener el mismo tamaño que en los experimentos que se muestran en la sección 4.3.4 (página 117)

de empates es del 5 %, la tasa de error de *F* es del 2 % y la de *AUC* es del 4 %. Esto indica que dada la misma proporción de empates, *F* es más estable que *AUC*, lo cuál significa que se tendrán menos errores al concluir qué sistema es mejor cuando las comparaciones se realicen con *F* que cuando se realicen con *AUC*.

2. Si se fija la tasa de error a un determinado valor, la proporción de empates que se obtiene para *AUC* es mayor que la que se obtiene para *F*. Por ejemplo, dada una tasa de error del 2 %, el porcentaje de empates para *F* es del 5 %, mientras que para *AUC* es aproximadamente del 11 %. Esto significa que si se quiere estar seguro de que un determinado sistema *A* es mejor que otro sistema *B* asumiendo una determinada tasa de error, se tendrán menos empates entre sistemas y se necesitará un umbral de equivalencia más pequeño (una menor diferencia de resultados entre sistemas) utilizando *F* que utilizando *AUC*.

Por tanto, *F* ha mostrado en los experimentos realizados en esta sección que es más estable y tiene un mayor poder de discriminación que *AUC*. Esto no significa que se deba de utilizar *F* en lugar de *AUC* para evaluar sistemas de AV, sino que hay que tener más cuidado a la hora de obtener conclusiones sobre los resultados obtenidos utilizando *AUC* que cuando se utiliza *F*.

4.3.4. Sensibilidad

Otra forma de evaluar la confianza que se puede depositar en los resultados obtenidos por una medida de evaluación es el método descrito en Voorhees and Buckley (2002), denominado método de swap y que fue descrito en la sección 2.6.2.2 (página 84). La idea principal de este método consiste en contar el número de veces en las cuáles dos runs difieren en cuanto a qué sistema es mejor, condicionado al tamaño de la diferencia de los resultados de los dos sistemas.

Para aplicar este método se utilizó el algoritmo de la Figura 4.4 (página 118), el cuál es una adaptación del algoritmo de la Figura 2.11 (página 85) para el estudio realizado en esta sección. El algoritmo de la Figura 4.4 permite calcular la tasa de error de cada ranura[3]. Dada una determinada diferencia entre dos sistemas (la representada por una ranura), la tasa de error de esa ranura representa la probabilidad de obtener una discrepancia sobre qué sistema es mejor al evaluar a los dos sistemas sobre dos conjuntos del mismo tamaño y totalmente disjuntos.

En el algoritmo de la Figura 4.4, *S* representa un conjunto de runs, mientras que *x* e *y* son runs pertenecientes a *S*. La colección de evaluación (conjunto de preguntas y respuestas) de la que se parte se representa por *Q*. M(*x*,Q_i) representa el resultado del run *x* sobre la subcolección Q_i de acuerdo con la medida de evaluación M.

[3]Recordar que cada ranura representa una diferencia en rendimiento (desde 0.01 hasta 0.2 en incrementos de 0.01) entre los resultados de dos sistemas

```
por cada par de runs x,y ϵ S
  por cada ejecución desde 1 hasta 500
    seleccionar Q_i , Q'_i ⊂ Q, donde Q_i ∩ Q'_i == φ y |Q_i| == |Q'_i| == c;
    d_M(Q_i) = M(x, Q_i) − M(y, Q_i);
    d_M(Q'_i) = M(x, Q'_i) − M(y, Q'_i);
    contador(RANURA(|d_M(Q_i)|))++;
    si d_M(Q_i) * d_M(Q'_i) < 0 entonces
      contador_swap(RANURA(|d_M(Q_i)|))++;
por cada ranura b
  tasa_error(b) = contador_swap(b)/contador(b);
```

Figura 4.4: Algoritmo para calcular la tasa de error de cada ranura de acuerdo con el método de swap de Voorhees and Buckley (2002) para una determinada medida de evaluación M

Dado que para realizar el experimento es necesario que los subconjuntos que se utilizan en cada comparación (Q_i y $Q_i^{'}$ en el algoritmo de la Figura 4.4 de la página 118) sean disjuntos entre si, su tamaño puede ser como mucho de hasta la mitad del tamaño de la colección de la que se parte. Es decir, puesto que la colección que se utiliza (la colección de evaluación en inglés del AVE 2008 tal y como se indicó en la sección 4.3.2 de la página 114) tiene algo más de 1000 respuestas, el tamaño utilizado en los experimentos fue c=500 [4] respuestas.

Además, observando la tasa de error de todas las ranuras se puede estimar la diferencia en resultados que se necesita para concluir qué sistema es mejor dado un determinado tamaño de colección de evaluación y un determinado valor de confianza. Por ejemplo, se puede hallar la diferencia que se debe dar para poder alcanzar la conclusión de que *el sistema A es mejor que el sistema B* con una confianza del 95 % seleccionando la diferencia que representa la primera ranura (contando desde las que representan diferencias más pequeñas hasta las que representan diferencias mayores) donde la tasa de error sea menor a 0.05 (lo cuál indica confianza mayor o igual al 95 %).

Una vez se tiene esta diferencia se puede calcular lo que se denomina sensibilidad de una medida de evaluación. La sensibilidad es la proporción de comparaciones durante el experimento en las cuáles se cumple la diferencia mínima requerida para una confianza dada (Sakai, 2007b). De este modo, cuanto más comparaciones cumplan dicha diferencia más sensible es la medida. Es por ello que cuanto más sensible es la medida, más útil será ésta para la discriminación de sistemas ya que habrá menos empates. Por tanto, la sensibilidad representa otra forma de evaluar el poder de discriminación de una medida.

Basándose en los resultados obtenidos para las 21 ranuras evaluadas tras apli-

[4] Se decidió utilizar el mismo tamaño de subcolecciones en los experimentos de la sección 4.3.3 (página 115) por cuestiones de homogeneidad

Cuadro 4.3: Resultados obtenidos tras aplicar el método de swap a *F* y *AUC* con un nivel de confianza del 95 % y con c=500: (i) Diferencia requerida para concluir que un sistema es mejor que otro con el nivel de confianza establecido; (ii) Máximo valor obtenido durante los experimentos; (iii) Diferencia requerida para ver qué sistema es mejor, relativa al máximo rendimiento observado ((i) / (ii)); (iv) Porcentaje de comparaciones realizadas en el experimento que cumplen la diferencia requerida (sensibilidad)

	(i)	**(ii)**	**(iii)**	**(iv)**
Medida F	0.09	0.77	11.65 %	74.39 %
AUC	0.12	0.96	12.47 %	52.08 %

car el algoritmo de la Figura 4.4 utilizando los runs y la colección en inglés del AVE 2008 con c=500, se ha construido el Cuadro 4.3 (página 119) para estudiar la sensibilidad de *F* y *AUC*. El Cuadro 4.3 muestra la sensibilidad (columna (iv)) de ambas medidas a un nivel de confianza del 95 %. La interpretación de los resultados del Cuadro 4.3 para, por ejemplo, *F* es la siguiente:

- De acuerdo con lo mostrado en la columna (i), se necesita una diferencia de al menos 0.09 al comparar resultados entre sistemas para concluir que un sistema es mejor que otro con una tasa de error menor al 5 % (confianza del 95 %) cuando se utiliza *F* como medida de evaluación.
- De entre todos los valores de *F* observados durante el experimento, el mayor fue de 0.77 como muestra la columna (ii).
- La diferencia necesaria para concluir qué sistema es mejor (mostrada en la columna (i)) en términos relativos a la máxima diferencia observada en el experimento (columna (ii)) es del 11.65 % como muestra la columna (iii). Este valor se haya dividiendo el resultado de la columna (i) entre el de la columna (ii).
- De entre todas las comparaciones realizadas utilizando *F*, un 74.39 % de ellas cumple la diferencia que se muestra en la columna (i). Este porcentaje refleja la sensibilidad de *F* y aparece en la columna (iv).

Los resultados del Cuadro 4.3 muestran que la sensibilidad de *F* es mayor que la de *AUC*. De nuevo cabe decir que estos resultados no deben servir para rechazar el uso de *AUC* en favor de *F* al evaluar sistemas de AV, sino para tener en cuenta que al realizar experimentos con *AUC* se necesitará una mayor diferencia entre sistemas para decidir qué sistema es mejor que si se utiliza *F*. Además, a partir de los resultados se deduce también que habrá menos empates entre sistemas al utilizar *F*, lo cuál es consistente con los resultados obtenidos en la sección 4.3.3 (página 115).

4.3.5. Adecuación a la Evaluación de la Validación

Al utilizar distintas medidas de evaluación lo más importante es conocer qué está midiendo cada medida y para qué escenario es más adecuado utilizar cada una (más importante incluso que conocer la confianza que se puede depositar en los resultados obtenidos por la medida). Teniendo este conocimiento y una vez conocido el escenario en el cuál se desea realizar la evaluación, es más fácil decidir cuál es la medida más apropiada para los propósitos del investigador.

Al evaluar sistemas de AV que realizan validación hay que tener en cuenta que el objetivo final del sistema de AV es mejorar los resultados de un sistema de QA. Para conseguir esta mejora, el sistema de AV ha de conseguir que el número de respuestas correctas se incremente, mientras que la cantidad de respuestas incorrectas disminuye.

Cuando un sistema de AV que realiza validación recibe una respuesta proveniente de un sistema de QA, hay cuatro posibles comportamientos distintos dependiendo de si la respuesta es o no correcta y de si el sistema de AV decide validar o rechazar la respuesta. Cada uno de estos comportamientos se corresponde con una posición en la matriz de confusión mostrada en el Cuadro 4.1 (página 106). Además, cada uno de estos comportamientos tiene una contribución distinta al objetivo final de mejorar los resultados en QA, por lo que pueden ser ordenados desde el que contribuye en mayor medida, hasta el que lo hace en menor medida. El orden de preferencias que se propone es el siguiente:

1. **Validar respuestas correctas:** este es el mejor comportamientos que se puede obtener ya que es el que contribuye en mayor medida a mejorar los resultados de un sistema de QA.

2. **Rechazar respuestas incorrectas:** este comportamientos también es deseable al permitir la reducción de respuestas incorrectas y que se consideren otras respuestas que podrían ser correctas. Sin embargo, la mejora en QA depende de la capacidad del sistema de AV detectando respuestas correctas, por lo que se le da menos importancia y ocupa la segunda posición en el orden de preferencias establecido.

3. **Rechazar respuestas correctas:** aunque éste es un comportamiento incorrecto, no contribuye al hecho de que un sistema de QA devuelva respuestas incorrectas. Además, este error se puede solucionar si se detecta alguna respuesta correcta a la misma pregunta. Dado que la posibilidad de recuperarse del error está condicionada a la capacidad de encontrar una respuesta correcta, se considera esta opción en tercer lugar dentro del ranking de preferencias.

4. **Validar respuestas incorrectas:** este es el peor comportamiento que se puede esperar y el que contribuye en mayor medida a obtener peores resultados en QA. Dado que no existe posibilidad de recuperarse del error cometido, se considera como la peor opción y la menos deseable.

De acuerdo a este orden de preferencias, se han comparado los resultados obtenidos utilizando *F* y *AUC* con el fin de comprobar cuál es la medida más adecuada para evaluar sistemas de AV que contribuyen a la mejora de los resultados en QA. Hay que destacar que *cobertura* y *tp rate* son diferentes nombres del mismo concepto, por lo que la principal diferencia entre los dos enfoques radica en el uso de *precisión* y *fp rate* (dejando a un lado el modo en el que se combinan para calcular *F* o *AUC*).

Por un lado, ambas medidas controlan la correcta validación de respuestas por medio de *cobertura* y *tp rate*, dando valores altos de evaluación a los sistemas que validan una alta proporción de las respuestas correctas. Las diferencias se encuentran en cuanto al control que realizan ambas medidas de la incorrecta validación de respuestas. Estas diferencias se deben al uso de *precisión* y *fp rate*.

Un valor bajo de *precisión* indica que se ha validado una cantidad alta de respuestas incorrectas, mientras que un valor alto significa lo contrario. Por tanto, *precisión* premia a los sistemas que validan una cantidad baja de respuestas incorrectas, mientras que penaliza a los que validan altas cantidades de respuestas incorrectas. De este modo, *precisión* contribuye a controlar el peor de los cuatro comportamientos, es decir, la incorrecta validación de respuestas.

Cuadro 4.4: Matriz de confusión de un sistema participante en inglés en el AVE 2008

	Respuestas correctas	**Respuestas incorrectas**	**Totales filas**
Respuestas validadas	68	129	197
Respuestas rechazadas	11	811	822
Totales columnas:	79	940	

Sin embargo, este comportamiento no está tan bien controlado por *fp rate*. Un valor bajo de *fp rate*[5] no significa que se haya validado una baja cantidad de respuestas incorrectas, sino que se ha validado una baja proporción de respuestas incorrectas con respecto al número total de respuestas incorrectas de la colección de evaluación. Se puede entender mejor este concepto utilizando el ejemplo del Cuadro 4.4 (página 121), el cuál muestra la matriz de confusión de un sistema de AV que participó en inglés en el AVE 2008.

Según los datos del Cuadro 4.4, dicho sistema está validando incorrectamente 129 respuestas, obteniendo un valor de *fp rate* de 0.14 (siendo el mejor valor posible 0). Sin embargo, 129 respuestas representan casi el doble de las respuestas que el sistema de AV está validando correctamente (68 respuestas), lo cuál mide *precisión* otorgando un valor de 0.35 (siendo el mejor valor posible 1). Por tanto,

[5]Hay que tener en cuenta que el mejor valor posible de *fp rate* es 0 y el peor 1, es decir, va al contrario que *precisión*

cuando un sistema de QA que utilice para realizar validación al sistema mostrado en el Cuadro 4.4 devuelva una respuesta, hay más posibilidades de que la respuesta sea incorrecta (65 % de opciones) que de que sea correcta (35 %). Esto se debe a que de acuerdo con los resultados obtenidos, solamente un 35 % de las respuestas validadas por dicho sistema son en realidad correctas. Está claro que éste es un comportamiento que no es deseable en AV. Sin embargo, *fp rate* está dando al sistema de AV un buen resultado (0.14 cuando el mejor posible es 0), mientras que *precisión* le da un resultado bajo (0.35 siendo el mejor posible 1). Por tanto, *precisión* aporta mayor información que *fp rate* en cuanto a la mejora de resultados que se puede conseguir al incluir un módulo de AV dentro de un sistema de QA.

Estos resultados sugieren que *F* es más adecuada que *AUC* cuando se desea evaluar la contribución que puede tener un sistema de AV en la mejora de los resultados en QA. *AUC* sería más útil en las situaciones en las cuáles el propósito de la evaluación es premiar en la misma proporción la detección de respuestas correctas e incorrectas sobre colecciones no balanceadas.

4.3.6. Discusión sobre la *Medida F*

Los resultados obtenidos en la sección 4.3.5 (página 120) podrían sugerir que la *precisión* debería de recibir más importancia en *F* que la *cobertura*. Para obtener este comportamiento se tendrían que utilizar valores de β menores que 1 en la Fórmula (2.6) de la página 38. Por ejemplo, un valor de $\beta = 0.5$ daría dos veces más importancia a la *precisión* que a la *cobertura*.

Sin embargo, en este trabajo se considera que el valor de β debe de ser escogido en función de los objetivos de la evaluación a realizar. Por ejemplo, si se quiere premiar a los sistemas de AV que sólo validan respuestas cuando están muy seguros de su decisión con el objetivo de reducir al máximo la cantidad de respuestas incorrectas (esto podría ser importante en escenarios de diagnóstico médico donde una respuesta incorrecta podría suponer un alto riesgo), entonces se utilizaría *F* dando más peso a la *precisión*. Un sistema con buenos resultados en este escenario podría tener dificultades para encontrar todas las respuestas correctas de una colección debido a que la *cobertura* no tiene mucho peso.

Por otro lado, hay escenarios donde es más importante tener a la salida la mayor cantidad de respuestas correctas sin importar si el usuario tiene que descartar las incorrectas. Para realizar la evaluación en estos escenarios tendría sentido utilizar *F* dando más peso a la *cobertura*.

Por tanto, en este trabajo se considera que es conveniente no dar un valor fijo de β como el más apropiado para ser utilizado al realizar evaluaciones de AV. Para este estudio se ha hecho uso del valor más utilizado ($\beta = 1$), el cuál se ha utilizado también para realizar la evaluación de sistemas de AV que se expondrá en el Capítulo 5 (página 125). Los experimentos realizados en este capítulo para evaluar la estabilidad y el poder de discriminación de *F* se pueden realizar para otros valores de β en caso de que se desee estudiar las características de F con otros valores de β distintos de 1.

4.4. Recapitulación

Este capítulo se ha centrado en el estudio de medidas para evaluar sistemas de AV. Este estudio ha servido para aportar medidas que permiten comparar sistemas de AV entre si y estudiar el impacto en resultados que supondría la inclusión de módulos de AV en QA.

En el capítulo se han propuesto dos grupos de medidas distintas, cada uno de los cuáles se centra en evaluar una de las dos funcionalidades que puede realizar un sistema de AV dentro de uno de QA: validación y selección.

1. Al evaluar la *validación de respuestas* es fácil observar que el problema a resolver consiste en uno de clasificación binaria en el cuál hay que decidir si una respuesta es correcta o incorrecta, por lo que se podría pensar en el uso de *accuracy* (la cuál es ampliamente utilizada para evaluar clasificación binaria). Sin embargo, se ha observado que *accuracy* no es adecuada debido a la naturaleza no balanceada de las colecciones de evaluación, lo que ha llevado a proponer el uso de *precisión*, *cobertura* y su media armónica (*medida F*) sobre las respuestas correctas.

2. Para realizar la evaluación de la *selección de respuestas* se han propuesto medidas que permiten estudiar la mejora que podría suponer el uso de módulos de AV dentro de un sistema de QA individual o multi-flujo. En concreto, se han estudiado medidas centradas en evaluar la correcta selección de respuestas, la capacidad de los sistemas de AV para detectar preguntas sin respuestas correctas y una estimación del rendimiento que podría alcanzar un sistema de QA que utilizase un módulo de AV que al detectar preguntas sin ninguna respuesta correcta solicita nuevas respuestas para esas preguntas.

Finalmente, se ha realizado un estudio comparando el uso de la *medida-F* y de una medida típica en la evaluación de clasificadores con colecciones no balanceadas, *AUC*, la cuál se obtiene a través del análisis del espacio ROC. Para realizar la comparación se ha estudiado la estabilidad y el poder de discriminación de ambas medidas, donde la *medida-F* ha obtenido mejores resultados. Además, se ha estudiado la adecuación de ambas medidas a la evaluación de sistemas de AV que pueden mejorar los resultados en QA. En este estudio, la *medida-F* ha mostrado ser capaz de establecer un mejor control sobre la validación de respuestas incorrectas en comparación con *AUC*, lo cuál es importante tener en cuenta para mejorar los resultados en QA.

Capítulo 5

Marco de Evaluación Desarrollado: Answer Validation Exercise

En este capítulo se describe una metodología para evaluar sistemas de Validación de Respuestas, además de mostrarse cómo se puso en práctica dicha metodología como una tarea de evaluación internacional: el Answer Validation Exercise (AVE) celebrado dentro del marco del CLEF en 2006, 2007 y 2008, cuya experiencia sirvió para ir refinando la metodología hasta su versión final.

La definición de esta metodología parte del modelo propuesto en el Capítulo 3 (página 89) de realizar la Validación de Respuestas basándose en RTE, y tiene como propósito establecer un marco común donde los sistemas de AV puedan ser comparados de forma competitiva entre si, haciendo uso de las medidas de evaluación expuestas en el Capítulo 4 (página 105). La experiencia positiva de su puesta en marcha dentro del AVE muestra que esta metodología puede ser aplicada a las evaluaciones de sistemas de AV.

A lo largo del capítulo se describen los objetivos que motivaron el desarrollo del marco de evaluación propuesto, su relación con las evaluaciones de sistemas de QA, las modificaciones que se realizaron en la metodología inicialmente propuesta y la manera de generar los recursos de evaluación necesarios. Además, se realiza un análisis de los resultados obtenidos en cada edición del AVE y de las técnicas utilizadas por los diversos sistemas participantes, así como los resultados obtenidos al hacer uso de las medidas de evaluación planteadas. Se concluye con un resumen de las principales aportaciones.

5.1. Objetivos

El marco de evaluación propuesto tiene como objetivo promover el desarrollo y la evaluación de módulos para realizar la validación de las respuestas generadas por sistemas de QA. Para ello, a los sistemas a evaluar se les propone validar au-

tomáticamente las respuestas generadas por sistemas reales de QA. De este modo, los sistemas deben de emular, en cierto sentido, a los evaluadores humanos de QA y decidir dada una respuesta, si ésta es o no correcta.

La entrada a los sistemas que van a ser evaluados está formada por un conjunto de tripletas {*Pregunta, Respuesta, Texto Soporte*}, para cada una de las cuáles hay que devolver un valor binario que indique si se valida o se rechaza la *Respuesta* dada a la *Pregunta* de acuerdo con la evidencia proporcionada por el *Texto Soporte*. El hecho de que la decisión sobre si validar o rechazar la *Respuesta* se tenga que tomar de acuerdo con el *Texto Soporte* tiene como objetivo tratar de promover la introducción de técnicas más sofisticadas de análisis textual que las basadas en buscar evidencias en recursos externos como, por ejemplo, la Web (algunas de estas técnicas fueron descritas en la sección 2.4.3 de la página 64).

Para fomentar la introducción de estas técnicas más complejas, el marco de evaluación propone la tarea de AV como un problema de reconocimiento de implicación textual siguiendo la propuesta realizada en la sección 3.1 (página 89). Es decir, combinar la respuesta con la forma afirmativa de la pregunta para formar una hipótesis que en caso de ser implicada por el texto soporte indique que la respuesta es valida. Además, el hecho de proponer la evaluación en términos de clasificación binaria (decidir si una respuesta es o no correcta) permite la posibilidad de utilizar técnicas de aprendizaje automático.

Los principales objetivos que se establecieron a la hora de desarrollar el marco de evaluación propuesto fueron:

1. Crear una metodología de evaluación de sistemas de AV con el ánimo de mejorar el rendimiento actual de los sistemas de QA y promover:
   - el desarrollo de sistemas de AV que tengan más en cuenta el análisis textual que la redundancia de respuestas
   - la utilización de módulos de AV que permitan disminuir el número de respuestas incorrectas y aumentar el número de respuestas correctas a la salida de un sistema de QA
   - el uso de arquitecturas de QA que hagan uso de criterios de selección de respuestas basados en AV
   - el uso de técnicas de aprendizaje automático para llevar a cabo la validación y la selección de respuestas
   - el desarrollo de sistemas de AV en varios idiomas
2. Proponer medidas para realizar la evaluación de sistemas de AV que realicen validación o selección de respuestas. Además, estas medidas no deben de permitir realizar solo una evaluación intrínseca, sino también realizar una evaluación extrínseca que permita cuantificar el posible beneficio que los sistemas de AV podrían aportar a los sistemas de QA. El hecho de obtener evidencias acerca de este posible beneficio serviría para promover la incorporación de módulos de AV dentro de los sistemas de QA.

3. Desarrollar una nueva tarea de evaluación, el Answer Validation Exercise (AVE), que ponga en práctica la metodología de evaluación propuesta.

4. Crear un conjunto de recursos que puedan ser utilizados en la tarea propuesta y en el desarrollo de futuros sistemas. Estos recursos deben de recoger la salida real de sistemas de QA con el propósito de hacer más real la evaluación.

5. Plantear la tarea de evaluación como un ejercicio competitivo en el que las condiciones de evaluación sean las mismas para todos los participantes y sea posible comparar los resultados obtenidos. Para ello han de utilizarse las mismas colecciones, la mismas medidas de evaluación y dar un intervalo de tiempo acotado a los participantes para realizar el procesamiento necesario.

6. Utilizar la evaluación desarrollada en cada edición de la tarea para analizar los resultados obtenidos por los sistemas participantes y determinar las mejores aproximaciones para la resolución del problema.

7. Mejorar la metodología de evaluación a lo largo de las diferentes ediciones de la tarea.

5.2. Relación entre la Evaluación de Búsqueda de Respuestas y el Marco Propuesto

La Figura 5.1 (página 128) muestra gráficamente la relación entre la evaluación de sistemas de QA y el marco de evaluación propuesto. Según el esquema de la Figura 5.1, la evaluación propuesta reutiliza las preguntas, respuestas y textos soporte generados en la evaluación de sistemas de QA, así como los juicios realizados por evaluadores humanos sobre las respuestas.

Debido a que en las evaluaciones de QA, para cada respuesta se suelen utilizar cuatro posibles juicios humanos (*correcta*, *incorrecta*, *inexacta* y *no soportada*), pero solo dos en AV (*correcta* o *incorrecta*), se hace necesario realizar una transformación entre ambos. La transformación se realiza de la forma que se propuso en la sección 3.2.3 (página 95). Es decir, las respuestas *correctas* en QA se etiquetan como *correctas* en AV, las respuestas *incorrectas* y las *no soportadas* en QA se etiquetan como *incorrectas* en AV y por último, las respuestas juzgadas como *inexactas* en QA se descartan en la evaluación de sistemas de AV.

Finalmente, la salida de los sistemas participantes en el marco de evaluación propuesto se compara con la de los resultados de las evaluaciones humanas sobre los sistemas de QA (tras su transformación a valores binarios), llevándose a cabo la evaluación.

Por tanto, la evaluación propuesta no tiene asociado un alto coste debido a que los recursos más costosos (preguntas y juicios) se obtienen a partir de los ya generados para una evaluación de sistemas de QA. De este modo, este marco de evaluación propone reutilizar recursos de QA con el propósito de evaluar y comparar a

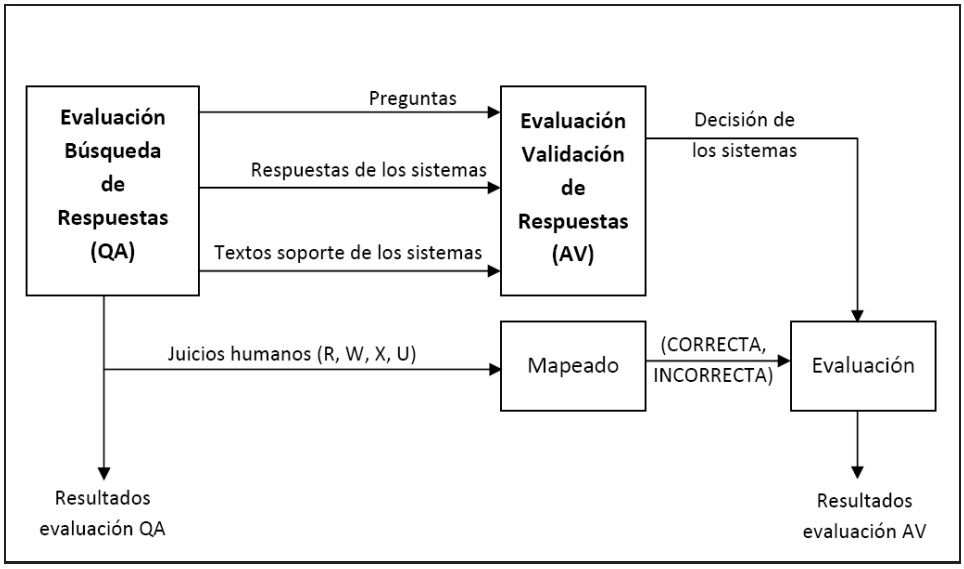

Figura 5.1: Gráfico sobre la relación entre la evaluación de sistemas de QA y la evaluación de sistemas de AV propuesta.

componentes de sistemas de QA (los módulos de AV). Además de la reducción en tiempo y esfuerzo, la reutilización de los juicios humanos (tras realizar una transformación automática) permite realizar una evaluación automática, lo cuál puede ser utilizado en el desarrollo de sistemas de AV basados en generación y prueba.

A la hora de poner en práctica la metodología propuesta como una tarea de evaluación, el hecho de proponer el AVE como una subtarea dentro de la evaluación de sistemas de QA del CLEF, la cuál se desarrolla en varios idiomas como por ejemplo inglés, francés, castellano, holandés, etc; permite reutilizar la salida de sistemas reales de QA en diversos idiomas. De este modo se alcanzan dos de los objetivos planteados en la sección 5.1 (página 125): realizar una evaluación sobre datos reales de sistemas de QA y hacerlo en varios idiomas.

Por otro lado, la temporización del AVE estuvo relacionado con la de la tarea principal de QA del CLEF. En concreto, la organización de la tarea de QA creaba inicialmente un conjunto de preguntas para las cuáles los sistemas participantes tenían que generar respuestas (acompañadas por su correspondientes textos soporte). A continuación, las respuestas de estos sistemas eran evaluadas por expertos humanos. Durante este periodo de evaluación manual se creaban y distribuían las colecciones de evaluación del AVE, las cuáles se obtenían a partir del conjunto de preguntas, respuestas y textos soporte generados por todos los sistemas de QA. De este modo, estas colecciones se creaban sin conocer los juicios de cada respuesta (si es correcta o no) tanto por parte de los participantes, como de la organización.

Al distribuir las colecciones de evaluación del AVE se solicitaba a los sistemas participantes que devolvieran sus decisiones antes de una determinada fecha, la cuál era anterior al momento en el que se distribuían los juicios sobre las res-

puestas de los sistemas de QA. De este modo se conseguía que no hubiera ningún participante en el AVE que pudiera tener conocimiento acerca de estos juicios.

Una vez se tenían los juicios humanos de cada respuesta emitida por los sistemas de QA, se realizaba la transformación automática de éstos (correcta, incorrecta, no soportada o inexacta) a un valor binario (correcta o incorrecta). A partir de estos juicios se realizaba la evaluación automática de los sistemas participantes en el AVE descartando las respuestas que fueron evaluadas como inexactas en QA.

5.3. Metodología de Evaluación

El Answer Validation Exercise (AVE) se propuso por primera vez en el año 2006 dentro del marco del CLEF con el objetivo de promover el desarrollo y la evaluación de módulos para validar las respuestas generadas por sistemas de QA. Durante las distintas ediciones del AVE se puso en marcha la metodología de evaluación propuesta, la cuál se fue refinando aprovechando la experiencia obtenida en cada edición, tal y como se describe a continuación.

5.3.1. Evolución de la Tarea

Al poner en práctica dentro del AVE la metodología de evaluación propuesta e ir refinándola con la experiencia obtenida, se fueron modificando las tareas que se propusieron inicialmente a los participantes. De este modo, en cada edición del AVE se fueron planteando nuevos retos a los sistemas participantes de modo que el comportamiento solicitado fuera cada vez más próximo al que tiene que realizar un módulo de AV dentro de un sistema de QA. La evolución desde el punto de vista de los sistemas participantes se detalla a continuación:

1. Los sistemas recibían inicialmente una serie de pares {*Texto, Hipótesis*}, donde la hipótesis era el resultado de la combinación de una pregunta y una respuesta a dicha pregunta, mientras que el texto correspondía con el fragmento de texto devuelto para soportar la veracidad de la respuesta. A los sistemas participantes se les proponía decidir si el texto implicaba a la hipótesis, lo que en este contexto significaba que la respuesta que dio lugar a la hipótesis era correcta. En caso contrario se consideraba que la respuesta era incorrecta. Este planteamiento de la tarea es muy similar al propuesto en las evaluaciones de los RTE Challenges y permitía omitir a los participantes el problema de la generación automática de hipótesis.

2. En la segunda edición se decidió que los participantes tuvieran que enfrentarse con el problema de la generación automática de hipótesis, de modo que en vez de pares {*Texto, Hipótesis*} se suministraron tripletas {*Pregunta, Respuesta, Texto Soporte*}. A partir de esta entrada, cada sistema debía indicar si la *Respuesta* de cada tripleta era o no correcta de acuerdo con el correspondiente *Texto Soporte*. Esta formulación permitió que los sistemas

participantes pudieran estudiar diversos enfoques para tratar la generación automática de hipótesis o plantear la opción de no crear hipótesis si consideraban que no era necesario para el tipo de procesamiento que realizaban.

3. A los sistemas de AV se les propuso realizar en la primera edición del AVE solamente validación de respuestas, de modo que el siguiente paso en la segunda edición del AVE fue plantear a los participantes que realizasen también selección de respuestas. Es decir, dado un conjunto de respuestas a una misma pregunta, cada sistema debía de seleccionar una o ninguna de esas respuestas como la candidata final a esa pregunta. En cierto modo, se pedía que cada sistema de AV simulase el comportamiento de uno de QA ya que para cada pregunta se permitía como máximo una respuesta.

4. Después de plantear que los sistemas realizasen selección de respuestas, se decidió llevar esta tarea un paso más adelante en la tercera edición del AVE. En concreto, se solicitó a los sistemas participantes que tuviesen especial cuidado a la hora de realizar la selección de respuestas de modo que fuesen capaces de detectar las preguntas para las cuáles ninguna de las respuestas candidatas era correcta, no seleccionando en estos casos ninguna respuesta. Por tanto, no sólo se pidió realizar la selección de respuestas correctas, sino también detectar al mismo tiempo las preguntas para las cuáles no había ninguna respuesta correcta.

5.3.2. Formulación basada en RTE

La metodología propuesta tiene como objetivo evaluar sistemas de AV que deben decidir por cada *Respuesta* a una *Pregunta* si la *Respuesta* es correcta o no de acuerdo a un determinado *Texto Soporte*. Esta metodología se definió inicialmente a partir del modelo de AV basado en RTE descrito en el Capítulo 3 (página 89). De acuerdo con este modelo, el sistema de AV tendría que combinar primero la *Pregunta* y la *Respuesta* para formar una hipótesis y a continuación hacer uso de un subsistema de RTE para comprobar si el *Texto Soporte* implica o no a la hipótesis creada. En caso de haber implicación, se valida la *Respuesta* y en caso contrario ésta se rechaza.

La primera vez que se puso en práctica esta metodología fue en el AVE 2006. En esta primera edición se decidió omitir el problema de la generación automática de hipótesis y suministrar a los participantes colecciones de evaluación con las hipótesis ya construidas, de modo que la tarea propuesta era similar a la desarrollada en los RTE Challenges. De este modo, los sistemas tenían que devolver un valor binario (SI o NO) para cada par texto-hipótesis indicando si el texto implicaba o no la hipótesis, es decir, si la *Respuesta* era o no correcta de acuerdo con el *Texto Soporte*.

Dado que la generación automática de hipótesis es una tarea que se debe realizar dentro de un sistema de AV basado en RTE, se decidió redefinir la metodología para hacer afrontar esta tarea a los sistemas a evaluar con el objetivo de poner

a prueba a los sistemas de AV en un entorno más real. De este modo, la nueva formulación consistía en dado un par {*Respuesta*, *Texto Soporte*} a una *Pregunta*, devolver un valor binario para indicar si se considera que la *Respuesta* a la *Pregunta* es correcta de acuerdo al *Texto Soporte*, o si se considera lo contrario. Este enfoque se puso en práctica satisfactoriamente en el AVE 2007 y 2008, lo cuál permitió evaluar a los sistemas en un entorno más real.

5.3.3. Detección de Respuestas Correctas

La metodología propuesta se centró inicialmente en evaluar solamente a los sistemas de AV cuando realizan validación. Es decir, la evaluación se centró en la detección de respuestas correctas y solo éstas. Para evaluar la validación se utilizaron las medidas de *precisión*, *cobertura* y su media armónica (*medida-F*) sobre las respuestas correctas, las cuáles fueron descritas en la sección 4.1 (página 105).

Como se mencionó al describir estas medidas, hay que tener en cuenta que los resultados obtenidos mediante estas medidas no permiten realizar comparación entre colecciones con distinta proporción de respuestas correctas e incorrectas. Por este motivo se propuso el uso de dos sistemas "baseline" para realizar comparaciones dentro de una misma colección: un sistema que siempre valida todas las respuestas (baseline 100 % SI), y un sistema que valida la mitad de las respuestas (baseline 50 % SI).

Este tipo de evaluación se realizó durante las tres ediciones del AVE, observándose en la primera edición que este enfoque solo permitía realizar una evaluación intrínseca que no era suficiente para estudiar la mejora en resultados que podría obtener un sistema de QA al incorporar un módulo de AV (lo que era uno de los objetivos del marco de evaluación propuesto). Es cierto que la comparación realizada con el baseline 100 % SI permite estudiar si los sistemas propuestos de AV mejorarían el rendimiento de un hipotético sistema de QA que generase las respuestas de la colección utilizada y no realizase una etapa de validación sobre ellas. Sin embargo, estos resultados no se podían comparar con los de sistemas reales de QA (por ejemplo aquellos participantes en la tarea de QA del CLEF) para comprobar si se obtenía mejora sobre enfoques reales. Esta observación sirvió para añadir a la metodología medidas de evaluación para alcanzar este propósito.

5.3.4. Selección de Respuestas Correctas

Tras analizar la evaluación realizada durante el AVE 2006, se decidió incorporar a la metodología propuesta la evaluación de la selección de respuestas, poniéndose en práctica dicha evaluación en el AVE 2007 y 2008. De este modo, además de decidir si cada *Respuesta* era o no correcta, los sistemas tenían que seleccionar también una *Respuesta* por *Pregunta* de modo que para cada *Pregunta* no se podía seleccionar más de una *Respuesta*, y dentro de cada *Pregunta* al menos una de las respuestas validadas tenía que ser seleccionada.

El hecho de añadir la evaluación de la selección de respuestas no sólo permitió evaluar otra de las funciones que puede desempeñar un sistema de AV dentro de uno de QA, sino que además permitió estudiar la posible mejora en cuanto a resultados que podría suponer la incorporación de módulos de AV en sistemas de QA (la cuál se describirá en la sección 5.3.5 de la página 132).

Para realizar esta evaluación se propuso el uso de *qa_accuracy* (Fórmula (4.4) de la página 110), descrita en la sección 4.2.1 (página 109), que mide el acierto de un sistema seleccionando respuestas correctas dado un conjunto de preguntas. Los resultados obtenidos utilizando *qa_accuracy* se pueden comparar con los del sistema "baseline" *random_qa_accuracy* (Fórmula (4.6) de la página 110), el cuál representa la proporción media de respuestas correctas por cada pregunta.

5.3.5. Comparación con Sistemas de Búsqueda de Respuestas

En la evaluación que se plantea para medir el rendimiento de los sistemas de AV que realizan selección, el comportamiento que se tiene a la salida es comparable al de un sistema de QA que como máximo devuelve una respuesta por pregunta, ya que para cada pregunta no hay más de una respuesta. De este modo, se pueden comparar los resultados obtenidos por un sistema de QA utilizando *accuracy* con los de un sistema de AV que realiza selección sobre el mismo conjunto de preguntas utilizando la medida *qa_accuracy*, ya que ambas medidas miden la proporción de respuestas correctas.

En concreto, esta comparación se realiza entre un determinado número de sistemas individuales de QA, y un conjunto de módulos de AV que en cada pregunta realizan la selección de la respuesta de un sistema multi-flujo de QA formado por todos los sistemas de QA con los cuáles se realiza la comparación. Esta comparación permite estudiar si la combinación de distintos sistemas de QA en uno multi-flujo donde la selección de la respuesta final es llevada a cabo por un módulo de AV puede mejorar los resultados de los sistemas individuales. Además, en este escenario se pueden comparar los resultados con los obtenidos si se realizase una selección perfecta de respuestas, comprobando para cada sistema de AV qué porcentaje de dicha selección perfecta se ha alcanzado.

Al poner en práctica esta parte de la metodología dentro del AVE en las ediciones de 2007 y 2008, las comparaciones se realizaron entre los sistemas de AV que participaron en el AVE y los sistemas de QA participantes en el QA@CLEF del mismo año, ya que las colecciones de evaluación del AVE se generaron a partir de estos sistemas de QA. Los resultados obtenidos en estas comparaciones permitieron observar la mejora de resultados que podría suponer el uso de sistemas de AV que realizan selección de respuestas, y además animaron a algunos participantes a incluir sus módulos de AV dentro de sus sistemas de QA en el CLEF 2008.

5.3.6. Detección de Preguntas sin Respuestas Correctas

El análisis de los resultados obtenidos al evaluar la selección de respuestas en el AVE 2007 mostró que había sistemas de AV capaces de detectar preguntas para las cuáles ninguna respuesta de las dadas por los sistemas de QA era correcta, y en las que por tanto no tenía sentido seleccionar ninguna respuesta. Detectar estas preguntas supone un comportamiento correcto que, sin embargo, no tenía en cuenta la medida de evaluación propuesta (en concreto *qa_accuracy*, que sirve para evaluar la correcta selección de respuestas y fue descrita en la sección 4.2.1 de la página 109). Hay que tener en cuenta que esta capacidad de los sistemas de AV puede ser útil por dos motivos:

1. Si se considera que no se puede encontrar ninguna respuesta correcta a una determinada pregunta, se puede tomar la decisión de no contestar a la pregunta, lo cuál puede ser ventajoso en algunos escenarios donde una respuesta incorrecta lleva asociado un alto coste. Este tipo de escenarios se estudian con más detalle en el Capítulo 6 (página 161).

2. Por otro lado, en estas preguntas se podrían solicitar nuevas respuestas a los sistemas de QA, pudiéndose obtener una respuesta correcta en esta segunda oportunidad.

Al observar la importancia que podría tener esta capacidad de los sistemas de AV, se decidió incorporar a la metodología propuesta la evaluación de la detección de preguntas sin respuestas correctas. Para llevar a cabo esta evaluación se propuso el uso de *qa_rej_accuracy* (Fórmula (4.7) de la página 110), descrita en la sección 4.2.2 (página 110). Esta medida se incorporó al AVE en la edición de 2008.

5.3.7. Estimación de la Mejora Potencial de Sistemas QA con AV

Como se indicó en la sección 5.3.6 (página 133), la detección de preguntas sin ninguna respuesta correcta podría permitir la mejora de resultados si se solicitasen nuevas respuestas a estas preguntas. Es por ello que se consideró importante estimar el rendimiento que se podría obtener teniendo en cuenta este comportamiento.

Para realizar esta estimación se incorporaron a la metodología propuesta las medidas descritas en la sección 4.2.3 (página 111), las cuáles tienen en cuenta el rendimiento de un sistema al realizar selección de respuestas y la correcta detección de preguntas sin respuestas correctas. Estas medidas son:

1. Por un lado *qa_accuracy_max* (Fórmula (4.8) de la página 111) supone un límite superior del rendimiento que se obtendría si se encontrase una respuesta correcta para cada pregunta que se ha detectado que no tiene ninguna respuesta correcta entre las candidatas.

2. Por otro lado, *estimated_qa_performance* (Fórmula (4.9) de la página 111) supone una estimación más real que *qa_accuracy_max* del rendimiento que

se obtendría al buscar respuestas las preguntas para las cuáles se ha detectado que no había respuestas correctas entre las candidatas. Para realizar esta estimación se asume que estas preguntas serán respondidas con la precisión observada anteriormente, la cuál está representada por *qa_accuracy* (Fórmula (4.4) de la página 110), que es la medida que se utiliza para evaluar la correcta selección de respuestas.

En la versión final de la metodología, a la hora de evaluar sistemas de AV que realizan selección, se propone elegir la medida de evaluación en función del comportamiento que se desea evaluar. Por ejemplo, *estimated_qa_performance* (Fórmula (4.9) de la página 111) es apropiada para estimar el rendimiento que se podría obtener al permitir buscar nuevas respuestas a las preguntas para las cuáles no se encuentran respuestas correctas. Sin embargo, si se desea evaluar solamente la correcta selección de respuestas sin tener en cuenta la detección de preguntas sin ninguna respuesta correcta, entonces se propone utilizar *qa_accuracy* (Fórmula (4.4) de la página 110). En caso de que se desee evaluar la detección de preguntas sin respuestas correctas, se sugiere hacer uso de *qa_rej_accuracy* (Fórmula (4.7) de la página 110).

5.4. Generación de los Recursos de Evaluación

Para realizar la evaluación descrita se hace necesario el uso de colecciones de evaluación, formadas básicamente por un conjunto de pares {*Respuesta*, *Texto Soporte*} agrupados por {*Pregunta*}. Este formato permite evaluar tanto la validación (decidir si una *Respuesta* es o no correcta) como la selección (elegir una o ninguna *Respuesta* por *Pregunta*). Según el marco definido, las colecciones se generan a partir de la salida de sistemas de QA partiendo de:

- Preguntas creadas por expertos humanos para la evaluación de sistemas de QA.
- Las respuestas y textos soporte (que soportan la veracidad de cada respuesta) generados por sistemas de QA para responder a las preguntas planteadas.
- Juicios humanos sobre la veracidad de las respuestas (de acuerdo con los textos soporte)

A la hora de construir colecciones para evaluar sistemas de AV basados en RTE se proponen dos enfoques distintos: uno en el cuál se suministran hipótesis ya construidas para que los sistemas no tengan que preocuparse de generarlas; y otro en el cuál no se dan las hipótesis con el fin de que los sistemas sean evaluados en un entorno más real. Para el AVE se hizo uso de ambos enfoques, cada uno de los cuáles se explica con más detalle en los siguientes subapartados.

En el caso concreto del AVE, los recursos se generaron a partir de la salida de los participantes en la tarea de QA del CLEF. A partir de esta salida, en cada

edición del AVE se construyeron y suministraron a los participantes colecciones para el desarrollo de los sistemas y colecciones para realizar la evaluación.

5.4.1. Omitiendo la Generación Automática de Hipótesis

El primer enfoque propuesto para crear colecciones de evaluación de sistemas de AV consiste en suministrar hipótesis ya construidas a partir de las preguntas y las respuestas de sistemas reales de QA. De este modo, los sistemas a ser evaluados no tienen que preocuparse del problema de la generación automática de hipótesis.

Estas colecciones están formadas por un conjunto de pares texto-hipótesis donde cada hipótesis se crea combinando una pregunta con una respuesta a dicha pregunta. Para cada uno de estos pares hay que decidir si el texto implica o no a la hipótesis, lo cual indica si la respuesta es o no correcta. De este modo, este tipo de colecciones son similares a las utilizadas en los RTE Challenges y por tanto pueden ser utilizadas también para evaluar sistemas de RTE.

Para construir estas colecciones se sigue el método descrito en la sección 3.2 (página 93), por lo que el proceso no es totalmente automático. Esto se debe a que durante el proceso se debe crear manualmente un patrón de hipótesis para cada pregunta. Posteriormente, cada patrón de hipótesis se instancia automáticamente con cada una de las respuestas a la pregunta que dio lugar al patrón para obtener así las distintas hipótesis de la colección. Finalmente se toman los textos soporte de cada respuesta y se crean las colecciones de pares texto-hipótesis definitivas. Una representación esquemática del proceso que se sigue para construir estas colecciones se muestra en la Figura 5.2 (página 135).

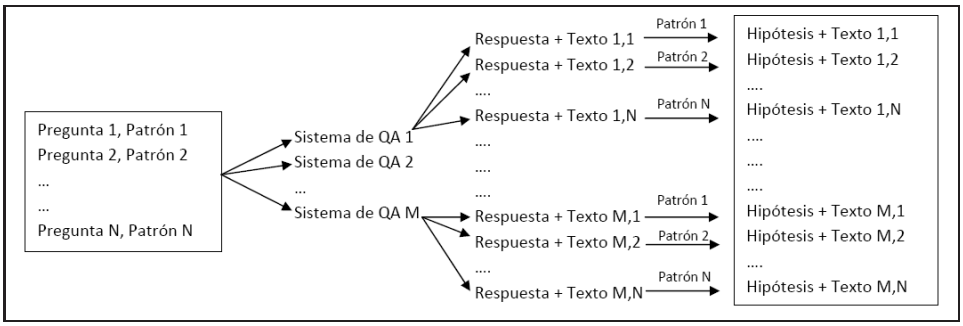

Figura 5.2: Proceso para construir pares texto-hipótesis a partir del conjunto disponible de respuestas.

En función de las características de los recursos de QA de los cuales se parte existen dos posibilidades distintas para generar colecciones de este tipo:

1. En caso de que la salida de sistemas de QA de la que se parte no contenga fragmentos de texto soporte sino identificadores de documentos soporte (siendo estos documentos de gran longitud), se hace necesario extraer los textos soporte. La colección descrita en la sección 3.2 (página 93) es de este

tipo, por lo que se propone el método descrito en la sección 3.2.2 (página 94) para extraer automáticamente los textos soporte. Sin embargo, en estos casos pueden generarse pares con errores en el valor del par debido a que los juicios humanos de las respuestas están realizados considerando el documento entero y no fragmentos de texto (errores que se vieron en la sección 3.2.5 de la página 96).

2. En caso de que la salida de sistemas de QA contenga textos soporte, no se tiene que realizar ninguna extracción automática. En estas situaciones se tiene que el texto de cada par se corresponde con el texto soporte devuelto para la respuesta que dio lugar a la hipótesis del par. Dado que los juicios que se utilizan han sido realizados teniendo en cuenta estos textos, en estos casos se omiten los errores que se tenían en la opción anterior.

Las colecciones formadas por pares texto-hipótesis presentan la ventaja de que los sistemas a ser evaluados reciben hipótesis ya construidas, permitiendo a los investigadores omitir tal tarea. Sin embargo, aparte de que el proceso para construirlas no es totalmente automático, mediante el uso de este enfoque se pueden introducir errores sintácticos en las hipótesis que pueden dificultar el procesamiento de los sistemas. Además, hay ocasiones en las cuáles una respuesta incorrecta puede resultar en una hipótesis implicada por su respectivo texto soporte, como se mostró en la sección 3.2.5 (página 96). Por estos motivos, este tipo de colecciones fue utilizado únicamente en el AVE 2006, aunque la metodología para generarlas puede ser utilizada para crear nuevas colecciones.

En el caso concreto del AVE 2006 se suministraron colecciones de desarrollo solamente en inglés y español, extrayéndose los textos de ambas colecciones automáticamente. Esto se debió a que en los recursos de los que se partió (la salida de los sistemas participantes en la tarea de QA del CLEF 2003, 2004 y 2005) se contaba con los identificadores del documento soporte y no con fragmentos de texto soporte. En el Cuadro 5.1 (página 136) se puede ver el número de pares SI y NO que hubo en las colecciones de desarrollo del AVE 2006.

Cuadro 5.1: Pares SI y NO en las colecciones de desarrollo del AVE 2006

	Inglés	**Español**
Pares SI	676	400
Pares NO	2128	2302
Total	2804	2702

Los Cuadros 5.2 (página 137) y 5.3 (página 137) muestran el número de pares SI, NO y UNKNOWN (pares creados a partir de las respuestas evaluadas como *inexactas* o que no fueron evaluadas) que hubo en las colecciones de test del AVE 2006. Se puede ver que en estas colecciones el porcentaje de pares UNKNOWN es similar para todos los idiomas excepto en inglés y portugués, en donde hasta 5

runs de los inicialmente enviados a la tarea de QA del CLEF (a partir de la cuál se generaron las colecciones) no fueron finalmente evaluados y por tanto, no se pudo asignar un valor a los pares obtenidos a partir de dichos runs.

Cuadro 5.2: Pares SI, NO y UNKNOWN en las colecciones de test del AVE 2006

	Alemán	**Inglés**	**Español**	**Francés**
Pares SI	344	198	671	705
Pares NO	1064	1048	1615	2359
UNKNOWN	35	842	83	202
Total	1443	2088	2369	3266

Cuadro 5.3: Pares SI, NO y UNKNOWN en las colecciones de test del AVE 2006

	Italiano	**Holandés**	**Portugués**
Pares SI	187	81	188
Pares NO	901	696	604
UNKNOWN	52	30	532
Total	1140	807	1324

5.4.2. Permitiendo la Generación Automática de Hipótesis

En el otro tipo de colecciones que se plantean dentro del marco de evaluación propuesto no se crean hipótesis, por lo que las colecciones son más realistas al contener la salida de sistemas de QA tal y como la van a recibir los sistemas de AV. En concreto, en estas colecciones se tiene para cada *Pregunta* un conjunto de pares {*Respuesta*, *Texto Soporte*} que hay que validar o rechazar. Además, el hecho de que las respuestas estén agrupadas por pregunta facilita la selección de una o ninguna respuesta para cada pregunta. Por tanto, estas colecciones permiten evaluar tantos sistemas de AV que realizan validación (decidir si las respuestas son o no correctas) como sistemas que realizan selección (elegir una o ninguna respuesta por cada pregunta). Un ejemplo de este tipo de colecciones puede verse en la Figura 5.3 (página 138), la cuál contiene un fragmento de la colección de evaluación de inglés del AVE 2008.

Debido a que el hecho de agrupar todas las respuestas a una misma pregunta puede suponer aportar información adicional basada en el conteo de redundancias de respuestas, un comportamiento que se desea evitar en los sistemas a evaluar, al crear estas colecciones se eliminan las respuestas repetidas para una misma pregunta. De hecho, si una respuesta está contenida dentro de otra se elimina la de menor tamaño. Este procedimiento tiene como consecuencia una reducción en cuanto al número de respuestas disponibles inicialmente. Por ejemplo, en la colección de desarrollo del AVE 2007 en búlgaro esta reducción fue del 88.3 %.

```
<q id="0001" lang="EN">
        <q_str>What was the nationality of Jacques
        Offenbach?</q_str>
        <a id="0001_1" value="">
                <a_str>Germany</a_str>
                <t_str doc="Offenbach">Offenbach Offenbach Offenbach
                can refer to: The city Offenbach in Hesse,
                Germany.</t_str>
        </a>
        <a id="0001_2" value="">
                <a_str>France</a_str>
                <t_str doc="Jacques Offenbach">His son received the
                name "Jakob Offenbach" at birth, though he changed it
                to Jacques when he settled in France.</t_str>
        </a>
        <a id="0001_3" value="">
                <a_str>Thousand Oaks</a_str>
                <t_str doc="LA111794-0288">Ventura College's
                production of George Bernard Shaw's "Arms and the
                Man" and  Moorpark College's version of the Jacques
                Offenbach operetta "La Vie  Parisienne" are the
                costume shows; in Thousand Oaks, Cal Lutheran
                University is  mounting the contemporary drama "Minor
                Demons."</t_str>
        </a>
        ...
</q>
```

Figura 5.3: Fragmento de la colección de evaluación en inglés del AVE 2008

Las colecciones del AVE 2007 y 2008 fueron generadas siguiendo este procedimiento. Los Cuadros 5.4 (página 138) y 5.5 (página 139) muestran el número de preguntas y respuestas (indicando también la distribución en respuestas correctas e incorrectas y el porcentaje de respuestas que se utilizaron sobre las disponibles inicialmente) de las colecciones de desarrollo de las que dispusieron los participantes en el AVE 2007. Estas colecciones fueron generadas a partir de la salida de los sistemas que participaron en la tarea de QA del CLEF 2006, es decir, de la misma fuente de la cuál se crearon las colecciones de evaluación del AVE 2006.

Cuadro 5.4: Número de preguntas y respuestas en las colecciones de desarrollo del AVE 2007

	Alemán	**Inglés**	**Español**	**Francés**
Preguntas	187	200	200	200
Respuestas(final)	504	1121	1817	1503
% sobre respuestas disponibles	31.5	62.28	53.44	50.1
Correctas	135	130	265	263
Incorrectas	369	991	1552	1240

Cuadro 5.5: Número de preguntas y respuestas en las colecciones de desarrollo del AVE 2007

	Italiano	**Holandés**	**Portugués**	**Búlgaro**
Preguntas	192	198	200	56
Respuestas(final)	476	528	817	70
% sobre respuestas disponibles	47.6	44	40.85	11.67
Correctas	86	100	153	49
Incorrectas	390	428	664	21

Los Cuadros 5.6 (página 139) y 5.7 (página 139) muestran el número de preguntas y respuestas (indicando también la distribución en respuestas correctas, incorrectas y UNKNOWN, así como el porcentaje de respuestas que se utilizaron sobre las disponibles inicialmente) de las colecciones sobre las que se realizó la evaluación en 2007. Estas colecciones se generaron a partir de la salida de los sistemas participantes en la tarea de QA del CLEF 2007.

Cuadro 5.6: Número de preguntas y respuestas en las colecciones de test del AVE 2007

	Alemán	**Inglés**	**Español**	**Francés**
Preguntas	113	67	170	122
Respuestas(final)	282	202	564	187
% sobre respuestas disponibles	48.62	60.3	66.35	75.4
Correctas	67	21	127	85
Incorrectas	197	174	424	86
UNKNOWN	18	7	13	16

Cuadro 5.7: Número de preguntas y respuestas en las colecciones de test del AVE 2007

	Italiano	**Holandés**	**Portugués**	**Rumano**
Preguntas	103	78	149	100
Respuestas(final)	103	202	367	127
% sobre respuestas disponibles	88.79	51.79	30.58	52.05
Correctas	16	31	148	45
Incorrectas	84	165	198	58
UNKNOWN	3	6	21	24

Hay que tener en cuenta que la edición de 2007 del QA@CLEF fue la primera en la cual las preguntas fueron agrupadas por topics. En esta organización por topics, la primera pregunta a un determinado topic es autocontenida en el sentido de que no hay necesidad de buscar información fuera de la pregunta para responderla. Sin embargo, el resto de preguntas del topic pueden hacer referencia a información presente en las preguntas o respuestas anteriores dentro de dicho topic (como por ejemplo referencias a algo anterior mediante el uso de pronombres). Debido a este motivo, para crear las colecciones del AVE sólo se hizo uso de las preguntas autocontenidas (la primera de cada topic) y de las respectivas respuestas dadas por los sistemas de QA a dichas preguntas. Este proceso de considerar menos preguntas junto con la eliminación de redundancias en las respuestas tuvo como consecuencia la reducción del tamaño de las colecciones.

Las colecciones suministradas para el desarrollo de sistemas a los participantes del AVE 2008 se crearon a partir de la salida de los sistemas participantes en la tarea de QA del CLEF 2006 y 2007. Es decir, se utilizaron las colecciones de desarrollo y evaluación del AVE 2007 (tras eliminar las respuestas con valor *UNKNOWN*). Los Cuadros 5.8 (página 140) y 5.9 (página 140) muestran el número de preguntas y respuestas (indicando también la distribución en respuestas correctas e incorrectas y el porcentaje de respuestas que se utilizaron sobre las disponibles inicialmente) de las colecciones de desarrollo del AVE 2008.

Cuadro 5.8: Número de preguntas y respuestas en las colecciones de desarrollo del AVE 2008

	Alemán	**Inglés**	**Español**	**Francés**
Preguntas	295	267	369	318
Respuestas(final)	768	1316	2368	1674
% sobre respuestas disponibles	35.1	61.64	55.72	51.54
Correctas	202	151	392	348
Incorrectas	566	1165	1976	1326

Cuadro 5.9: Número de preguntas y respuestas en las colecciones de desarrollo del AVE 2008

	Italiano	**Holandés**	**Portugués**	**Rumano**	**Búlgaro**
Preguntas	292	276	348	82	56
Respuestas(final)	576	724	1163	103	70
% sobre respuestas disponibles	51.61	45.53	36.34	42.21	11.67
Correctas	102	131	301	45	49
Incorrectas	474	593	862	58	21

Las colecciones de evaluación del AVE 2008 se crearon a partir de la salida de los sistemas participantes en la tarea de QA del CLEF 2008. Los Cuadros 5.10 (página 141) y 5.11 (página 141) muestran el número de preguntas y respuestas (indicando también la distribución en respuestas correctas, incorrectas y UNKNOWN y el porcentaje de respuestas que se utilizaron sobre las disponibles inicialmente) de las colecciones de evaluación del AVE 2008.

Cuadro 5.10: Número de preguntas y respuestas en las colecciones de test del AVE 2008

	Alemán	**Inglés**	**Español**	**Francés**
Preguntas	119	160	136	108
Respuestas(final)	1027	1055	1528	199
% sobre respuestas disponibles	39.61	57.37	49.98	60.3
Correctas	111	79	153	52
Incorrectas	854	940	1354	126
UNKNOWN	62	36	21	21

Cuadro 5.11: Número de preguntas y respuestas en las colecciones de test del AVE 2008

	Holandés	**Portugués**	**Rumano**	**Vasco**	**Búlgaro**
Preguntas	128	149	119	104	27
Respuestas(final)	228	1014	497	541	27
% sobre respuestas disponibles	42.54	43.63	48.58	55.09	21.4
Correctas	44	208	52	39	12
Incorrectas	177	747	406	483	9
UNKNOWN	7	59	39	19	6

Un resumen de las colecciones generadas en cada idioma durante las tres ediciones del AVE se puede ver en el Cuadro 5.12 (página 142), el cuál muestra los datos de colecciones para evaluar sistemas de AV generadas en 10 idiomas distintos. Estas colecciones están disponibles en el sitio web del AVE[1] para investigadores registrados en el CLEF, con lo cuál pueden seguir siendo utilizadas para el desarrollo y evaluación de sistemas.

[1] `http://nlp.uned.es/clef-qa/ave/`

Cuadro 5.12: Número de preguntas y respuestas resultantes del AVE 2006, 2007 y 2008

		Respuestas		
	Preguntas	**Correctas**	**Incorrectas**	**Totales**
Español	505	545	3330	3875
Inglés	427	230	2105	2335
Alemán	414	313	1420	1733
Francés	426	400	1452	1852
Italiano	292	102	474	576
Holandés	404	175	770	945
Portugués	497	509	1609	2118
Rumano	201	97	464	561
Búlgaro	83	61	30	91
Vasco	104	39	483	522

5.5. Análisis de Resultados

A lo largo de esta sección se discuten los resultados obtenidos por los distintos participantes en cada edición del AVE.

5.5.1. Resultados AVE 2006

En la primera edición del AVE participaron once grupos en siete idiomas distintos (alemán, inglés, español, francés, italiano, holandés y portugués), recibiéndose un total de 38 runs. El Cuadro 5.13 (página 143) muestra los grupos participantes y el número de runs que envió cada uno de ellos en cada idioma. Como se puede observar en el Cuadro, entre los participantes siempre hubo al menos dos grupos distintos participando en un mismo idioma, permitiendo de este modo la comparación entre distintos enfoques en todos los idiomas. Inglés y español (aquellos idiomas para los cuáles se suministró una colección de desarrollo) fueron los idiomas con más participación con 11 y 9 runs respectivamente.

Los sistemas participantes en el AVE 2006 fueron evaluados respecto a la calidad de la validación que realizaban utilizando las medidas descritas en la sección 4.1 (página 105), es decir, *precisión*, *cobertura* y su media armónica (*medida F*) sobre las respuestas correctas. Los resultados obtenidos por los participantes se muestran en los Cuadros del A.1 (página 203) al A.7 (página 205) del Anexo A (página 203). Un resumen de los resultados se puede ver en el Cuadro 5.14 (página 144), donde se muestran los mejores resultados de cada participante en cada idioma.

Cuadro 5.13: Participantes y runs por cada idioma en el AVE 2006

	Alemán	Inglés	Español	Francés	Italiano	Holandés	Portugués	Total
Fernuniversität in Hagen (FUH)	2							2
Language Computer Corporation (LCC)		1	1					2
U. Rome "Tor Vergata"		2						2
U. Alicante (Kozareva)	2	2	2	2	2	2	1	13
U. Politécnica de Valencia		1						1
U. Alicante (Ferrández)		2						2
LIMSI-CNRS				1				1
U. Twente	1	2	2	1	1	2	1	10
UNED (Herrera)			2					2
UNED (Rodrigo)			1					1
ITC-irst		1						1
Proyecto R2D2			1					1
Total	5	11	9	4	3	4	2	38

Estos resultados no permiten realizar comparación entre distintos idiomas dada la distinta distribución de respuestas correctas en cada idioma, pero si se pueden comparar los resultados dentro de cada idioma con los de dos "baselines": los resultados de un sistema que valida todas las respuestas (baseline 100 % SI), y los resultados de un sistema que valida la mitad de las respuestas (baseline 50 % SI).

En los resultados se puede ver que excepto en portugués (Cuadro A.6 de la página 205), en el resto de idiomas hubo siempre algún sistema que logró mejorar los resultados de los "baselines" propuestos. De hecho, la mayor parte de los sistemas presentados consiguieron superar a los "baselines". Este dato indica que los enfoques propuestos en esta edición podían conseguir devolver un conjunto de respuestas candidatas mejor que el que se obtendría si no se hubiese realizado validación.

Por otro lado, los mejores resultados se obtuvieron en inglés y español. El hecho de que estos dos idiomas fueran los únicos para los cuáles se suministró una colección de desarrollo parece ser el motivo principal de estos resultados.

5.5.2. Resultados AVE 2007

En el AVE 2007 participaron nueve grupos (dos menos que en la anterior edición) en cuatro idiomas distintos (alemán, español, inglés y portugués), volviendo

Cuadro 5.14: Resumen resultados del AVE 2006. Mejores valores de F obtenidos para cada sistema e idioma

	Alemán	Inglés	Español	Francés	Italiano	Holandés	Portugués
Language Computer Corporation		0.46	0.61				
Fernuniversität in Hagen	0.54						
UNED (Herrera)			0.57				
UNED (Rodrigo)			0.53				
U. Rome "Tor Vergata"		0.41					
ITC-irst		0.39					
U. Alicante (Kozareva)	0.47	0.37	0.53	0.47	0.41	0.30	0.15
Proyecto R2D2			0.49				
U. Alicante (Ferrández)		0.32					
LIMSI-CNRS				0.11			
U. Twente	0.14	0.30	0.47	0.09	0.17	0.39	0.35
U. Politécnica de Valencia		0.08					
Baseline 100 % SI	0.39	0.27	0.45	0.37	0.29	0.19	0.38
Baseline 50 % SI	0.33	0.24	0.37	0.32	0.26	0.17	0.32

a ser inglés y español los idiomas con más participación con 8 y 5 runs cada uno respectivamente. La participación respecto a la edición de 2006 se redujo debido, principalmente, a que no se suministraron colecciones con las hipótesis ya construidas, con lo que hubo grupos interesados en el desarrollo de sistemas de RTE que participaron en la edición anterior y no en ésta. El Cuadro 5.15 muestra los grupos participantes y el número de runs que envió cada uno de ellos en cada idioma.

En esta edición se evaluó por primera vez tanto la validación como la selección de respuestas. La validación fue evaluada utilizando las mismas medidas de la edición anterior (*precisión*, *cobertura* y *F*). En cuanto a la selección, sólo se evaluó la correcta selección haciendo uso de *qa_accuracy* (Fórmula (4.4) de la página 110), descrita en la sección 4.2.1 (página 109).

Los resultados obtenidos en cuanto a la validación de respuestas se muestran en los Cuadros del A.8 (página 206) al A.11 (página 207) del Anexo A (página 203). Un resumen de estos resultados se puede ver en el Cuadro 5.16 (página 145), donde se muestran los mejores resultados de cada participante en cada idioma.

Cuadro 5.15: Participantes y runs por cada idioma del AVE 2007.

	Alemán	Inglés	Español	Portugués	Total
Fernuniversität in Hagen (FUH)	2				2
U. Évora				1	1
Iasi		1			1
DFKI		2			2
INAOE			2		2
U. Alicante		2			2
Proyecto Text Mess		2			2
U. Jaén			2		2
UNED		1	1		2
Total	2	8	5	1	16

Cuadro 5.16: Resumen resultados del AVE 2007. Mejores valores de F obtenidos para cada sistema e idioma

	Alemán	Inglés	Español	Portugués
Fernuniversität in Hagen	0.72			
DFKI		0.55		
INAOE			0.53	
U. Alicante		0.39		
Proyecto Text-Mess		0.36		
Iasi		0.34		
UNED		0.34	0.47	
U. Jaén			0.37	
U. of Évora				0.68
Baseline 100 % SI	0.4	0.19	0.37	0.6
Baseline 50 % SI	0.34	0.18	0.32	0.46

En esta edición se pudo observar que todos los sistemas menos uno obtuvieron mejores resultados que los “baselines” propuestos al evaluar la validación de respuestas. Esta mejora en resultados respecto a la edición anterior pudo ser causada tanto por la disponibilidad en esta edición de colecciones de desarrollo en todos los idiomas, como por el hecho de que se contaba con la experiencia de la edición

anterior.

Por otro lado, en los Cuadros del A.12 (página 207) al A.15 (página 208) del Anexo A (página 203) se muestran los resultados en cuanto a la evaluación de la selección de respuestas. En estos Cuadros se han mezclado los resultados de los sistemas de AV y de QA (participantes en el QA@CLEF 2007) sobre el subconjunto de preguntas consideradas en el AVE 2007. En el Cuadro 5.17 (página 146) se muestra un resumen de estos resultados, mostrándose para cada sistema y cada idioma el mejor resultado y pudiéndose ver también los resultados de los dos mejores sistemas de QA de cada idioma. Además, en estos Cuadros se muestran también los resultados de los "baselines" propuestos en la sección 4.2.1 (página 109): la selección perfecta de respuestas y un sistema que valida todas las respuestas y selecciona aleatoriamente una por pregunta (baseline random, el cuál se corresponde con la medida *random_qa_accuracy* de la sección 4.2.1 de la página 109).

Cuadro 5.17: Resumen resultados del AVE 2007. Mejores valores de qa_accuracy obtenidos para cada sistema e idioma. Adicionalmente se presentan los valores de qa_accuracy de los dos mejores sistemas de QA de cada idioma.

	Alemán	**Inglés**	**Español**	**Portugués**
Selección perfecta	0.54	0.3	0.59	0.74
DFKI		0.21		
Iasi		0.21		
U. Alicante		0.19		
Fernuniversität in Hagen	0.5			
Mejor sistema QA	0.35	0.18	0.49	0.61
INAOE			0.45	
UNED		0.16	0.42	
U. Jaén			0.41	
U. of Évora				0.44
Proyecto Text-Mess		0.15		
Segundo sistema QA	0.32	0.13	0.38	0.41
Baseline random	0.28	0.1	0.25	0.44

Exceptuando portugués, donde sólo hubo un participante en el AVE 2007, en cada idioma hay sistemas de AV capaces de conseguir más de un 70 % de la selección perfecta. De hecho, los mejores sistemas de AV en alemán e inglés obtuvieron mejores resultados que los sistemas de QA, consiguiendo un 93 % de la selección perfecta en el caso de alemán.

Estos resultados son una muestra de la mejora que podría suponer en QA la in-

corporación de módulos de AV para realizar la selección de respuestas. De hecho, estos resultados suponen también una prueba de que la combinación de distintos sistemas de QA entre si (lo que se denomina un sistema multi-flujo de QA), utilizando un módulo de AV para realizar la selección de la respuesta final podría lograr mejorar los resultados de los sistemas individuales de QA.

5.5.3. Resultados AVE 2008

En la edición de 2008 participaron nueve grupos (el mismo número que en la edición de 2007) en cinco idiomas distintos (alemán, inglés, castellano, francés y rumano) con un total de 24 runs. En el Cuadro 5.18 (página 147) se muestran los grupos participantes y el número de runs enviados por cada grupo y para cada idioma. De nuevo fueron español e inglés los idiomas con mayor participación, con 6 y 8 runs respectivamente.

Cuadro 5.18: Participantes y runs por cada idioma del AVE 2008.

	Alemán	**Inglés**	**Español**	**Francés**	**Rumano**	**Total**
Fernuniversität in Hagen (FUH)	2					2
LIMSI				2		2
Iasi		2			2	4
DFKI	1	1				2
INAOE			2			2
U. Alicante		1	2			3
UNC		2				2
U. Jaén		2	2	2		2
LINA				1		1
Total	3	8	6	5	2	24

Además, la participación en el AVE 2008 mostró evidencias del creciente interés en sistemas de Validación de Respuestas por parte de los participantes en la tarea de QA del CLEF, ya que 6 de los 9 participantes de esta edición tomaron también parte en la tarea principal de QA. De hecho, dos de los participantes del AVE 2008 hicieron uso de sus sistemas de validación (Glöckner, 2008b; Téllez-Valero et al., 2009) como componentes de sus sistemas de QA (Hartrumpf et al., 2008; Téllez-Valero et al., 2009), mejorando de este modo sus resultados.

Los Cuadros del A.16 (página 209) al A.20 (página 210) del Anexo A (página 203), muestran los resultados en cuanto a *precisión, cobertura* y *F* sobre las respuestas correctas en cada idioma. Es decir, los resultados en cuanto a la evaluación de la validación de respuestas. Un resumen de estos resultados se puede ver

en el Cuadro 5.19 (página 148), donde se muestran los mejores resultados de cada participante en cada idioma.

Cuadro 5.19: Resumen resultados del AVE 2008. Mejores valores de F obtenidos para cada sistema e idioma

	Alemán	**Español**	**Francés**	**Inglés**	**Rumano**
DFKI	0.61			0.64	
LIMSI			0.61		
LINA			0.51		
U. Alicante		0.44		0.49	
Fernuniversität in Hagen	0.39				
INAOE		0.39			
UNC				0.21	
Iasi				0.19	0.23
Baseline 100 % SI	0.21	0.18	0.45	0.14	0.20
Baseline 50 % SI	0.19	0.17	0.37	0.13	0.19
U. Jaén		0.06	0.08	0.02	

De nuevo, en esta edición hubo una gran cantidad de sistemas capaces de mejorar los resultados de los "baselines" propuestos. De hecho, en algunos idiomas como alemán, español e inglés los mejores sistemas superaron ampliamente a los "baselines" propuestos.

Los Cuadros del A.21 (página 211) al A.25 (página 213) del Anexo A (página 203), muestran los resultados en cuanto a la evaluación de la selección de respuestas, mezclando sistemas de AV y de QA ordenados de acuerdo a *estimated_qa_performance*. Estos Cuadros contienen también los valores obtenidos para las medidas *qa_accuracy*, *%_mejor_combinación* (la cuál se corresponde con la medida *normalized_qa_accuracy* de la sección 4.2.1 de la página 109 traducida a porcentaje), *qa_rej_accuracy* y *qa_accuracy_max*. Los valores de *qa_accuracy* y *estimated_qa_performance* son iguales en el caso de los sistemas de QA puesto que se considera que estos sistemas responden a todas las preguntas (y por tanto *qa_rej_accuracy* vale 0). Además, hay que tener en cuenta que no se pueden comparar los resultados entre distintos idiomas pero si con los del baseline *random* dado en cada idioma, así como con los resultados del mejor sistema de QA de cada idioma. En el Cuadro 5.20 (página 149) se muestra un resumen de estos resultados, mostrándose para cada sistema y cada idioma el mejor resultado. Además, en el Cuadro 5.20 se pueden ver también los resultados de los dos mejores sistemas de QA de cada idioma.

La interpretación gráfica de estos Cuadros se muestra en las Figuras de la A.1 (página 214) a la A.5 (página 216) del Anexo A (página 203). En estos gráficos

Cuadro 5.20: Resumen resultados del AVE 2008. Mejores valores de estimated_qa_accuracy obtenidos para cada sistema e idioma. Adicionalmente se presentan los valores de estimated_qa_accuracy de los dos mejores sistemas de QA de cada idioma

	Alemán	**Español**	**Francés**	**Inglés**	**Rumano**
Selección perfecta	0.77	0.85	0.73	0.56	0.65
DFKI	0.52			0.34	
Mejor sistema QA	0.38	0.54	0.47	0.21	0.22
U. Alicante		0.37		0.27	
INAOE		0.34			
LIMSI			0.32		
LINA			0.29		
Iasi				0.24	0.25
UNC				0.17	
Segundo sistema QA	0.37	0.25	0.19	0.17	0.19
Fernuniversität in Hagen	0.32				
Baseline random	0.11	0.11	0.33	0.09	0.10
U. Jaén		0.06	0.04	0.01	

el valor de *qa_accuracy_max* es 1 para la selección perfecta, lo cuál se corresponde con una selección perfecta de respuestas correctas (si hay alguna) para cada pregunta y la detección de todas las preguntas que no tienen ninguna respuesta correcta. Sin embargo, el valor de *estimated_qa_performance* de la selección perfecta no es 1 puesto que se asume que las preguntas sin respuesta correcta detectadas en *qa_rej_accuracy* serán respondidas con un acierto igual al expresado por *qa_accuracy* en dicho "baseline". Este valor de *qa_accuracy* representa el acierto de la mejor combinación de los sistemas de QA que toman parte, acierto que no es perfecto (es decir, no es 1).

En tres idiomas (alemán, inglés y rumano) hubo al menos un sistema de AV con mejores resultados (respecto a *qa_accuracy*) que el mejor sistema de QA del mismo idioma. Por otro lado, en los idiomas en donde el mejor valor de *qa_accuracy* no fue obtenido por un sistema de AV, el mejor sistema de QA tuvo un resultado de más de un 50 % mejor que el de los demás sistemas de QA. Si se considera a un sistema de AV como un selector de las respuestas candidatas de un sistema multi-flujo de QA, entonces los sistemas de AV tienen un comportamiento similar al de un conjunto de clasificadores. En aprendizaje automático, se espera que la combinación de un conjunto de clasificadores tenga mayor precisión que un clasificador individual excepto en el caso de que un elemento del conjunto mejore los resultados de los demás sistemas en un alto porcentaje (Dietterich, 1997). Por tan-

to parece obvio que se debe trabajar para conseguir una mejor selección en estos casos.

5.6. Análisis de las medidas

Las tres ediciones celebradas del AVE permitieron estudiar el comportamiento de las medidas propuestas en el Capítulo 4 (página 105) para evaluar sistemas de AV, así como las situaciones para las cuáles son más útiles cada una de ellas. A lo largo de los siguiente subapartados se muestran las principales conclusiones que se alcanzaron para cada grupo de medidas.

5.6.1. Evaluación de la Validación de Respuestas

Las medidas propuestas para evaluar la validación de respuestas se centraron en evaluar la detección de respuestas correctas y solamente estas, midiendo la *precisión*, *cobertura* y su media armónica (*medida F*) sobre las respuestas correctas. Estas medidas se utilizan cuando se desea identificar a los sistemas con mejor rendimiento en cuanto a su habilidad para detectar las respuestas correctas y solamente éstas. Además, estas medidas permiten realizar una comparación con un hipotético sistema de QA que no realiza una etapa de validación. Sin embargo, estas medidas no permiten estudiar la mejora que supondría el uso de un módulo de AV dentro de un sistema de QA.

5.6.2. Evaluación de la Correcta Selección de Respuestas

Para evaluar el comportamiento de sistemas de AV que realizan selección de respuestas se propuso a partir del AVE 2007 el uso de *qa_accuracy* (Fórmula (4.4) de la página 110)), la cuál mide el acierto de un sistema de AV que selecciona como mucho una respuesta por pregunta.

La mayor contribución del uso de esta medida fue que permitió comparar los resultados de los sistemas de AV con los de los sistemas QA, de modo que se pudieron estudiar las posibles mejoras en rendimiento que se podrían obtener en QA al incorporar módulos de AV. Sin embargo, la evaluación con *qa_accuracy* no tiene en cuenta la capacidad de los sistemas de AV para detectar preguntas para las cuáles no se ha dado ninguna respuesta correcta, lo cuál se observó en la evaluación realizada en el AVE 2007 y motivó el desarrollo de nuevas medidas.

5.6.3. Evaluación del Rendimiento Potencial de Sistemas de QA con Módulos de AV

La evaluación realizada en el AVE 2007 sobre la selección de respuestas mostró que las medidas propuestas hasta entonces no tenían en cuenta la capacidad de los sistemas de AV detectando preguntas para las cuáles no se había dado ninguna respuesta correcta. Por este motivo, en el AVE 2008 se propuso el uso de

qa_rej_accuracy (Fórmula (4.7) de la página 110) para medir la correcta detección de preguntas que no tienen ninguna respuesta correcta.

Dado que la detección de estas preguntas sin respuestas correctas podría permitir la mejora de resultados si se permitiese buscar nuevas respuestas a estas preguntas, se propuso una combinación de *qa_rej_accuracy* y *qa_accuracy* que dio lugar a *estimated_qa_performance* (Fórmula (4.9) de la página 111). Haciendo uso de *estimated_qa_performance* se estima que las preguntas para las cuáles no se han encontrado respuestas correctas (representadas por *qa_rej_accuracy*) serán respondidas con el acierto observado anteriormente (el cuál está representado por *qa_accuracy*).

Durante el AVE 2008 se pudo observar que los rankings a los que da lugar *estimated_qa_performance* son muy similares a los generados utilizando *qa_accuracy*. Sin embargo, hay ocasiones en las cuáles hay discrepancias entre ambas medidas. Por ejemplo, en la evaluación realizada en inglés en el AVE 2008 se dieron dos casos que muestran el comportamiento de *estimated_qa_performance* frente a *qa_accuracy*. Ambos casos se pueden ver en el Cuadro 5.21 (página 152), que es un fragmento del Cuadro A.24 (página 213), y se comentan a continuación:

1. En el primer caso se tiene que el sistema *Iftene_2* obtiene mejores resultados que el sistema *Ferrández* de acuerdo a *qa_accuracy* (0.24 de *Iftene_2* por 0.19 de *Ferrández*). Esto indica que *Iftene_2* es mejor que *Ferrández* a la hora de seleccionar respuestas. Sin embargo, el sistema *Ferrández* es mejor de acuerdo con *estimated_qa_performance* (ya que obtiene un valor de 0.27 y el sistema *Iftene_2* obtiene 0.24). Esto significa que si se considera la posible mejora de rendimiento que se podría alcanzar cuando se detecta que todas las respuestas a una pregunta son incorrectas y se solicitan nuevas respuestas al sistema de QA, entonces el rendimiento del sistema *Ferrández* es mejor que el del sistema *Iftene_2*, ya que el sistema *Ferrández* es mejor que el sistema *Iftene_2* detectando preguntas para las cuáles no se ha dado ninguna respuesta correcta (lo cuál se refleja en que el sistema *Ferrández* tiene un valor de 0.4 de *qa_rej_accuracy*, mientras que el sistema *Iftene_2* tiene 0.01). Por tanto, el sistema *Ferrández* podría ayudar en mayor medida que el sistema *Iftene_2* a mejorar los resultados de un sistema de QA. Además, se puede ver cómo el ranking de acuerdo a *estimated_qa_performance* es más similar al obtenido con la *medida F* (mostrado en el Cuadro A.19 de la página 210), la cuál tiene en cuenta la precisión de un sistema a la hora de detectar respuestas incorrectas.

2. En el segundo caso se tiene que el sistema de QA *dfki_1* obtiene mejores resultados que el sistema de AV *Castillo_2* de acuerdo a *qa_accuracy* (0.17 del sistema *dfki_1* por 0.16 del sistema *Castillo_2*). Sin embargo, de acuerdo con *estimated_qa_performance* los dos sistemas tienen el mismo resultado (un valor de 0.17). De nuevo esto indica que los sistemas de AV que detectan las preguntas para las cuáles no se ha dado ninguna respuesta correcta pueden

dar lugar a un mejor rendimiento en QA, y por tanto esta capacidad ha de ser tenida en cuenta.

Cuadro 5.21: Resultados de algunos sistemas de AV y QA en inglés en el AVE 2008: (1) estimated_qa_performance, (2) qa_accuracy (%_mejor_combinación), (3) qa_rej_accuracy, (4) qa_accuracy_max

Grupo	Sistema	Tipo de sistema	(1)	(2)	(3)	(4)
U. Alicante	Ferrández	AV	0.27	0.19 (57.41 %)	0.4	0.59
Iasi	Iftene_2	AV	0.24	0.24 (70.37 %)	0.01	0.25
UNC	Castillo_2	AV	0.17	0.16 (46.30 %)	0.1	0.26
DFKI	dfki_1	QA	0.17	0.17 (50 %)	0	0.17

Por tanto parece claro que *estimated_qa_performance* aporta más información que *qa_accuracy* a la hora de evaluar sistemas de AV que realizan selección de respuestas debido a que *estimated_qa_performance* tiene en cuenta la habilidad de un sistema a la hora de detectar preguntas para las cuáles no se ha dado ninguna respuesta correcta. De este modo se consigue dar una estimación más realista del rendimiento que se puede obtener en QA al utilizar módulos de AV si en caso de no encontrarse respuestas correctas a una pregunta, se solicitan nuevas respuestas.

5.7. Análisis de las Técnicas Utilizadas

Son varias las técnicas y enfoques que se han aplicado por parte de los participantes a lo largo de las tres ediciones del AVE. Los Cuadros del 5.22 (página 153) al 5.24 (página 155) muestran las distintas técnicas utilizadas por los participantes de cada edición.

El hecho de que la tarea propuesta en el AVE 2006 fuera similar a la desarrollada en los RTE Challenges provocó que hubiese un alto número de participantes provenientes de dicha tarea que utilizasen los sistemas presentados al RTE-2 (Bar-Haim et al., 2006) haciendo algunos pequeños cambios en algunos casos. Sin embargo, estos participantes hicieron notar tras la evaluación que el hecho de que las colecciones del AVE fueran creadas directamente a partir de la salida real de sistemas de QA hacía a esta tarea más compleja que la de los RTE Challenges puesto que los sistemas tenían que tratar en ocasiones con algún grado de ruido, como por ejemplo errores sintácticos (Zanzotto and Moschitti, 2006). Hubo un participante que tuvo en cuenta este hecho y realizó un procesamiento previo con el fin de tratar de corregir este tipo de errores, consiguiendo mejorar de este modo sus resultados (Glöckner, 2007b).

Cuadro 5.22: Resumen de las técnicas utilizadas por los participantes en el AVE 2006.

	LCC	**U. Rome**	**ITC-IRST**	**U. Alicante(ofe)**	**U. Alicante(Kozareva)**	**U. Twente**	**U.P. Valencia**	**LIMSI**	**FUH**	**UNED (Herrera)**	**UNED (Rodrigo)**
Lógica	X			X					X		
Procesamiento léxico		X	X	X				X	X X		
Procesamiento sintáctico		X	X	X		X		X	X		
Uso de corpus adicional			X		X				X		
Medidas de solapamiento		X			X	X				X	
Paráfrasis						X		X			
Entidades nombradas											X
Procesamiento semántico									X		

Cuadro 5.23: Resumen de las técnicas utilizadas por los participantes en el AVE 2007.

	Iasi	INAOE	UNED	FUH	U. Évora	DFKI	U. Jaén	U. Alicante	Text-mess
Genera hipótesis	X	X		X				X	X
WordNet	X			X					
Chunking		X			X		X		
n-grams, longest common subsequences		X				X	X	X	X
Transformaciones de frases	X	X							
Entidades nombradas	X	X	X				X		X
Expresiones numéricas	X	X	X	X	X				X
Expresiones temporales			X	X	X				X
Resolución de correferencia				X					
Análisis de dependencias	X					X		X	
Similitud sintáctica	X	X				X		X	
Funciones (sujeto, objeto, etc)	X				X	X			
Transformaciones sintácticas	X								
Desambigüación del sentido de la palabra				X					
Análisis semántico	X			X	X				
Etiquetado de rol semántico				X					
Lógica de primer orden				X	X				
Demostrador de teoremas				X	X				
Similitud semántica	X				X				

Cuadro 5.24: Resumen de las técnicas utilizadas por los participantes en el AVE 2008.

	Iasi	**INAOE**	**FUH**	**DFKI**	**U. Jaén**	**U. Alicante**	**LIMSI**	**LINA**	**UNC**
Genera hipótesis	X					X			
WordNet	X		X		X	X			X
Chunking	X				X		X	X	
n-grams, longest common subsequences				X	X	X			X
Transformaciones de frases	X						X		
Entidades nombradas	X	X	X	X		X	X	X	
Expresiones numéricas	X	X	X	X		X	X	X	
Expresiones temporales	X	X	X				X	X	
Resolución de correferencia									
Análisis de dependencias	X			X			X		
Funciones (sujeto, objeto, etc)	X			X			X		
Transformaciones sintácticas	X						X		
Desambigüación del sentido de la palabra			X						
Análisis semántico	X		X						
Etiquetado de rol semántico			X						
Lógica de primer orden	X		X						
Demostrador de teoremas			X						

5.7.1. Generación Automática de Hipótesis

Los errores debidos a la generación semiautomática de hipótesis del AVE 2006 se eliminaron en las siguientes ediciones (las del AVE 2007 y 2008) al no suministrarse hipótesis en las colecciones de evaluación. Esta modificación trajo consigo una reducción en la participación al surgir el problema de la generación automática de hipótesis, aunque hubo participantes que no necesitaron construir hipótesis al no ser necesarias para el funcionamiento de sus sistemas. De hecho, mientras que en la edición de 2007 la mitad de los participantes (5 de 9) generaron hipótesis, en la de 2008 sólo las generaron 2 participantes de los 9 que hubo en total.

La mayoría de los participantes que decidieron generar hipótesis se basaron en el procedimiento utilizado para crear las hipótesis de las colecciones del AVE 2006 (el cuál fue descrito en la sección 5.4.1 de la página 135), pero realizándolo de forma automática. Para ello, a partir de cada pregunta se creaba automáticamente un patrón de hipótesis partiendo de un patrón genérico (que había sido creado a mano), el cuál dependía de diversas características de la pregunta como el tipo de respuesta esperada, la partícula interrogativa, etc. A continuación, cada patrón era instanciado por las respuestas correspondientes, dando lugar a las distintas hipótesis.

Mientras que algunos participantes se limitaron a crear una hipótesis por respuesta (Ferrández et al., 2007; Wang and Neumann, 2007; Iftene and Balahur, 2008), en la edición de 2007 hubo un participante que construyó dos hipótesis para cada respuesta con el propósito de cubrir los casos de activa y pasiva (Téllez-Valero et al., 2007). Por otro lado y debido a la dificultad para construir una hipótesis textual precisa, Glöckner (2007a) prefirió construir una hipótesis lógica a partir de las formas lógicas de la pregunta y la respuesta.

5.7.2. Procesamiento Realizado

El procesamiento más utilizado por parte de los participantes del AVE se realizó a nivel léxico, principalmente mediante el cálculo de distintas medidas de solapamiento entre palabras o n-gramas. Además, Bosma and Callison-Burch (2007) propusieron en la edición de 2006 el uso de medidas de solapamiento que empleaban paráfrasis de las hipótesis adquiridas de forma automática a partir de corpus paralelo, consiguiendo de este modo mejorar sus resultados. Dado que este método no requiere el uso de ninguna herramienta específica del idioma, la disponibilidad de corpus paralelo en un determinado idioma permite la posibilidad de usar este método para mejorar el rendimiento de un sistema de AV.

El procesamiento sintáctico empezó a ser utilizado ampliamente en el AVE debido principalmente a que los participantes que utilizaron sistemas de los RTE Challenges ya lo tenían incorporado. Este hecho provocó que al no suministrarse hipótesis en las colecciones de evaluación a partir de la edición de 2007, el análisis sintáctico fuera realizado solamente por algunos de los sistemas que generaban hipótesis. En general, los sistemas que trabajaron a este nivel obtuvieron buenos

resultados, dándose en el AVE 2008 el hecho de que los sistemas con los mejores resultados en cada idioma (el grupo *DFKI* en alemán e inglés, el grupo *LIMSI* en francés y el grupo *Iasi* en rumano) excepto en español, realizasen algún tipo de procesamiento sintáctico.

Por otro lado, el procesamiento a nivel semántico no ha sido empleado por demasiados participantes. Aunque en la edición de 2007 se produjo un incremento del número de sistemas (3 en la edición de 2007 en comparación con 1 en la edición de 2006) trabajando a este nivel que invitaba a pensar en que cada vez sería más utilizado, esta tendencia no se cumplió en el AVE 2008 (sólo lo emplearon 2 sistemas). Los sistemas que lo emplearon realizaron sobre todo análisis semántico y etiquetado de rol semántico.

Después de realizar una comparación entre las herramientas utilizadas y los resultados obtenidos, parece que el hecho de utilizar un mayor número de herramientas o realizar un procesamiento más complejo no garantizó la obtención de mejores resultados. Por ejemplo, en el AVE 2008 se puede observar que de acuerdo con el Cuadro 5.24 (página 155) los sistemas del grupo *Iasi* utilizaron más herramientas que los del grupo *DFKI* (*Iasi* utilizó un chunker y un analizador semántico mientras que *DFKI* no). Sin embargo, como se puede ver en los Cuadros A.19 (página 210) y A.24 (página 213), los resultados en inglés de *Iasi* son peores que los obtenidos por *DFKI*. Otro ejemplo similar se puede ver con los sistemas del grupo *FUH*, el cuál utilizó también más herramientas que el grupo *DFKI* (*FUH* utilizó desambigüación del sentido de la palabra, análisis semántico y etiquetado de rol semántico mientras que *DFKI* no). De nuevo los resultados del grupo *DFKI* fueron mejores (mirar Cuadros A.16 de la página 209 y A.21 de la página 211) a pesar de que utilizaba menos herramientas que *FUH*.

Por otro lado, uno de los participantes de la edición de 2006 llevó a cabo un estudio sobre el uso de entidades nombradas en Validación de Respuestas (Rodrigo et al., 2007). El sistema propuesto hacía uso solamente de información sobre entidades nombradas en el texto y la hipótesis, consiguiendo uno de los mejores resultados en español en dicha edición. Este resultado invita a pensar que las entidades nombradas pueden jugar un papel importante en los sistemas de AV. De hecho, en la edición de 2008 se notó un incremento en el uso de información acerca de entidades nombradas, habiendo 7 grupos que las consideraron. En concreto, los participantes que generaron hipótesis fueron los que dieron una mayor importancia a las entidades nombradas (Iftene and Balahur, 2008; Ferrández et al., 2008b; Wang and Neumann, 2008). Estos sistemas usaron la restricción de que para validar una respuesta, todas las entidades nombradas de la hipótesis tenían que estar presentes en el correspondiente texto soporte. Por tanto parece que el reconocimiento de entidades nombradas se estaba empezando a utilizar cada vez más como una importante fuente de información en AV.

Finalmente, a partir de la edición de 2007 hubo sistemas que empezaron a tratar otros tipos de información más relacionada con la tarea de QA como el uso del tipo esperado de respuesta (Téllez-Valero et al., 2007, 2009; Wang and Neumann, 2008; Moriceau et al., 2008; Iftene and Balahur, 2008). Esta información se utilizó

para comprobar si el tipo esperado de respuesta coincidía con el tipo de la respuesta a validar. Mientras que algunos sistemas utilizaron esta información como un atributo más en combinación con otros (Téllez-Valero et al., 2007, 2009), otros sistemas lo utilizaron como una restricción que tenían que cumplir las respuestas para ser validadas (Wang and Neumann, 2008).

5.7.3. Decisión de Validación

Respecto a la decisión final de validación, la mayoría de los participantes hizo uso de técnicas de aprendizaje automático usando como atributos distintas medidas de solapamiento a nivel léxico, sintáctico y/o semántico. A este respecto hubo un participante que en el AVE 2008 incluyó también el uso de características que medían el no solapamiento entre términos de la pregunta, la respuesta y el texto soporte (Téllez-Valero et al., 2009). Estas características mostraron ser más discriminativas que las tradicionales basadas en solapamiento y mejoraron los resultados de su sistema.

En cuanto a los clasificadores utilizados, las máquinas de vectores soporte (en inglés Support Vector Machines, SVM) y los árboles de decisión fueron los clasificadores más empleados. Sin embargo, no parece haber evidencias acerca del mejor rendimiento de uno u otro de estos clasificadores.

Aunque el uso de lógica no ha sido el método más aplicado, los sistemas que lo han empleado junto con conocimiento adicional han obtenido algunos de los mejores resultados de cada edición. En concreto, salvo en alemán en la edición de 2008, los mejores resultados de cada idioma donde algún sistema utilizase lógica fueron obtenidos por un sistema de este tipo (el grupo *LCC* en inglés y español en el AVE 2006, y el grupo *FUH* en alemán en las ediciones de 2006 y 2007).

Sin embargo, este tipo de métodos presentan el inconveniente de que tienen un alto coste computacional. Con el objetivo de reducir este coste y poder utilizar su módulo de AV dentro de un sistema de QA en tiempo real, Glöckner (2008b) decidió comprobar en la edición de 2008 si los textos soporte contenían una respuesta correcta en lugar de comprobar la validez de las respuestas candidatas. Básicamente, la mejora en tiempo se obtenía al contar de antemano con un análisis de la colección de la cuál se extraían los textos soporte, y no tener que realizar este análisis sobre los textos de la colección de evaluación, mientras que en el caso de validar las respuestas si era necesario realizar dicho análisis. Es por ello que algunos errores de este sistema se produjeron cuando a pesar de que el texto soporte contenía una respuesta correcta, la respuesta era incorrecta.

5.8. Recapitulación

En este capítulo se ha propuesto una metodología para evaluar sistemas de AV partiendo de la propuesta de AV basada en RTE que fue realizada en el Capítulo 3 (página 89) y las medidas de evaluación descritas en el Capítulo 4 (página 105).

Esta metodología permite reutilizar el esfuerzo realizado en las evaluaciones de QA al generar preguntas por expertos humanos y realizar manualmente los juicios de las respuestas, facilitando su implantación.

Esta metodología fue puesta en práctica como una tarea de evaluación internacional de sistemas de AV. La tarea se denominó Answer Validation Exercise (AVE) y se desarrolló dentro del CLEF durante las ediciones de 2006, 2007 y 2008 en un total de 10 idiomas distintos y con un total de 16 grupos distintos como participantes. Los recursos generados en varios idiomas fueron utilizados para realizar la evaluación de los sistemas participantes y están disponibles para la comunidad científica, de modo que se puede seguir con el desarrollo y evaluación de sistemas de AV.

Finalmente, las medidas propuestas mostraron ser útiles para evaluar la validación de respuestas, y en el caso de la evaluación de la selección de respuestas permitieron estudiar además el impacto en resultados que supondría la utilización de módulos de AV dentro de los sistemas de QA. En concreto, los resultados obtenidos por los participantes del AVE muestran que realizando la selección de preguntas por medio de sistemas de AV se podrían mejorar los resultados actuales en QA, especialmente si se combinan varios sistemas de QA para que formen un sistema multi-flujo. Esta mejora puede ser además incrementada si se aprovecha la capacidad de los sistemas de AV para detectar las preguntas para las cuáles no se ha dado ninguna respuesta correcta. En estos casos, los sistemas de AV podrían solicitar nuevas respuestas para estas preguntas, lo que permitiría la obtención de respuestas correctas en una segunda oportunidad.

Capítulo 6

Evaluación de Sistemas de Búsqueda de Respuestas que Incorporan Validación de Respuestas

En el Capítulo 5 (página 125) se evaluó el impacto que podría suponer la incorporación de módulos de AV dentro de los sistemas de QA. Para ello, se evaluaron los resultados que se obtendrían al utilizar sistemas de AV que llevan a cabo la selección de la respuesta final a una pregunta de entre las respuestas candidatas generadas por uno o varios sistemas de QA. Además, al realizar esta evaluación se tuvo en cuenta la habilidad de los sistemas de AV para detectar las preguntas para las cuáles no hay ninguna respuesta candidata correcta. Con este propósito, se estimaron los resultados que se obtendrían si se solicitasen nuevas respuestas a estas preguntas.

Sin embargo, la capacidad de los sistemas de AV de detectar preguntas para las cuáles no se encuentra ninguna respuesta correcta tiene otras utilidades dentro de los sistemas de QA. Esta funcionalidad sería de gran utilidad en escenarios donde se prefiere tener preguntas sin responder a preguntas respondidas incorrectamente. Este interés puede darse en escenarios en los cuáles una respuesta incorrecta tiene asociado un riesgo o un coste alto, como por ejemplo en diagnóstico médico. En este tipo de escenarios, los sistemas de AV se podrían utilizar para tomar la decisión sobre si responder o no a una determinada pregunta para la cuál se han encontrado una serie de respuestas candidatas. Sin embargo, a la posibilidad de indicar que se prefiere no responder a una pregunta porque no se está seguro de encontrar una respuesta correcta no se le ha prestado suficiente atención en QA.

En este capítulo se propone una nueva medida de evaluación para evaluar sistemas de QA que prefieren no responder a dar una respuesta incorrecta. Es decir, sistemas que pretenden reducir la cantidad de respuestas incorrectas y para ello dejan sin responder a las preguntas para las cuáles no están seguros de poder en-

contrar una respuesta correcta.

En este capítulo se explora la motivación para permitir a los sistemas de QA no responder, así como la atención que se ha prestado anteriormente a la evaluación de este comportamiento. A continuación se realiza la propuesta de una medida de evaluación, llamada *c@1*, para el escenario propuesto, justificándose los motivos por los cuáles se desechan otras posibles medidas. Además, se realiza también un estudio empírico acerca de la confianza que se puede depositar en los resultados obtenidos haciendo uso de la medida propuesta en comparación con otras medidas de QA. Finalmente se estudia la aplicación de *c@1* en una evaluación real, el ResPubliQA 2009 celebrado dentro del marco del CLEF, donde se observaron las ventajas de utilizar *c@1* para la evaluación de sistemas de QA a los que se permite no responder.

6.1. Permitiendo no Responder a los Sistemas de Búsqueda de Respuestas

En muchos de los escenarios en los cuáles se utilizan sistemas de QA es preferible tener respuestas incorrectas a tener preguntas sin responder, las cuáles se suelen considerar también como respuestas incorrectas. Sin embargo, hay otros escenarios en los cuáles el usuario prefiere no obtener respuesta a tener una respuesta incorrecta. Esto sucede, por ejemplo, en escenarios en los cuáles una respuesta incorrecta tiene asociado un determinado riesgo o consecuencias indeseables, como por ejemplo en diagnóstico médico, donde es preferible no obtener respuestas a obtener respuestas incorrectas, puesto que éstas últimas podrían acarrear consecuencias negativas y muy graves.

Este capítulo se centra en un escenario de QA en el cuál es preferible no obtener respuestas a obtener respuestas incorrectas. En concreto, el escenario de QA que se propone tiene las siguientes características:

- Para cada pregunta se puede devolver como máximo una sola respuesta
- Un sistema puede decidir no responder a una pregunta en caso de no estar seguro de poder encontrar una respuesta correcta a esa pregunta
- Dejar sin contestar una pregunta tiene más valor que responderla incorrectamente, pero en ningún caso el hecho de no responder supone una respuesta correcta

Hay que aclarar que el hecho de que un sistema decida no responder a una pregunta porque no está seguro de poder encontrar una respuesta correcta es distinto de reconocer que la pregunta no tiene respuesta. El reconocimiento de preguntas sin respuesta se propuso en algunas evaluaciones de QA como el TREC o el CLEF, en las cuáles no se garantizaba que todas las preguntas tuvieran respuesta en la colección de documentos en la cuál se realizaba la búsqueda (Voorhees, 2001a;

Magnini et al., 2003). Estas preguntas se denominaban NIL, y cuando un sistema devolvía como respuesta la cadena de texto NIL a una de ellas (indicador de que el sistema considera que la pregunta no tiene respuesta), la respuesta se consideraba correcta. Por tanto, el hecho de no responder en el escenario propuesto indica que no se conoce la respuesta (y en ningún caso se considera como una respuesta correcta, pero tampoco incorrecta). Sin embargo, responder con la cadena de texto NIL significa que la pregunta no tiene respuesta, de modo que NIL se considera una respuesta correcta si la pregunta no tiene respuesta, y en caso contrario se considera incorrecta.

El primer paso para realizar la evaluación dentro de este escenario consiste en permitir a los sistemas de QA la posibilidad de no responder. De este modo, un sistema de QA tiene dos opciones dada una pregunta: responder o no responder. En el caso de las preguntas respondidas, las respuestas devueltas podrán ser evaluadas como correctas o incorrectas[1]. Además, hay que recordar que el hecho de no responder no constituye en ningún caso una respuesta correcta, pero tampoco se considera una respuesta incorrecta. Este escenario tiene la tabla de contingencia mostrada en el Cuadro 6.1 (página 163), donde:

n_{ac}: número de preguntas a las cuáles se ha respondido correctamente
n_{aw}: número de preguntas a las cuáles se ha respondido incorrectamente
n_u: número de preguntas que se han dejado sin responder
n: número total de preguntas ($n = n_{ac} + n_{aw} + n_u$)

Cuadro 6.1: Tabla de contingencia asociada a un escenario de QA donde se permite dejar preguntas sin responder

	Correcta (C)	**Incorrecta ($\neg$C)**
Respondida (A)	n_{ac}	n_{aw}
No Respondida ($\neg$A)	n_u	

6.2. Trabajo Relacionado

La medida de evaluación que se propone en este capítulo sirve para evaluar la confianza de un sistema de QA sobre la corrección de sus respuestas. Es decir, evaluar el siguiente comportamiento: si el sistema de QA considera que todas la respuestas candidatas a una pregunta son incorrectas, entonces toma la decisión de no responder a dicha pregunta con el objetivo de disminuir la cantidad de respuestas incorrectas que genera.

[1]Para simplificar el escenario propuesto se ha decidido descartar los juicios de *inexacta* y *no soportada*

A lo largo de las evaluaciones de QA ha habido algunas medidas orientadas a evaluar este comportamiento como ya se mencionó en la sección 2.3.3 (página 52). A continuación se describen los motivos por los cuáles se considera que las medidas propuestas anteriormente no son adecuadas para el escenario planteado.

Aunque *accuracy* (Fórmula (6.1)) no está orientada a evaluar la confianza de los sistemas en sus respuestas, es interesante estudiarla puesto que es una medida frecuentemente utilizada en QA y que muestra la poca atención que ha recibido la posibilidad de dejar preguntas sin responder en QA. *Accuracy* sólo premia las respuestas correctas, de tal modo que una pregunta sin responder tiene el mismo valor que una pregunta respondida incorrectamente. De hecho, en las evaluaciones realizadas con *accuracy* es mejor dar una respuesta, aunque se crea que puede ser incorrecta, que no responder. Esto sucede porque cabe la posibilidad de que la respuesta emitida sea realmente correcta y mejore el valor final de *accuracy*.

$$accuracy = \frac{n_{ac}}{n} \tag{6.1}$$

La primera vez que se evalúo a sistemas de QA a los cuáles se les permitía no responder fue en el TREC 2001 (Voorhees, 2001a). Aunque la principal medida de evaluación fue *MRR* (medida descrita en la sección 2.3.3.1 de la página 52), también se evalúo a los sistemas participantes por medio del porcentaje de preguntas que respondían y la proporción de estas preguntas que respondían correctamente. Sin embargo, no se propuso ninguna combinación de estos dos valores en uno solo.

Una medida en la que se podría pensar para ser utilizada en el escenario propuesta es *CWS*, la cuál fue descrita en la sección 2.3.3.5 (página 53) y permite evaluar la confianza de los sistemas en sus respuestas. Sin embargo, al utilizar *CWS* no hay ninguna manera de indicar que un sistema decide no responder a una determinada pregunta.

Otras medidas utilizadas para evaluar la confianza de un sistema de QA en sus respuestas son *K* y *K1*, las cuáles fueron descritas en la sección 2.3.3.6 (página 54). Ambas medidas están basadas en una función de utilidad, de modo que un sistema podría indicar que decide no responder cuando se da el valor de confianza 0 a una respuesta. Sin embargo, tal y como ya se indicó cuando se describieron ambas medidas, no está clara la interpretación del valor final de *K* y *K1*.

Dado que las medidas de evaluación estudiadas no parecen adecuadas cuando se permite no responder, parece necesario proponer una medida que tenga en cuenta esta posibilidad.

6.3. Considerando Preguntas sin Responder en la Evaluación

En esta sección se estudia la incorporación a la evaluación de sistemas de QA de la posibilidad de no responder preguntas. Para realizar dicho estudio, primero se propone una función de utilidad que tiene en cuenta a las preguntas sin responder.

Sin embargo, se observa que esta función de utilidad no da un valor razonable (en el sentido de que no se considera la habilidad de un sistema decidiendo si devuelve o no una respuesta) a las preguntas sin responder, motivo por el cuál se define una nueva medida de evaluación: *c@1*. Finalmente, se estudian otras estimaciones para las preguntas sin responder, observando que *c@1* es la que ofrece el comportamiento más razonable.

6.3.1. Estudio de una Función de Utilidad que contempla Preguntas sin Responder

Como primer paso para definir una medida de evaluación para el escenario propuesto, se realizó el estudio de una función de utilidad que considera a las preguntas que no han sido respondidas. La función de utilidad propuesta, denominada *U(i)*, otorga a cada pregunta *i* uno de los siguientes valores:

- -1 si la pregunta ha sido respondida incorrectamente
- 0 si la pregunta no ha sido respondida
- 1 si la pregunta ha sido respondida correctamente

Si se consideran *n* preguntas, la función de utilidad resultante es la mostrada en la Fórmula (6.2). El comportamiento de esta medida es sencillo: el hecho de no responder no añade ningún valor, mientras que las respuestas incorrectas añaden valores negativos y las respuestas correctas añaden valores positivos. Al realizar el cálculo sobre un conjunto de preguntas, un resultado final positivo significa que ha habido más respuestas correctas que incorrectas. Por el contrario, un resultado final negativo indica que ha habido más respuestas incorrectas que correctas.

$$UF = \frac{1}{n}\sum_{i=1}^{n} U(i) = \frac{n_{ac} - n_{aw}}{n} \tag{6.2}$$

Para poder interpretar mejor el peso que la función de utilidad está otorgando a las preguntas sin responder se tiene que transformar la Fórmula (6.2) en otra que permita considerar a estas preguntas como un parámetro. Para preservar el ranking creado por la Fórmula (6.2), se realiza una transformación monótona aplicando la función f(x)=0.5x+0.5. El resultado obtenido al realizar la transformación es el mostrado en la Fórmula (6.3), de modo que la Fórmula (6.3) da lugar al mismo ranking que la Fórmula (6.2).

$$\begin{aligned} 0{,}5\frac{n_{ac} - n_{aw}}{n} + 0{,}5 = \frac{0{,}5}{n}\left[n_{ac} - n_{aw} + n\right] = \\ = \frac{0{,}5}{n}\left[n_{ac} - n_{aw} + n_{ac} + n_{aw} + n_u\right] = \frac{0{,}5}{n}\left[2n_{ac} + n_u\right] = \\ = \frac{n_{ac}}{n} + 0{,}5\frac{n_u}{n} \end{aligned} \tag{6.3}$$

El primer sumando de la Fórmula (6.3) se corresponde con *accuracy* (Formula (6.1) de la página 164), mientras que el segundo sumando se corresponde con 0.5 veces la proporción de preguntas que se dejaron sin responder (representadas por n_u/n). Es decir, en caso de utilizarse la Fórmula (6.3), las preguntas sin responder contribuyen al resultado final con el mismo valor que contribuirían si la mitad de ellas hubiesen sido respondidas correctamente. De acuerdo con este resultado, el hecho de no responder está siendo premiado en la Fórmula (6.3) en la misma proporción para todos los sistemas sin tener en cuenta el rendimiento mostrado sobre las preguntas que han sido respondidas. Esto no parece razonable debido a que no se tiene en cuenta la capacidad de un sistema decidiendo si responder o no a una pregunta, ya que las preguntas sin responder se valorarán siempre igual. En los siguientes apartados se trata este problema.

6.3.2. Intuición para el Valor de las Preguntas sin Responder

El razonamiento realizado en la sección anterior para interpretar la función de utilidad propuesta sugería que las preguntas sin responder contribuían al resultado final en la misma proporción que contribuirían si la mitad de ellas hubiesen sido respondidas correctamente. Generalizando este resultado se obtiene la siguiente hipótesis, la cuál se va a considerar durante el resto del capítulo:

Las preguntas sin responder reciben el mismo valor que obtendrían éstas si una determinada proporción de ellas hubiesen sido respondidas correctamente

Esta hipótesis sugiere que a la hora de evaluar las preguntas sin responder, se podría considerar que éstas se van a responder en una segunda oportunidad. Es decir, el sistema de QA considera que ninguna de las respuesta candidatas que ha encontrado para una determinada pregunta es correcta, lo cuál podría llevarle a no contestar a la pregunta y buscar nuevas respuestas con la esperanza de encontrar una respuesta correcta en una segunda oportunidad. Por tanto, lo que se necesita para realizar la evaluación en el escenario propuesto es una estimación de la *probabilidad de tener una respuesta correcta teniendo en cuenta también a las preguntas sin responder*. De acuerdo con la tabla de contingencia del Cuadro 6.1 (página 163), esta probabilidad se expresaría como se muestra en la Fórmula (6.4).

$$\begin{aligned} P(C) &= P(C \cap A) + P(C \cap \neg A) = \\ &= P(C \cap A) + P(C/\neg A) * P(\neg A) \end{aligned} \tag{6.4}$$

En la Fórmula (6.4), $P(C \cap A)$ puede ser estimado mediante n_{ac}/n (que es el valor de *accuracy* mostrado en la Fórmula (6.1) de la página 164), mientras que $P(\neg A)$ puede ser estimado por n_u/n. Por tanto queda por estimar $P(C/\neg A)$.

El valor que se le suele otorgar a $P(C/\neg A)$ en las evaluaciones de QA es 0, de modo que una pregunta sin responder recibe el mismo valor que una pregunta respondida incorrectamente. Sin embargo, la hipótesis sobre la cuál se trabaja en este capítulo se basa en otorgar a $P(C/\neg A)$ un valor distinto de 0. En el caso de

la función de utilidad estudiada en la sección 6.3.1 (página 165), la Fórmula (6.3) (página 165) correspondería a $P(C)$ cuando $P(C/\neg A)$ tiene el valor 0.5.

Siguiendo esta intuición está claro que la medida para realizar la evaluación en el escenario propuesto ha de tener dos componentes: el *accuracy* global y una estimación acerca de la corrección de las preguntas que han sido dejadas sin responder. Por ejemplo, si se utilizase la medida representada por la Fórmula (6.3), se estaría suponiendo que las preguntas sin responder contribuyen al resultado final en la misma medida que contribuirían si la mitad de ellas hubiesen sido respondidas correctamente. Dado que, como ya se ha indicado anteriormente, no parece razonable otorgar el mismo valor para todos los sistemas (en este caso 0.5), en el próximo apartado se propone una estimación para $P(C/\neg A)$ que da lugar a una medida más razonable.

6.3.3. La Medida Propuesta: c@1

Tras el razonamiento realizado en el apartado anterior, lo que se necesita para definir una medida de evaluación en este caso es una estimación acerca de la corrección de las preguntas que han sido dejadas sin responder. De acuerdo con la hipótesis de trabajo establecida en la sección 6.3.2 (página 166), se considera que estas preguntas podrían ser respondidas en un segundo ciclo. Dado que en el primer ciclo ya se ha observado la capacidad respondiendo preguntas correctamente (capacidad representada por $P(C \cap A)$ = n_{ac}/n), se propone utilizar esta observación para estimar $P(C/\neg A)$ en lugar de utilizar un valor arbitrario (como por ejemplo 0.5).

Haciendo uso de esta estimación se propone la medida *c@1* (*correctness at one*), definida por la Fórmula (6.5).

$$c@1 = \frac{n_{ac}}{n} + \frac{n_{ac}}{n}\frac{n_u}{n} = \frac{1}{n}(n_{ac} + \frac{n_{ac}}{n}n_u) \tag{6.5}$$

Las características principales de *c@1* son:

1. Un sistema de QA que responde a todas las preguntas obtiene el mismo resultado utilizando *c@1* que utilizando *accuracy* puesto que n_u valdría 0 y por tanto se tendría que *c@1*=n_{ac}/n.

2. Las preguntas que no han sido respondidas añaden al resultado final un valor equivalente al que obtendrían estas preguntas si hubiesen sido respondidas con el *accuracy* observado. La principal consecuencia de esta estimación es que no todos los sistemas reciben el mismo valor, sino que los sistemas que son capaces de dar más respuestas correctas recibirán más valor. Esto parece razonable ya que los mejores sistemas son también los que han mostrado un mejor rendimiento a la hora de detectar cuándo una respuesta suya era correcta.

3. Un sistema que no devuelve ninguna respuesta recibe el valor 0 ya que n_{ac}=0 en los dos sumandos de *c@1*.

De acuerdo con estas características se puede interpretar *c@1* en términos de probabilidad como $P(C)$ (Fórmula (6.4) de la página 166), estimando $P(C/\neg A)$ como $P(C \cap A)$. En el siguiente apartado se estudian otras posibles estimaciones para $P(C/\neg A)$ distintas del valor global de *accuracy* observado (valor dado por $P(C \cap A)$ = n_{ac}/n), comprobándose que ninguna de estas estimaciones es mejor que la utilizada en *c@1*.

6.3.4. Otras estimaciones para P(C/¬ A)

Este apartado se centra en estudiar el resultado de utilizar en la Fórmula (6.4) (página 166) otras estimaciones para $P(C/\neg A)$ distintas de la utilizada en *c@1* (el valor $P(C \cap A)$). A lo largo de las distintas propuestas se ve que no hay otra estimación capaz de dar una medida razonable cuando se tienen en cuenta a las preguntas sin responder. Para realizar estas estimaciones se asume que no se sabe nada acerca de la corrección de las preguntas sin responder. Es decir, no se sabe si en el caso de que el sistema de QA haya encontrado una respuesta y haya decidido descartarla por no estar seguro de su validez, si esa respuesta era en realidad correcta o no.

Las estimaciones que se proponen para $P(C/\neg A)$ son los valores mínimo y máximo que pueden observarse (0 y 1 respectivamente), así como otros valores que se pueden obtener a partir de la tabla de contingencia mostrada en el Cuadro 6.1 (página 163). Estas estimaciones son:

1. $P(C/\neg A) \equiv 0$
2. $P(C/\neg A) \equiv 1$
3. $P(C/\neg A) \equiv P(\neg C/\neg A) \equiv 0{,}5$
4. $P(C/\neg A) \equiv P(C/A)$
5. $P(C/\neg A) \equiv P(\neg C/A)$

En los próximos subapartados se procede a analizar el resultado de utilizar cada una de estas estimaciones.

6.3.4.1. P(C/¬ A)≡ 0

Al utilizar esta estimación se considera que el hecho de no responder tiene el mismo valor que el hecho de dar una respuesta incorrecta, es decir, no vale nada. Como consecuencia se tiene que $P(C)$ tiene el mismo valor que *accuracy* (n_{ac}/n), lo cuál se puede ver en la Fórmula (6.6).

$$P(C) = P(C \cap A) + P(C/\neg A) * P(\neg A) = P(C \cap A) \qquad (6.6)$$

Utilizando esta estimación no se cumple la premisa del escenario planteado en este capítulo de que el hecho de no responder tiene más valor que responder incorrectamente. Por tanto esta estimación queda descartada.

6.3.4.2. P(C/¬ A)≡ 1

Utilizando esta estimación se considera que todas las preguntas sin responder serán respondidas correctamente en una segunda oportunidad, obteniendo para $P(C)$ lo mostrado en la Fórmula (6.7). De este modo, un sistema que no responda a ninguna pregunta obtendría un resultado perfecto, lo cuál no tiene sentido. Por tanto no es razonable utilizar esta estimación y queda descartada.

$$P(C) = P(C \cap A) + P(C/\neg A) * P(\neg A) = P(C \cap A) + P(\neg A) \qquad (6.7)$$

6.3.4.3. P(C/¬ A)≡ P(¬ C/¬ A)≡ 0.5

Un argumento para considerar esta estimación podría ser que al no tener ninguna observación sobre la corrección de las preguntas sin responder, se podría asumir la misma probabilidad para $P(C/\neg A)$ y $P(\neg C/\neg A)$. En este caso $P(C)$ se correspondería con la Fórmula (6.8), la cuál da lugar a la Fórmula (6.3) (página 165) que ya se discutió en la sección 6.3.1 (página 165). Como ya se observó anteriormente, en este caso se está dando a las preguntas sin responder un valor constante y arbitrario que es independiente del rendimiento del sistema, lo cuál no parece razonable.

$$P(C) = P(C \cap A) + P(C/\neg A) * P(\neg A) = P(C \cap A) + 0{,}5 * P(\neg A) \quad (6.8)$$

Hay además otros argumentos en contra de utilizar esta estimación y, por tanto, de utilizar la función de utilidad definida en la sección 6.3.1 (página 165) por la Fórmula (6.2) (página 165), la cuál produce el mismo ranking. Utilizando la Fórmula (6.2), un sistema que diese un mayor número de respuestas incorrectas que de respuestas correctas obtendría un valor menor que 0. Hay que tener en cuenta que 0 es el valor que obtendría un sistema que dejase sin contestar a todas las preguntas. Sin embargo, en el escenario planteado se desea premiar a los sistemas que no responden con el propósito de reducir la cantidad de respuestas incorrectas, no al mero hecho de dejar preguntas sin responder. Es por ello que esta estimación queda descartada.

6.3.4.4. P(C/¬ A)≡ P(C/A)

Al hacer uso de esta estimación, las preguntas sin responder reciben el mismo valor que la precisión observada sobre las preguntas respondidas, lo cuál está representado por $P(C/A)$= n_{ac}/(n_{ac}+ n_{aw}). En este caso el valor de $P(C)$ sería el mostrado en la Fórmula (6.9).

$$\begin{aligned} P(C) = P(C \cap A) + P(C/\neg A) * P(\neg A) = \\ = P(C/A) * P(A) + P(C/A) * P(\neg A) = P(C/A) = \frac{n_{ac}}{n_{ac} + n_{aw}} \end{aligned} \qquad (6.9)$$

El resultado final que se obtiene es la precisión observada sobre las preguntas que han sido respondidas. Sin embargo, ésta no es una medida razonable ya que daría el máximo valor posible a un sistema que decidiese dejar todas las preguntas sin responder salvo una de la cuál estuviese seguro que es correcta. Mediante el uso de esta medida no se promovería mucho avance dentro del escenario propuesto, ya que sería fácil engañar a la medida de evaluación y a cambio sólo se conseguiría una respuesta correcta.

Además, si se estima que $P(C/\neg A)$ es igual a $P(C/A)$, se está considerando que el sistema de QA decide de forma aleatoria si responder o no responder (ya que la probabilidad de que la respuesta sea correcta es la misma en ambos casos). Sin embargo, en el escenario propuesto se quiere premiar a los sistemas que eligen no responder porque son capaces de detectar que las respuestas que ha encontrado a una pregunta son incorrectas. Por tanto, esta estimación queda también descartada.

6.3.4.5. P(C/¬ A)≡ P(¬ C/A)

La última estimación propuesta consiste en considerar que las preguntas sin responder contribuyen al resultado final en la misma proporción en la cuál el sistema de QA falla dando respuestas. Es decir, utilizar la proporción de preguntas respondidas incorrectamente. Estimando $P(C/\neg A)$ como $P(\neg C/A) \equiv n_{aw}$ / (n_{ac}+ n_{aw}), $P(C)$ se transformaría en la Fórmula (6.10).

$$\begin{aligned} P(C) &= P(C \cap A) + P(C/\neg A) * P(\neg A) = \\ &= P(C \cap A) * P(\neg C/A) * P(\neg A) = \\ &= \frac{n_{ac}}{n} + \frac{n_{aw}}{n_{ac} + n_{aw}} * \frac{n_u}{n} \end{aligned} \tag{6.10}$$

Es muy fácil obtener buenos resultados de acuerdo con esta medida sin tener en realidad un buen rendimiento. De hecho, se puede obtener un valor cercano al máximo respondiendo incorrectamente sólo una pregunta y dejando sin responder a las demás. Por tanto, esta estimación queda descartada.

6.4. Evaluación de c@1

Como se mencionó en la sección 2.6 (página 76), además de conocer lo que evalúa una determinada medida, es interesante conocer también la confianza que se puede depositar en los resultados obtenidos con esa medida. Esta confianza se refiere a lo seguro que se puede estar al obtener conclusiones sobre qué sistema es mejor al comparar distintos sistemas utilizando esa medida.

Para realizar un estudio sobre la confianza que se puede depositar en los resultados obtenidos utilizando *c@1* en comparación con los de *accuracy* (medida ampliamente utilizada en las evaluaciones de QA), se hizo uso del método descrito

en Buckley and Voorhees (2000) para estudiar la estabilidad y el poder de discriminación de *c@1*, y el método descrito en Voorhees and Buckley (2002) para comprobar la sensibilidad de la medida propuesta. Ambos métodos fueron descritos en la sección 2.6.2 (página 81) y utilizados para comparar medidas de evaluación de AV en la sección 4.3 (página 111).

Hay que tener en cuenta que los resultados obtenidos en estos experimentos no pretenden mostrar que una medida sea mejor que la otra. Está claro que cada medida sirve para evaluar un aspecto distinto y que la elección de la medida se ha de realizar en función de los objetivos de evaluación establecidos. Lo que se pretende mediante este estudio es mostrar la confianza que puede depositar un investigador sobre los resultados obtenidos por su sistema al utilizar *c@1* en comparación con los de *accuracy*.

En los próximos apartados se describen primero los datos que se utilizan para realizar el estudio y a continuación se muestran y comentan los resultados obtenidos al estudiar la estabilidad y la sensibilidad de *c@1*.

6.4.1. Datos Utilizados

Para realizar los experimentos de estabilidad y sensibilidad se utilizaron las colecciones y runs del ResPubliQA 2009[2], el cuál se celebró dentro del marco del CLEF (Peñas et al., 2010). El ResPubliQA 2009 fue una evaluación de QA en la cuál los sistemas participantes tenían que devolver respuestas a un conjunto de 500 preguntas. La respuesta tenía que ser un párrafo extraído de la colección JRC-Acquis[3], que recoge documentación de la Unión Europea. La tarea se propuso de modo que para cada pregunta siempre había una respuesta correcta en la colección JRC-Acquis, por lo que no existían preguntas NIL.

En cada pregunta un sistema podía tomar la decisión de devolver como máximo una respuesta o no responder si consideraba que no era capaz de encontrar una respuesta correcta para esa pregunta. De este modo, el marco del ResPubliQA 2009 cumple las características del escenario planteado en este capítulo (ver sección 6.1 de la página 162).

Además, en los casos en los cuáles un sistema indicaba que prefería no responder a una pregunta, el sistema podía devolver la respuesta que hubiera emitido si tuviera que responder a todas las preguntas. Es decir, para las preguntas que se dejaban sin responder, los sistemas podían devolver la respuesta que consideraban que era la más prometedora pero que había sido descartada posteriormente por no estar seguros acerca de su corrección. Estas respuestas hipotéticas fueron también evaluadas en el marco del ResPubliQA con el fin de comprobar el rendimiento de la validación realizada por los sistemas participantes.

La principal medida de evaluación del ResPubliQA 2009 fue *c@1*, utilizándose *accuracy* como medida de evaluación secundaria. Para el cálculo de *accuracy*

[2] http://celct.isti.cnr.it/ResPubliQA/

[3] http://wt.jrc.it/lt/Acquis/

se tuvo en cuenta las respuestas devueltas para las preguntas que no se habían respondido, de modo que *accuracy* no hacía distinción entre preguntas respondidas y preguntas no respondidas, sino que sólo tenía en cuenta preguntas respondidas correctamente o preguntas respondidas incorrectamente. De este modo, los resultados del ResPubliQA permiten comparar a *c@1* y *accuracy* sobre los mismos datos.

La tarea se propuso en nueve idiomas distintos (búlgaro, inglés, francés, alemán, italiano, portugués, rumano, español y vasco), utilizando las mismas preguntas en todos los idiomas. De este modo se pudo realizar una comparación de los resultados entre distintos idiomas, aunque teniendo en cuenta que no era posible realizar una comparación estricta debido a diversos factores como distintas traducciones de las preguntas, distintas características de los idiomas, etc.

En cuanto a participación, en el ResPubliQA 2009 hubo 44 runs en distintos idiomas. Para el estudio realizado en este capítulo se han utilizado todos estos runs sin hacer distinción entre idiomas[4].

6.4.2. Estabilidad y Poder de Discriminación

Para comparar la estabilidad y el poder de discriminación de *c@1*, *accuracy* y *UF* (la función de utilidad definida por la Fórmula (6.2) de la página 165 y que se decidió incluir en este estudio debido a que se utilizó para definir *c@1*) se utilizó el método propuesto por Buckley and Voorhees (2000), el cuál fue descrito en la sección 2.6.2.1 (página 81) y utilizado en los experimentos de la sección 4.3.3 (página 115). Hay que recordar que la estabilidad hace referencia al error asociado a la conclusión *el sistema X es mejor que el sistema Y*, de modo que cuanto más estable es una medida menor es el error. Además, el método de Buckley and Voorhees (2000) permite calcular también el poder de discriminación de una medida, de modo que cuanto más discriminativa es la medida, menos empates habrá entre sistemas y menor será la diferencia de resultados requerida para concluir qué sistema es mejor.

Para realizar los experimentos se utilizó el algoritmo de la Figura 4.2 (página 115) para obtener los datos necesarios para calcular la tasa de error (Fórmula (2.28) de la página 83). Dicha tasa se utiliza para medir la estabilidad (a menor tasa de error más estable es una medida de evaluación). Además, los datos obtenidos por el algoritmo sirvieron también para calcular la proporción de empates (Fórmula (2.29) de la página 83), que sirve para evaluar el poder de discriminación de una medida (cuanto menor es la proporción de empates mayor es el poder de discriminación).

[4]Dado que el hecho de mezclar datos de distintos idiomas podría modificar los resultados, se decidió realizar el mismo experimento utilizando solamente los datos de inglés, donde hubo 14 runs. Los resultados que se obtuvieron con los datos de inglés fueron muy similares a los conseguidos utilizando todos los runs. Es por ello que se decidió mostrar en este estudio los resultados obtenidos utilizando todos los runs (un total de 44), debido a que los métodos empleados están diseñados para ser más fiables cuantos más datos se utilicen, ya que así es posible realizar más comparaciones entre sistemas

En cuanto a la elección del umbral de equivalencia, al igual que en los experimentos realizados en la sección 4.3.3 (página 115), se decidió variar el umbral de equivalencia desde 0.01 hasta 0.10 (siguiendo el trabajo realizado por Sakai (2007b)) y dibujar para cada medida de evaluación una curva *proporción de empates - tasa de error*.

En la Figura 6.1 (página 173) se muestran las curvas *proporción de empates - tasa de error* para las medidas *c@1*, *accuracy* y *UF* utilizando los datos de los 44 runs y las colecciones del ResPubliQA 2009 con $c = 250$[5]. En la Figura se puede observar cómo la tasa de error de las tres medidas decrece al aumentar el porcentaje de empates. Este decrecimiento se corresponde al incremento del valor del umbral de equivalencia, el cuál produce una menor tasa de error, pero por contra hace aumentar el porcentaje de empates. En la Figura también se puede observar que las curvas de *c@1* y *accuracy* son muy similares, lo cuál indica que ambas medidas tienen una estabilidad y poder de discriminación similares.

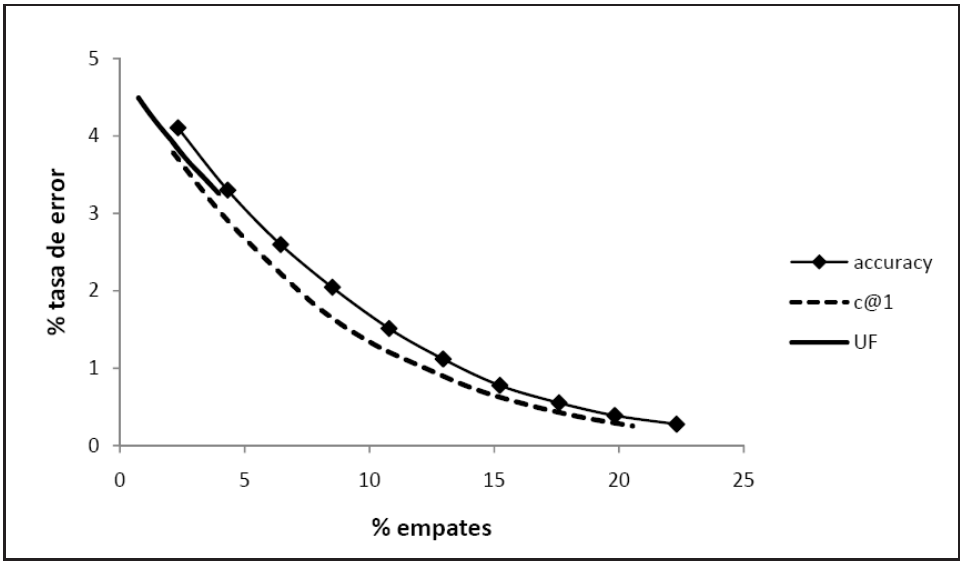

Figura 6.1: Curvas *proporción de empates* / *tasa de error* para *accuracy*, *c@1* y *UF* con c = 250

Por otro lado, mientras que los rangos de valores entre los cuáles se mueven las curvas de *c@1* y *accuracy* son bastantes similares, la curva de *UF* se mueve en un rango menor. Esto significa que los cambios en el valor del umbral de equivalencia no afectan demasiado a la estabilidad y al poder de discriminación de *UF*. Por ejemplo, por mucho que se aumente el umbral de equivalencia, el porcentaje de empates no aumenta demasiado (de hecho no supera el 5 %), lo cuál muestra que *UF* es la más discriminativa de las tres medidas comparadas.

[5]Se decidió utilizar este tamaño para las subcolecciones de los experimentos de esta sección debido a que es el mismo que se utiliza en los experimentos expuestos en la sección 6.4.3 (página 174)

Adicionalmente, se pueden obtener dos interpretaciones distintas al observar las curvas:

1. Si se fija el porcentaje de empates, se obtiene una tasa de error similar para *c@1* y *accuracy*. Si se desease reducir el porcentaje de empates en un experimento con una determinada medida, la consecuencia sería tener una mayor tasa de error, lo cuál significaría tener más errores al concluir que un sistema es mejor que otro.

2. En caso de fijar la tasa de error, también se tiene un porcentaje de empates similar para *c@1* y *accuracy*. Si se desease reducir la tasa de error de un determinado experimento utilizando una determinada medida, como contrapunto se tendría que habría más empates entre sistemas y la diferencia necesaria para concluir qué sistema es mejor sería mayor.

Los resultados obtenidos muestran que las tres medidas comparadas son bastante estables, obteniendo *c@1* y *accuracy* una menor tasa de error que *UF* cuando aumenta el porcentaje de empates. Además se puede ver que *c@1* tiene un comportamiento similar al de *accuracy*, medida de evaluación ampliamente utilizada en QA. De hecho, la tasa de error de *c@1* es ligeramente inferior a la de *accuracy* cuando se tiene el mismo porcentaje de empates. Por otro lado, los resultados obtenidos en estos experimentos son similares a los conseguidos por otras medidas de evaluación de QA en otros estudios similares (Sakai, 2007a). Por tanto este estudio muestra que en los resultados obtenidos utilizando *c@1* se puede depositar la misma confianza que en los resultados obtenidos por otras medidas de evaluación utilizadas en QA.

6.4.3. Sensibilidad

Otra forma de evaluar la confianza que se puede depositar en los resultados obtenidos por una medida de evaluación es el método propuesto por Voorhees and Buckley (2002), denominado método de swap, descrito en la sección 2.6.2.2 (página 84) y utilizado con medidas de evaluación de AV en la sección 4.3.4 (página 117). La idea principal de este método es la de contar el número de veces en las cuáles dos runs difieren en cuanto a qué sistema es mejor, condicionada al tamaño de la diferencia de los resultados de los sistemas. Para aplicar este método se utilizó el algoritmo de la Figura 4.4 (página 118).

Dado que para realizar el experimento es necesario que los subconjuntos que se utilizan en cada comparación (Q_i y $Q_i^{'}$ en el algoritmo de la Figura 4.4 de la página 118) sean disjuntos entre si, su tamaño puede ser como mucho de hasta la mitad del tamaño de la colección de la que se parte. Es decir, puesto que las colecciones del ResPubliQA 2009 tenían 500 preguntas, el tamaño utilizado en los experimentos fue de c=250[6].

[6]Se utilizó el mismo tamaño de subcolecciones en los experimentos de la sección 6.4.2 (página 172) por cuestiones de homogeneidad

El Cuadro 6.2 (página 175) muestra los resultados obtenidos tras aplicar el método de swap a las medidas *accuracy*, *c@1* y *UF*, con c=250. Las distintas medidas han sido ordenadas en el Cuadro de acuerdo a los valores de la columna (iv), que muestra la sensibilidad para una confianza del 95 %. La interpretación de los resultados del Cuadro para *c@1* es la siguiente:

- De acuerdo con la columna (i), se necesita una diferencia de 0.09 entre los resultados de distintos sistemas para concluir que un sistema es mejor que otro con una tasa de error menor al 5 % (confianza del 95 %) cuando se utiliza *c@1* como medida de evaluación.
- De entre todos los valores de *c@1* observados durante el experimento el mayor fue de 0.77 como muestra la columna (ii).
- La diferencia necesaria para concluir qué sistema es mejor (mostrada en la columna (i)), en términos relativos a la máxima diferencia observada en el experimento (columna (ii)) es de 11.69 % como muestra la columna (iii). Este valor se haya dividiendo el resultado de la columna (i) entre el de la columna (ii).
- De entre todas las comparaciones realizadas utilizando *c@1*, un 58.40 % de ellas cumple la diferencia requerida que se muestra en la columna (i). Este porcentaje refleja la sensibilidad de *c@1* y aparece en la columna (iv).

Cuadro 6.2: Resultados obtenidos tras aplicar el método de swap a *accuracy*, *c@1* y *UF* con un nivel de confianza del 95 % y con c=250: (i) Diferencia requerida para concluir que un sistema es mejor que otro con el nivel de confianza establecido; (ii) Máximo valor obtenido durante los experimentos; (iii) Diferencia requerida para ver qué sistema es mejor relativa al máximo rendimiento observado ((i) / (ii)); (iv) Porcentaje de comparaciones realizadas en el experimento que cumplen la diferencia requerida (sensibilidad)

	(i)	**(ii)**	**(iii)**	**(iv)**
UF	0.17	0.48	35.12 %	59.30 %
c@1	0.09	0.77	11.69 %	58.40 %
accuracy	0.09	0.68	13.24 %	55.00 %

Los resultados mostrados en el Cuadro 6.2 (página 175) muestran que la sensibilidad de las tres medidas comparadas es bastante similar, siendo además estos resultados parecidos a los obtenidos para otras medidas de QA en otros estudios (Sakai, 2007a). De hecho, en el Cuadro 6.2 se puede observar que aunque la diferencia requerida para concluir si un sistema es mejor que otro utilizando *c@1* y *accuracy* es la misma (0.09), el porcentaje de comparaciones que cumplen esta diferencia en los experimentos realizados es levemente mayor con *c@1* que con

accuracy. Esto significa que *c@1* tiene un poder de discriminación ligeramente superior al de *accuracy*. Por tanto, estos resultados indican que la confianza que se puede depositar en *c@1* en cuanto a la discriminación de sistemas es similar a la de otras medidas de evaluación utilizadas actualmente en QA.

6.5. Caso de Estudio

Además del estudio realizado sobre la confianza que se puede depositar en *c@1*, también se realizó un estudio sobre la interpretación de los resultados obtenidos por sistemas de QA dentro de un escenario real utilizando *c@1*. El principal objetivo del estudio era comparar los resultados obtenidos utilizando *c@1*, con los obtenidos utilizando *accuracy* sobre datos reales con el propósito de determinar situaciones en las cuáles *c@1* puede aportar más información que *accuracy*.

Para realizar este estudio se utilizaron los resultados obtenidos por los sistemas participantes en el ResPubliQA 2009. Del análisis de los resultados de estos sistemas hay dos situaciones que muestran claramente las ventajas de *c@1*.

La primera de estas situaciones se refleja en el Cuadro 6.3 (página 176), donde se muestran los resultados de los sistemas *loga092de* y *base092de* respecto a *c@1*, detallando para cada sistema el número de preguntas respondidas correctamente, las respondidas incorrectamente y las que se han dejado sin responder. Se puede observar que ambos sistemas respondieron correctamente un número similar de preguntas (187 el sistema *loga092de* y 189 el sistema *base092de*). Sin embargo, los resultados de acuerdo con *c@1* son diferentes. De hecho, el sistema *loga092de* obtiene mejores resultados que el sistema *base092de* a pesar de que éste último respondió correctamente 2 preguntas más[7].

Cuadro 6.3: Resultados de algunos sistemas participantes en el ResPubliQA 2009 que obtuvieron un número similar de respuestas correctas pero distintos valores de *c@1*: (i) número de preguntas respondidas correctamente; (ii) número de preguntas respondidas incorrectamente; (iii) número de preguntas sin responder

Sistema	c@1	(i)	(ii)	(iii)
loga092de	0.44	187	230	83
base092de	0.38	189	311	0

Esta diferencia en resultados se debe a que el sistema *base092de* está respondiendo incorrectamente más preguntas que el sistema *loga092de*, el cuál prefiere no contestar a determinadas preguntas ya que no está seguro acerca de la corrección de sus respuestas. De este modo, estos resultados muestran que *c@1* es capaz

[7]Hay que tener en cuenta que de acuerdo con los resultados obtenidos en la sección 6.4.3 de la página 174, la diferencia observada entre estos dos sistemas no es significativa (la diferencia observada es 0.06 y la diferencia estimada para concluir qué sistema es mejor es 0.09). Sin embargo, se decidió utilizar este ejemplo debido a que muestra claramente una de las virtudes de *c@1*

de premiar las situaciones en las cuáles un sistema mantiene el número de respuestas correctas, y para lograr reducir las respuestas incorrectas decide no responder a determinadas preguntas.

El segundo resultado a destacar en este estudio se muestra en el Cuadro 6.4 (página 177), donde se pueden ver los resultados de los sistemas *icia091ro* y *uaic092ro* utilizando *c@1* y *accuracy*[8], el número de preguntas respondidas correctamente, el número de preguntas respondidas incorrectamente y el número de preguntas que se dejaron sin responder. Además, se indica para las preguntas sin responder el número de ocasiones en las cuáles la hipotética respuesta devuelta con fines de evaluar la validación fue correcta o incorrecta.

Cuadro 6.4: Resultados de algunos sistemas participantes en el ResPubliQA 2009 que obtuvieron el mismo valor de *accuracy* pero distinto valor de *c@1*: (i) número de preguntas respondidas correctamente; (ii) número de preguntas respondidas incorrectamente; (iii) número de preguntas sin responder; (iv) número de preguntas sin responder donde la hipotética respuesta devuelta fue evaluada como correcta; (v) número de preguntas sin responder donde la hipotética respuesta devuelta fue evaluada como incorrecta

Sistema	**c@1**	**accuracy**	**(i)**	**(ii)**	**(iii)**	**(iv)**	**(v)**
icia091ro	0.58	0.47	237	156	107	0	107
uaic092ro	0.47	0.47	236	264	0	0	0

El Cuadro 6.4 muestra que los sistemas *icia091ro* y *uaic092ro* tienen un número similar de respuestas correctas y el mismo valor de *accuracy*, pero un valor distinto de *c@1*. El motivo por el cuál se tiene esta diferencia respecto a *c@1* es el mismo que el que se tenía en el ejemplo anterior (el mostrado en el Cuadro 6.3 de la página 176). Es decir, la reducción de respuestas incorrectas manteniendo el número de respuestas correctas.

Sin embargo, en este caso se desea recalcar el hecho de que los valores de *accuracy* son similares porque todas las hipotéticas respuestas que da el sistema *icia091ro* a las preguntas que deja sin responder son incorrectas. Es decir, el sistema *icia091ro* ha utilizado un criterio muy bueno para decidir qué preguntas deja sin responder, logrando de este modo reducir el número de respuestas incorrectas mientras mantiene el número de respuestas correctas. Como consecuencia, ambos sistemas tienen el mismo valor de *accuracy*, pero *c@1* da un mejor resultado al sistema *icia091ro* que al sistema *uaic092ro*. Por tanto, este ejemplo muestra que en aquellos escenarios donde es posible dejar preguntas sin responder con el objetivo de reducir la cantidad de respuestas incorrectas, *c@1* discrimina mejor que

[8]Cabe recordar que como se indicó en la sección 6.4.1 de la página 171, para calcular *accuracy* se tuvo en cuenta la corrección de las respuestas devueltas a las preguntas que se dejaron sin responder. Es decir, *accuracy* no distingue los casos en los que se elige responder de los casos en los cuáles no se elige responder

accuracy al comparar sistemas de QA, permitiendo detectar los enfoques más prometedores.

6.6. Recapitulación

En este capítulo se ha planteado que los módulos de AV pueden ser utilizados dentro del contexto de un sistema de QA para decidir dado un conjunto de respuestas candidatas a una pregunta si se responde o no a la pregunta. Es decir, si el sistema de AV considera que entre las respuestas candidatas no hay ninguna correcta, entonces elige no responder a esa pregunta con el propósito de reducir la cantidad de respuestas incorrectas. Esta capacidad de los sistemas de AV puede ser interesante en escenarios de QA donde una respuesta incorrecta tiene un coste alto, como por ejemplo en diagnóstico médico, en los cuáles puede ser preferible no obtener respuesta a tener una respuesta incorrecta. De este modo, el escenario que se ha planteado en este capítulo se caracteriza por permitir a un sistema de QA no responder a una pregunta si no está seguro de encontrar una respuesta correcta, y dar más valor a las preguntas sin responder que a las preguntas respondidas incorrectamente.

Para evaluar sistemas de QA que deciden dejar preguntas sin responder con el propósito de reducir la cantidad de respuestas incorrectas, en este capítulo se ha definido una nueva medida llamada *c@1*. La hipótesis de trabajo que se ha utilizado para proponer esta medida se basa en que *las preguntas sin responder reciben el mismo valor que recibirían si una determinada proporción de ellas hubiesen sido respondidas correctamente*. En el caso de *c@1* está proporción la representa el rendimiento conseguido respondiendo preguntas correctamente, es decir, el valor global de *accuracy* observado.

Siguiendo este planteamiento, un sistema de QA que responda a todas las preguntas recibe el mismo resultado utilizando *c@1* que utilizando *accuracy*. En cambio, si el sistema deja sin responder algunas preguntas entonces: (1) se calcula el valor global de *accuracy*; (2) este valor de *accuracy* se asigna a la proporción de preguntas que se han dejado sin responder; y (3) *c@1* se calcula sumando el valor del paso (1) al valor obtenido en el paso (2). De este modo, *c@1* recompensa a los sistemas que mantienen el número de respuestas correctas y convierten preguntas respondidas incorrectamente en preguntas sin responder. Por tanto, en el escenario propuesto se ha planteado el uso de *c@1* en lugar de *acurracy* debido a que esta última medida no tiene en cuenta a las preguntas sin responder, considerándolas como si hubiesen sido respondidas incorrectamente.

Como alternativa a *c@1* se ha estudiado también el uso de una función de utilidad que tenga en cuenta a las preguntas sin responder. Sin embargo, esta función de utilidad no otorga un valor apropiado a las preguntas sin responder, las cuáles se premian en la misma proporción en todos los sistemas mediante un valor arbitrario. Se considera que este valor arbitrario no es apropiado ya que no tiene en cuenta la eficacia del sistema decidiendo si se ha sido capaz de encontrar una respuesta co-

rrecta o no, de modo que se premie en mayor medida el hecho de dejar preguntas sin responder cuando la eficacia decidiendo si una respuesta es o no correcta es mayor. Esta eficacia está representada por el rendimiento observado respondiendo preguntas correctamente, ya que un mayor número de respuestas correctas indica que el sistema ha sido capaz de detectar cuándo una respuesta suya era correcta. Este valor es el que se utiliza en *c@1*, de modo que *c@1* permite detectar en mayor medida que la función de utilidad a los sistemas que muestran una mayor capacidad para decidir si se ha encontrado una respuesta correcta. De este modo, al dejar algunas preguntas sin responder, estos sistemas reducen la cantidad de respuestas incorrectas.

Además de definir *c@1*, en este capítulo se ha llevado a cabo también un estudio acerca de la confianza que se puede depositar en los resultados obtenidos utilizando esta medida de evaluación. El estudio ha mostrado que *c@1* es similar a otras medidas de evaluación utilizadas en QA en cuanto a su poder de discriminación y su estabilidad.

Finalmente, cabe destacar que la medida ha sido utilizada para realizar la evaluación de los sistemas participantes en el ResPubliQA 2009 celebrado en el marco del CLEF. En esta evaluación *c@1* mostró que en escenarios en los cuáles es preferible no obtener respuestas a tener respuestas incorrectas, es más adecuada que *accuracy* para detectar los enfoques más prometedores.

Capítulo 7

Conclusiones

Esta tesis se ha centrado en el desarrollo de una metodología de evaluación de sistemas de Validación de Respuestas que tienen como objetivo mejorar los resultados en Búsqueda de Respuestas. El primer paso ha consistido en la observación de los resultados actuales en evaluaciones de QA, cuyo análisis muestra aspectos que podrían mejorarse con el fin de obtener un mejor rendimiento por parte de este tipo de sistemas. Las principales observaciones realizadas fueron:

- Los rankings de respuestas que se devuelven contienen respuestas incorrectas que provocan que se consigan peores resultados de los que se podrían obtener si se realizase un filtrado efectivo de las respuestas incorrectas.

- Se ha podido observar que los distintos sistemas de QA se complementan entre si, de modo que aunque individualmente alcanzan resultados similares, en conjunto logran responder correctamente a un mayor número de preguntas. Esta observación tiene como consecuencia que una combinación efectiva de estos sistemas supondría una mejora en resultados en comparación con los sistemas individuales actuales.

- La arquitectura en cascada utilizada típicamente en QA provoca una alta dependencia entre módulos que no permite aprovechar todo el rendimiento de cada una de las módulos de un sistema de QA. Mediante un proceso de retroalimentación en el cuál el sistema de QA fuese capaz de conocer que ha generado una respuesta incorrecta, cada módulo podría reconfigurarse y realizar un nuevo procesamiento, lo cuál podría permitir reducir su dependencia con el resto de módulos. La consecuencia de este comportamiento podría ser la obtención de una respuesta correcta, lo cuál supondría la mejora de los resultados.

Una aproximación para afrontar estos problemas la representa el uso de sistemas de AV, motivo por el que este trabajo se ha centrado en la evaluación de este tipo de sistemas.

En este capítulo se muestran las principales conclusiones que se han obtenido a lo largo de este trabajo. Estas conclusiones hacen referencia al modelo de AV propuesto en el Capítulo 3 (página 89), las medidas de evaluación que se describieron en el Capítulo 4 (página 105), la metodología de evaluación mostrada en el Capítulo 5 (página 125), los recursos necesarios para la evaluación propuesta y la puesta en práctica de la metodología como una tarea de evaluación (el Answer Validation Exercise, AVE). Finalmente, se muestran las líneas de investigación que se han abierto y se pretenden abordar en trabajos futuros.

7.1. Modelo de Validación de Respuestas

La mayoría de los sistemas de AV desarrollados antes de este trabajo realizaban un procesamiento sencillo basado principalmente en la detección de redundancia de palabras entre la pregunta, la respuesta y alguna fuente de información externa, normalmente la Web, sin tener en cuenta las relaciones semánticas que deben existir entre una pregunta y una respuesta para que ésta sea considerada correcta. Sin embargo, este tipo de sistemas disminuyen su rendimiento al aumentar la complejidad de las preguntas, lo cuál contribuye también a disminuir la mejora que pueden aportar en QA.

Dado que la tarea de RTE está orientada a analizar las relaciones semánticas entre dos fragmentos de texto (llamados texto e hipótesis), el modelo presentado propone que los sistemas de AV realicen un mayor análisis de las relaciones entre la pregunta y la respuesta. Para comprobar la validez de este enfoque, a partir de la salida de sistemas reales de QA se construyó una colección (denominada SPARTE) de pares texto-hipótesis enfocada a la tarea de AV. El análisis de esta colección mostró que el enfoque propuesto era viable.

Este modelo se ha utilizado como enfoque inicial para la definición de la metodología de evaluación propuesta en el Capítulo 5 (página 125), donde además se ha puesto este modelo a disposición de los desarrolladores de sistemas de AV, algunos de los cuáles lo ha empleado. Por tanto, este modelo supone una aproximación al desarrollo de sistemas de AV. Sin embargo, hay que tener en cuenta que este modelo requiere la disponibilidad de un sistema de RTE y llevar a cabo la generación automática de hipótesis, proceso que en algunos casos podría dar lugar a errores al tomar la decisión de validación.

7.2. Metodología de Evaluación

Una vez mostrado el modelo de sistemas de AV basados en RTE, en el Capítulo 5 (página 125) se ha propuesto una metodología para realizar la evaluación de sistemas de AV. Uno de los motivos para desarrollar esta metodología lo constituye la observación de que las evaluaciones previas sobre este tipo de sistemas no aportaban información sobre la mejora que podría suponer el uso de módulos de AV en QA. De este modo, uno de los objetivos de la metodología propuesta consistía

no solo en estudiar el rendimiento de los sistemas de AV para comparar distintos enfoques entre si, sino también en evaluar el impacto en resultados que supondría su incorporación en QA.

La definición de esta metodología se realizó partiendo del modelo propuesto en el Capítulo 3 (página 89) y utiliza las medidas de evaluación del Capítulo 4 (página 105). Además, esta metodología se puso en práctica dentro de una tarea de evaluación internacional denominada Answer Validation Exercise (AVE), la cuál se celebró durante tres ediciones en el marco del CLEF. La experiencia de cada edición sirvió para ir refinando la metodología, de modo que su versión final queda a disposición de la comunidad científica y puede ser utilizada para realizar la evaluación de nuevos sistemas de AV, ampliándose si se desea con nuevas medidas de evaluación.

Finalmente, cabe mencionar que los organizadores de los RTE Challenges consideran que dentro de su comunidad se han de proponer nuevas metodologías de evaluación para reflejar escenarios más realistas, siguiendo el espíritu del trabajo desarrollado en esta tesis (Giampiccolo et al., 2007).

7.3. Recursos generados para la Evaluación

Durante el trabajo desarrollado en esta tesis se generaron también una serie de recursos que sirven para la evaluación de sistemas de AV. Estos recursos están representados principalmente por las colecciones de evaluación que se crearon en cada edición del AVE y que se han mostrado en el Capítulo 5 (página 125).

En total se desarrollaron colecciones en 10 idiomas distintos, y todas ellas se generaron a partir de la salida de sistemas reales de QA, por lo que estas colecciones suponen un recurso valioso para la evaluación de sistemas de AV en entornos reales.

Existen dos tipos de estas colecciones:

- Las primeras colecciones fueron creadas con un formato similar al de las colecciones utilizadas en las evaluaciones de los RTE Challenges siguiendo el modelo propuesto en el Capítulo 5 (página 125). Estas colecciones permiten que el sistema de AV a ser evaluado no tenga que tener en cuenta el problema de la generación automática de hipótesis. Además, su formato permite que puedan ser utilizadas también para el desarrollo y la evaluación de sistemas de RTE, como por ejemplo la realizada en Moschitti and Zanzotto (2007).

- El resto de las colecciones se crearon sin dar hipótesis, de modo que la evaluación planteada fuese más realista y los sistemas a ser evaluados recibieran la misma entrada que recibirían de un sistema de QA real.

Estos recursos suponen (hasta donde el autor de esta tesis conoce), las primeras colecciones de evaluación de sistemas de AV puestas a disposición de la comunidad científica, de modo que la disponibilidad de estas colecciones permite continuar el desarrollo y evaluación de sistemas de AV.

7.4. Medidas de Evaluación

En el Capítulo 4 (página 105) se describieron una serie de medidas para realizar la evaluación de sistemas de AV. Estas medidas permiten evaluar distintas funcionalidades de un sistema, facilitando de este modo que el investigador pueda centrarse en evaluar el aspecto que considere más relevante. Además de permitir la comparación entre sistemas de AV, algunas de estas medidas permiten también la comparación directa con sistemas de QA, de modo que su uso permite estudiar la mejora de resultados que supondría la incorporación de módulos de AV en QA. Estas medidas se han dividido en dos conjuntos dependiendo de la función que desempeña el módulo de AV que se quiere evaluar: validación o selección.

El desarrollo de las distintas ediciones del AVE sirvió para comprender mejor las medidas de evaluación propuestas, permitiendo conocer sus características principales y el escenario para el cuál es más útil utilizar cada una de ellas. Esta información es de gran importancia debido a que un mayor conocimiento de las medidas de evaluación permite a los investigadores alcanzar un mayor número de conclusiones, y con una mayor fiabilidad, a la hora de realizar e interpretar experimentos.

Al realizar comparaciones entre sistemas utilizando las medidas propuestas en AV, hay que tener en cuenta que los resultados pueden variar entre colecciones distintas. Por tanto, no se deben de realizar comparaciones entre los resultados obtenidos sobre distintas colecciones. Sin embargo, para una misma colección se pueden realizar comparaciones con los “baselines” propuestos.

Por otro lado, en el Capítulo 6 (página 161) se ha tenido en cuenta que el uso de módulos de AV podría ser también de utilidad en sistemas de QA que trabajan en escenarios donde es preferible no responder a dar una respuesta incorrecta, como por ejemplo en diagnóstico médico, para proponer una medida para evaluar sistemas de QA que actúan en este tipo de escenarios.

Los siguientes subapartados describen en mayor detalle los dos grupos de medidas propuestas para evaluar sistemas de AV, así como la medida propuesta para evaluar sistemas de QA en escenarios donde se prefiere tener preguntas sin responder a respuestas incorrectas.

7.4.1. Medidas para Evaluar la Validación

Este grupo de medidas se centran en la evaluación del rendimiento de sistemas de AV cuyo objetivo consiste en eliminar las respuestas incorrectas de entre un conjunto de respuestas candidatas. Las medidas propuestas (*precisión*, *cobertura* y su media armónica, denominada *medida-F*) premian a los sistemas que solamente devuelven respuestas correctas, teniendo en cuenta también que se detecte el mayor número de respuestas correctas. Estas medidas permiten la comparación con un hipotético sistema de QA que no realiza ningún tipo de validación.

Las medidas de precisión y cobertura se utilizan habitualmente para la evaluación de sistemas de IR (Manning et al., 2008) y de IE (Grishman and Sundheim,

1996), y han sido empleadas también para la evaluación de sistemas de AV antes de esta tesis (Magnini et al., 2002a; Tonoike et al., 2004). Sin embargo, en la evaluación de sistemas de AV estas medidas no se habían combinado en una única medida como se ha realizado en esta tesis (utilizando la *medida-F*). Además, el estudio realizado en la sección 4.3 (página 111) ha permitido conocer mejor la confianza que se puede depositar en los resultados obtenidos con estas medidas, y su adecuación a la evaluación en AV.

7.4.2. Medidas para Evaluar la Selección

El segundo grupo de medidas desarrolladas para la evaluación en AV está enfocado a la evaluación de sistemas de AV que seleccionan una o ninguna respuesta para cada pregunta de entre un conjunto de respuestas a la pregunta. Dentro de este grupo se han propuesto medidas para evaluar la correcta selección de respuestas, la detección de preguntas sin ninguna respuesta correcta (y en las que por tanto no se debe de seleccionar ninguna respuesta), y estimar el rendimiento que se obtendría cuando al detectar una pregunta que no tiene ninguna respuesta correcta se decide buscar nuevas respuestas a esa pregunta.

Estas medidas permiten comparar los resultados de sistemas de AV con los de QA. De este modo, haciendo uso de estas medidas se puede estudiar la aportación que pueden realizar los sistemas de AV en QA.

Las medidas propuestas sirven, en principio, para evaluar solamente la selección de respuestas y no se pueden aplicar a la evaluación de otras tareas. Sin embargo, hay que tener en cuenta que antes del desarrollo de esta tesis no se habían utilizado medidas para evaluar sistemas de AV que realizan selección de respuestas. Por tanto, las medidas que se han propuesto en esta tesis suponen el primer aporte al estado del arte en evaluación de selección de respuestas en AV.

7.4.3. Medidas que consideran Preguntas sin Responder

Otra de las medidas propuestas en esta tesis se ha descrito en el Capítulo 6 (página 161). Esta medida (denominada *c@1*) está enfocada a la evaluación de sistemas de QA que actúan en escenarios donde es preferible dejar una pregunta sin responder que responderla incorrectamente. Un módulo de AV se podría utilizar en este tipo de escenarios para que fuese el que tomase la decisión sobre si alguna de las respuestas candidatas obtenidas por el sistema de QA puede ser correcta, o por el contrario no se puede asegurar la corrección de ninguna de estas respuestas, en cuyo caso se toma la decisión de dejar sin responder a la pregunta.

La medida propuesta premia a los sistemas que mantienen el número de respuestas correctas y consiguen disminuir el número de respuestas incorrectas al dejar sin responder las preguntas para las cuáles no está seguro de poder encontrar una respuesta correcta. Para lograr este comportamiento, la medida propuesta premia en mayor medida a las preguntas que se dejan sin responder que a las preguntas respondidas incorrectamente.

Además, esta medida ha sido utilizada en una evaluación competitiva, el ResPubliQA 2009 del CLEF, donde ha mostrado su utilidad a la hora de detectar los enfoques más prometedores en este tipo de escenarios.

La propuesta de *c@1* supone un aporte al estado del arte en la evaluación de sistemas de QA, dentro del grupo de medidas que permiten evaluar la confianza de un sistema en sus respuestas (a este grupo pertenecen medidas como *CWS*, *K* y *K1*), aunque teniendo en cuenta que solo permite evaluar a sistemas de QA que devuelven como máximo una respuesta por pregunta.

7.5. Marco de Evaluación

Al poner en práctica la metodología de evaluación propuesta como una tarea de evaluación (denominada AVE), se abrió a los investigadores en AV la posibilidad de que pudieran contar con un marco real, donde evaluar el resultado de sus investigaciones y compararse sobre datos reales bajo las mismas condiciones con otros grupos. La tarea que se llevó a cabo supuso la primera evaluación de sistemas de AV y logró atraer la atención de investigadores interesados tanto en RTE como en QA.

Las distintas ediciones del AVE fueron proponiendo nuevas tareas a realizar por parte de los participantes, lo cuál permitió que los investigadores fueran incorporando progresivamente nuevas funcionalidades a sus sistemas. De este modo, se permitió que los investigadores pudieran ir probando sus sistemas en un entorno cada vez más real, poniendo cada vez mayor énfasis en mejorar los resultados en QA.

Por otro lado, los resultados obtenidos por los participantes del AVE suponen una importante fuente de información para la comunidad científica debido a la comparación realizada entre ellos y los resultados de sistemas reales de QA (aquellos que tomaron parte en la tarea de QA del CLEF). Estas comparaciones han permitido observar que los resultados en QA se pueden mejorar al incorporar módulos de AV.

De hecho, en el Capítulo 5 (página 125) se pudo ver que la mayoría de los sistemas participantes en el AVE lograron obtener mejores rankings que los generados por un hipotético sistema de QA que no realiza validación. Estos muestran que las tecnologías actuales en sistemas de AV permiten mejorar los resultados de QA a este respecto, el cuál representa uno de los tres aspectos que se observaron en el Capítulo 1 (página 25) que podrían suponer la mejora de resultados en QA. Esta observación ha fomentado que haya habido participantes del AVE que hayan utilizado un módulo de AV en su sistema de QA para realizar el ranking de las respuestas candidatas, consiguiendo de este modo mejorar sus resultados (Téllez-Valero et al., 2009).

Por otro lado, los resultados que se mostraron en el Capítulo 5 al evaluar sistemas de AV que realizan selección, mostraron que la combinación de distintos sistemas en una arquitectura multi-flujo donde la respuesta final es seleccionada

por un módulo de AV puede permitir mejorar los resultados de los sistemas individuales de QA. Por tanto, estos datos sugieren que los resultados en QA se podrían mejorar mediante la combinación de distintos sistemas de QA y la incorporación de un módulo de AV que seleccione la respuesta final, lo cuál fue una intuición de la que se partió también en el Capítulo 1 y que ya han utilizado algunos participantes del AVE en sus sistemas de QA para mejorar sus resultados (Hartrumpf et al., 2008).

Respecto a la mejora en rendimiento que podría obtenerse al permitir en un sistema de QA retroalimentación por medio de un módulo de AV, en la metodología de evaluación propuesta se estimó el rendimiento que podría obtenerse en caso de que un módulo de AV detecte que no se han generado respuestas correctas a una pregunta y se pidan nuevas respuestas al sistema de QA. A pesar de que hay que tener en cuenta que las conclusiones obtenidas en el AVE a este respecto se han de realizar con cuidado al tratarse de una estimación, los resultados han mostrado que este comportamiento podría suponer una mejora de rendimiento en QA. Es de esperar que al igual que los resultados anteriores referentes a la mejora al realizar validación y selección promovieron el desarrollo de sistemas de QA que incorporasen módulos de AV para realizar estas dos tareas, estos resultados sirvan de motivación para incluir también esta funcionalidad en sistemas reales de QA.

Por tanto, el desarrollo de este marco ha permitido la evaluación y comparación de distintos enfoques de AV, contribuyendo también al desarrollo de mejores sistemas de QA.

7.6. Trabajo Futuro

El trabajo desarrollado en esta tesis ha permitido abrir líneas de investigación en las que se pretende seguir trabajando. A continuación se describen las principales líneas de trabajo abiertas:

- *Estimación de los resultados que se obtendrían en QA mediante la incorporación de módulos de AV.* Las medidas propuestas en el Capítulo 4 (página 105) permiten la evaluación de sistemas de AV que realizan tanto validación como selección. Además, algunas de las medidas descritas en este capítulo han servido también para aportar información sobre si un determinado módulo de AV sería de utilidad para mejorar los resultados en QA, lo cuál es el objetivo de todo sistema de AV.

  Una línea de trabajo futuro consiste en estudiar el desarrollo de medidas de evaluación capaces de dar una estimación del rendimiento que alcanzaría un sistema de QA concreto utilizando un determinado módulo de AV. Esta evaluación serviría no sólo para desarrollar módulos de AV de carácter general, sino para el desarrollo de módulos de AV específicos para un determinado sistema de QA. Para realizar esta evaluación se tendría que partir de la evaluación de las respuestas generadas por el sistema de QA en particular, y

probar al sistema de AV sobre ese conjunto de respuestas candidatas para realizar la evaluación.

- *Extender el uso de la medida c@1 a escenarios donde se permite devolver más de una respuesta por pregunta.* El escenario que se ha planteado en el Capítulo 6 (página 161) para utilizar *c@1* contempla que cada pregunta puede recibir como máximo una respuesta. Una vez se ha comprobado la idoneidad del uso de *c@1* en el escenario propuesto, el siguiente paso consistiría en diseñar una medida que al igual que *c@1* valore en mayor medida las preguntas que se dejan sin responder, que a las que se les responde incorrectamente cuando se emite más de una respuesta por pregunta.

  En este nuevo escenario hay que tener en cuenta que para cada pregunta hay que calcular la corrección de las respuestas obtenidas a esa pregunta, de modo que se podría tener en cuenta la posición en la cuál se encuentra la primera respuesta correcta de entre las emitidas para esa pregunta de forma similar a como se realiza con *MRR* (medida de evaluación de QA que se describió en la sección 2.3.3.1 de la página 52). Pero además, esta medida tiene que valorar a las preguntas sin responder en lugar de con el *accuracy* observado con una medida que tenga en cuenta que se puede devolver más de una respuesta por pregunta. El uso de esta medida se emplazaría en escenarios donde, al igual que con *c@1*, es preferible no devolver una respuesta a devolver una respuesta incorrecta, pero en los cuáles el usuario podría observar para cada pregunta una lista de respuestas candidatas.

Bibliografía

Javier Artiles. *Web People Search*. PhD thesis, UNED University, October 2009.

Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. The Second PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venezia, Italy*, April 2006.

J.R. Beck and E.K. Shultz. The use of relative operating characteristic (ROC) curves in test performance evaluation. In *Archives of Pathology and Laboratory Medicine 110, pp. 13-2*, 1986.

Wauter Bosma and Chris Callison-Burch. Paraphrase Substitution for Recognizing Textual Entailment. In Peters et al. (2007), pages 502–509.

Andrew P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, July 1997.

Eric J. Breck, John D. Burger, Lisa Ferro, Lynette Hirschman, David House, Marc Light, and Inderjeet Mani. How to Evaluate your Question Answering System Every Day and Still Get Real Work Done. In *Proceedings of the Second Conference on Language Resources and Evaluation (LREC-2000)*, pages 1495–1500, 2000.

Eric Brill and Jun Wu. Classifier combination for improved lexical disambiguation. In *Proceedings of the 17th international conference on Computational linguistics*, pages 191–195, Morristown, NJ, USA, 1998. Association for Computational Linguistics.

Eric Brill, Jimmy J. Lin, Michele Banko, Susan T. Dumais, and Andrew Y. Ng. Data-Intensive Question Answering. In *Proceedings of the Tenth Text REtrieval Conference (TREC)*, 2001.

Chris Buckley. trec_eval IR evaluation package. *http://trec.nist.gov/trec_eval/trec_eval.8.0.tar.gz*, 2004.

Chris Buckley and Ellen M. Voorhees. Evaluating Evaluation Measure Stability. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in Information Retrieval*, pages 33–40. ACM, 2000.

Chris Buckley and Janet A. Walz. The TREC-8 Query Track. In *TREC*, 1999.

John Burger and Lisa Ferro. Generating an Entailment Corpus from News Headlines. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 49–54, June 2005.

John D. Burger, Lisa Ferro, Warren R. Greiff, John C. Henderson, Scott Mardis, Alex Morgan, and Marc Light. MITRE's Qanda at TREC-11. In *TREC*, 2002.

Nancy Chinchor. Overview of MUC-7/MET-2. In *Proceedings of Message Understanding Conference MUC-7*, 1999.

Nancy Chinchor, Patricia Robinson, and Erica Brown. Hub-4 Named Entity Task Definition. In *Proceedings of DARPA Broadcast News Workshop*, 1998.

Jennifer Chu-Carroll, Krzysztof Czuba, John Prager, and Abraham Ittycheriah. In Question Answering, Two Heads Are Better Than One. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 24–31, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

Cyrilh Cleverdon. The Cranfield tests on index language devices. In *Aslib Proceedings*, volume 19, pages 173–192, 1967.

Gordon V. Cormack and Thomas R. Lynam. Statistical Precision of Information Retrieval Evaluation. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 533–540, New York, NY, USA, 2006. ACM.

Silviu Cucerzan and David Yarowsky. Language Independent Named Entity Recognition Combining Morphological and Contextual Evidence. In *Joint SIGDAT Conference on EMNLP and VLC*, 1999.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL Recognising Textual Entailment Challenge. In Joaquin Quiñonero Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché Buc, editors, *MLCW*, volume 3944 of *Lecture Notes in Computer Science*, pages 177–190. Springer, 2005.

Hoa Trang Dang, Jimmy Lin, and Diane Kelly. Overview of the TREC 2006 Question Answering Track. In *Fifteenth Text REtrieval Conference (TREC 2006), Gaithersburg, Maryland*, 2006.

Thomas G. Dietterich. Machine learning research: Four current directions. In *AI Magazine, 18(4)*, pages 97–136, 1997.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. The Automatic Content Extraction (ACE) Program Tasks, Data, and Evaluation. In *Proceedings of LREC 2004*, 2004.

Chris Drummond and Robert C. Holte. What ROC Curves can't do (and Cost Curves can). In *Proceedings of the 1st Workshop on ROC Analysis in Artificial Intelligence (held in conjunction with ECAI 2004)*, pages 19–26, 2004.

Óscar Ferrández. *Textual Entailment Recognition and its Applicability in NLP Tasks*. PhD thesis, Universidad de Alicante, 2009.

Óscar Ferrández, Daniel Micol, Rafael Muñoz, and Manuel Palomar. On the Application of Lexical-Syntactic Knowledge to the Answer Validation Exercise. In Peters et al. (2008), pages 377–380.

Óscar Ferrández, Rafael Muñoz, and Manuel Palomar. TE4AV: Textual Entailment for Answer Validation. In *2008 IEEE International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE'08). October 19 – 22*, 2008a.

Óscar Ferrández, Rafael Muñoz, and Manuel Palomar. Studying the Influence of Semantic Constraints in AVE. In Peters et al. (2009), pages 460–467.

Daniel Ferrés and Horacio Rodríguez. Machine Learning with Semantic-Based Distances Between Sentences for Textual Entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 60–65, Prague, June 2007. Association for Computational Linguistics.

Charles P. Friedman and Jeremy C. Wyatt. Evaluation Methods in Medical Informatics. In *Springer-Verlag, New York*, 1997.

Jun-ichi Fukumoto, Tsuneaki Kato, and Fumito Masui. An Evaluation of Question Answering Challenge (QAC-1) at the NTCIR Workshop 3. *SIGIR Forum*, 38(1): 25–28, 2004a.

Jun-ichi Fukumoto, Tsuneaki Kato, and Fumito Masui. Question Answering Challenge for Five ranked answers and List answers. Overview of NTCIR4 QAC2 Subtask 1 and 2. In *Working Notes of the Fourth NTCIR Workshop Meeting. National Institute of Informatics. June 2-4, Tokyo, Japan*, 2004b.

Fredric C. Gey, Ray R. Larson, Mark Sanderson, Hideo Joho, Paul Clough, and Vivien Petras. GeoCLEF: The CLEF 2005 Cross-Language Geographic Information Retrieval Track Overview. In Peters et al. (2006), pages 908–919.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The Third PASCAL Recognizing Textual Entailment Challenge. In *ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, 2007.

Danilo Giampiccolo, Hoa Trang Dang, Bernardo Magnini, Ido Dagan, and Bill Dolan. The Fourth PASCAL Recognizing Textual Entailment Challenge. In *Text Analysis Conference (TAC). Notebook Papers and Results. November 17-19*, 2008.

Ingo Glöckner. Combining Logic and Aggregation for Answer Selection. In Peters et al. (2008), pages 372–376.

Ingo Glöckner. Towards logic-based question answering under time constraints. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2008a.

Ingo Glöckner. Answer Validation Through Robust Logical Inference. In Peters et al. (2007), pages 518–521.

Ingo Glöckner. RAVE: A Fast Logic-Based Answer Validator. In Peters et al. (2009), pages 468–471.

Ingo Glöckner, Sven Hartrumpf, and Johannes Leveling. Logical Validation, Answer Merging and Witness Selection - A Study in Multi-Stream Question Answering. In David Evans, Sadaoki Furui, and Chantal Soulé-Dupuy, editors, *RIAO*. CID, 2007.

Ralph Grishman and Beth Sundheim. Message Understanding Conference-6: a brief history. In *Proceedings of the 16th conference on Computational linguistics - Volume 1*, 1996.

Jeroen Groenendijk. The logic of interrogation: Classical Version. In *Proceedings of the Ninth Conference on Semantics and Linguistics Theory (SALT-9)*, pages 109–126. Cornell University Press, 1999.

James A. Hanley and Barbara J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, April 1982.

Sanda Harabagiu and Andrew Hickl. Methods for Using Textual Entailment in Open-Domain Question Answering. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 905-912, Sydney*, 2006.

Sanda Harabagiu and Steven Maiorano. Finding answers in large collections of texts: Paragraph indexing+ abductive inference. In *Proceedings of the AAAI Fall Symposium on Question Answering Systems*, pages 63–71, 1999.

Sanda Harabagiu, Dan Moldovan, Christine Clark, Mitchell Bowden, John Williams, and Jeremy Bensley. Answer Mining by Combining Extraction Techniques with Abductive Reasoning. In *TREC*, pages 375–382, 2003.

Sven Hartrumpf, Ingo Glöckner, and Johannes Leveling. Efficient Question Answering with Question Decomposition and Multiple Answer Streams. In Peters et al. (2009), pages 421–428.

John Henderson and Eric Brill. Exploiting Diversity in Natural Language Processing: Combining Parsers. In *Proceedings of the Fourth Conference on Empirical Methods in Natural Language Processing*, pages 187–194, 1999.

Jesús Herrera, Anselmo Peñas, and Felisa Verdejo. Question Answering Pilot Task at CLEF 2004. In Peters et al. (2005), pages 581–590.

Jesús Herrera, Anselmo Peñas, Álvaro Rodrigo, and Felisa Verdejo. UNED at PASCAL RTE-2 Challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy*, 2006.

W. Hersh. Evaluating interactive question answering. *Advances in Open Domain Question Answering*, pages 431–455, 2006.

Andrew Hickl and Jeremy Bensley. A Discourse Commitment-based Framework for Recognizing Textual Entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 171–176, Prague, June 2007.

Andrew Hickl, John Williams, Jeremy Bensley, Kirk Roberts, Bryan Rink, and Ying Shi. Recognizing Textual Entailment with LCC GROUNDHOG System. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy*, 2006.

L. Hirschman and R. Gaizauskas. Natural language question answering: the view from here. *Natural Language Engineering*, 7(4):275–300, 2001.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Michael Junk, and Chin-Yew Lin. Question Answering in Webclopedia. In *Proceedings of the Ninth Text REtrieval Conference, pages 655-664*, 2001.

David Hull. Using Statistical Testing in the Evaluation of Retrieval Experiments. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 329–338, New York, NY, USA, 1993. ACM.

Adrian Iftene. Building a Textual Entailment System for the RTE3 Competition. Application to a QA System. In *SYNASC '08: Proceedings of the 2008 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pages 116–122, Washington, DC, USA, 2008. IEEE Computer Society.

Adrian Iftene. *Textual Entailment*. PhD thesis, Universitatea Alexandru Ioan Cuza, Iasi, 2009.

Adrian Iftene and Alexandra Balahur. Answer Validation on English and Romanian Languages. In Peters et al. (2009), pages 448–451.

Eric C. Jensen, Steven M. Beitzel, Abdur Chowdhury, and Ophir Frieder. Repeatable Evaluation of Search Services in Dynamic Environments. *ACM Trans. Inf. Syst.*, 26(1):1, 2007.

Valentin Jijkoun and Maarten de Rijke. Answer Selection in a Multi-stream Open Domain Question Answering System. In Sharon McDonald and John Tait, editors, *ECIR*, volume 2997 of *Lecture Notes in Computer Science*, pages 99–111. Springer, 2004.

Tsuneaki Kato, Jun-ichi Fukumoto, and Fumito Masui. An Overview of NTCIR-5 QAC3. In *Proceedings of the Fifth NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access*, 2005.

E. Michael Keen. Presenting results of experimental retrieval comparisons. *Inf. Process. Manage.*, 28(4):491–502, 1992.

LDC. ACE English Annotation Guidelines for Entities. Version 5.6.1 2005.05.23, 2005. URL `http://www.ldc.upenn.edu/Projects/ACE/docs/English-Entities-Guidelines_v5.6.1.pdf`.

David Lewis. Relevant Implication. *Theoria*, 54(3):162–174, 1988.

Xin Li and Dan Roth. Learning Question Classifiers. In *Proceedings 19th International conference on Computacional Linguistics*, 2002.

Anne-Laure Ligozat, Brigitte Grau, Anne Vilnat, Isabelle Robba, and Arnaud Grappy. Lexical validation of answers in Question Answering. In *WI '07: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 330–333, Washington, DC, USA, 2007. IEEE Computer Society.

Dekang Lin and Patrick Pantel. Discovery of inference rules for question-answering. *Natural Language Engineering*, 7(4):343–360, 2001.

Jimmy Lin and Dina Demner-Fushman. Will Pyramids Built of Nuggets Topple Over? In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 383–390, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

Bernardo Magnini, Matteo Negri, Roberto Prevete, and Hristo Tanev. Is It the Right Answer? Exploiting Web Redundancy for Answer Validation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July*, pages 425–432, 2002a.

Bernardo Magnini, Matteo Negri, Roberto Prevete, and Hristo Tanev. Comparing Statistical and Content-Based Techniques for Answer Validation on the Web. In *Proceedings of the VIII Convegno AI*IA*, 2002b.

Bernardo Magnini, Simone Romagnoli, Alessandro Vallin, Jesús Herrera, Anselmo Peñas, Victor Peinado, Felisa Verdejo, and Maarten de Rijke. The Multiple Language Question Answering Track at CLEF 2003. In Carol Peters, Julio Gonzalo, Martin Braschler, and Michael Kluck, editors, *CLEF*, volume 3237 of *Lecture Notes in Computer Science*, pages 471–486. Springer, 2003.

Bernardo Magnini, Alessandro Vallin, Christelle Ayache, Gregor Erbach, Anselmo Peñas, Maarten de Rijke, Paulo Rocha, Kiril Ivanov Simov, and Richard F. E. Sutcliffe. Overview of the CLEF 2004 Multilingual Question Answering Track. In Peters et al. (2005), pages 371–391.

Bernardo Magnini, Danilo Giampiccolo, Pamela Forner, Christelle Ayache, Valentin Jijkoun, Petya Osenova, Anselmo Peñas, Paulo Rocha, Bogdan Sacaleanu, and Richard F. E. Sutcliffe. Overview of the CLEF 2006 Multilingual Question Answering Track. In Peters et al. (2007), pages 223–256.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

Teruko Mitamura, Eric Nyberg, Hideki Shima, Tsuneaki Kato, Tatsunori Mori, Chin-Yew Lin, Ruihua Song, Chuan-Jie Lin, Tetsuya Sakai, Donghong Ji, and Noriko Kando. Overview of the NTCIR-7 ACLIA Tasks: Advanced Cross-Lingual Information Access. In *Proceedings of the 7th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering, and CrossLingual Information Access*, 2008.

Dan Moldovan, Sanda Harabagiu, Marius Pasca, Rada Mihalcea, Roxana Girju, Richard Goodrum, and Vasile Rus. The Structure and Performance of an Open-Domain Question Answering System. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 563–570, 2000.

Dan I. Moldovan, Sanda M. Harabagiu, Roxana Girju, Paul Morarescu, V. Finley Lacatusu, Adrian Novischi, Adriana Badulescu, and Orest Bolohan. LCC Tools for Question Answering. In *TREC*, 2002.

Véronique Moriceau, Xavier Tannier, Arnaud Grappy, and Brigitte Grau. Justification of Answers by Verification of Dependency Relations - The French AVE Task. In *Working Notes for the CLEF 2008 Workshop, 17-19 September, Aarhus, Denmark*, 2008.

Alessandro Moschitti and Fabio Massimo Zanzotto. Fast and Effective Kernels for Relational Learning from Texts. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 649–656, New York, NY, USA, 2007. ACM.

David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Journal of Linguisticae Investigationes*, 30(1), 2007.

David D. Palmer and David S. Day. A Statistical Profile of the Named Entity Task. In *Proceedings ACL Conference for Applied Natural Language Processing*, pages 190–193, 1997.

Ted Pedersen. A simple approach to building ensembles of Naive Bayesian classifiers for word sense disambiguation. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 63–69, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

Anselmo Peñas, Pamela Forner, Richard Sutcliffe, Álvaro Rodrigo, Corina Forascu, Iñaki Alegria, Danilo Giampiccolo, Nicolas Moreau, and Petya Osenova. Overview of ResPubliQA 2009: Question Answering Evaluation over European Legislation. In *CLEF 2009, LNCS, to appear*, 2010.

Carol Peters, Paul Clough, Julio Gonzalo, Gareth J. F. Jones, Michael Kluck, and Bernardo Magnini, editors. *Multilingual Information Access for Text, Speech and Images, 5th Workshop of the Cross-Language Evaluation Forum, CLEF 2004, Bath, UK, September 15-17, 2004, Revised Selected Papers*, volume 3491 of *Lecture Notes in Computer Science*, 2005. Springer.

Carol Peters, Fredric C. Gey, Julio Gonzalo, Henning Müller, Gareth J. F. Jones, Michael Kluck, Bernardo Magnini, and Maarten de Rijke, editors. *Accessing Multilingual Information Repositories, 6th Workshop of the Cross-Language Evalution Forum, CLEF 2005, Vienna, Austria, 21-23 September, 2005, Revised Selected Papers*, volume 4022 of *Lecture Notes in Computer Science*, 2006. Springer.

Carol Peters, Paul Clough, Fredric C. Gey, Jussi Karlgren, Bernardo Magnini, Douglas W. Oard, Maarten de Rijke, and Maximilian Stempfhuber, editors. *Evaluation of Multilingual and Multi-modal Information Retrieval, 7th Workshop of the Cross-Language Evaluation Forum, CLEF 2006, Alicante, Spain, September 20-22, 2006, Revised Selected Papers*, volume 4730 of *Lecture Notes in Computer Science*, 2007. Springer.

Carol Peters, Valentin Jijkoun, Thomas Mandl, Henning Müller, Douglas W. Oard, Anselmo Peñas, Vivien Petras, and Diana Santos, editors. *Advances in Multilingual and Multimodal Information Retrieval, 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007, Revised Selected Papers*, volume 5152 of *Lecture Notes in Computer Science*, 2008. Springer.

Carol Peters, Thomas Deselaers, Nicola Ferro, Julio Gonzalo, Gareth J. F. Jones, Mikko Kurimo, Thomas Mandl, Anselmo Peñas, and Vivien Petras, editors. *Evaluating Systems for Multilingual and Multimodal Information Access, 9th Works-*

hop of the Cross-Language Evaluation Forum, CLEF 2008, Aarhus, Denmark, September 17-19, 2008, Revised Selected Papers, volume 5706 of *Lecture Notes in Computer Science*, 2009. Springer.

John Prager, Eric Brown, Anni Coden, and Dragomir R. Radev. Question-Answering by Predictive Annotation. In *Proceedings of the 23rd SIGIR Conference*, pages 184–191, 2000.

Foster Provost and Tom Fawcett. Robust Classification for Imprecise Environments. *Machine Learning*, 42(3):203–231, 2001.

Álvaro Rodrigo, Anselmo Peñas, Jesús Herrera, and Felisa Verdejo. The Effect of Entity Recognition on Answer Validation. In Peters et al. (2007), pages 483–489.

Álvaro Rodrigo, Anselmo Peñas, and Felisa Verdejo. Overview of the Answer Validation Exercise 2008. In Peters et al. (2009), pages 296–313.

Tetsuya Sakai. Evaluating Evaluation Metrics based on the Bootstrap. In *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006*, pages 525–532, 2006a.

Tetsuya Sakai. Bootstrap-Based Comparisons of IR Metrics for Finding One Relevant Document. In Hwee Tou Ng, Mun-Kew Leong, Min-Yen Kan, and Donghong Ji, editors, *AIRS*, volume 4182 of *Lecture Notes in Computer Science*, pages 374–389. Springer, 2006b.

Tetsuya Sakai. On the Reliability of Factoid Question Answering Evaluation. *ACM Trans. Asian Lang. Inf. Process.*, 6(1), 2007a.

Tetsuya Sakai. On the reliability of information retrieval metrics based on graded relevance. *Inf. Process. Manage.*, 43(2):531–548, 2007b.

Tetsuya Sakai, Noriko Kando, Chuan-Jie Lin, Teruko Mitamura, Hideki Shima, Donghong Ji, Kuang-Hua Chen, and Eric Nyberg. Overview of the NTCIR-7 ACLIA IR4QA Task. In *Proceedings of the 7th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering, and Cross-Lingual Information Access*, 2008.

Mark Sanderson and Justin Zobel. Information Retrieval System Evaluation: Effort, Sensitivity, and Reliability. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 162–169, New York, NY, USA, 2005. ACM.

Erik F. Tjong Kim Sang. Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of CoNLL-2002, Taipei, Taiwan*, pages 155–158, 2002.

Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of CoNLL-2003, Edmonton, Canada*, pages 142–147, 2003.

Diana Santos, Nuno Seco, Nuno Cardoso, and Rui Vilela. HAREM: An Advanced NER Evaluation Contest for Portuguese. In Nicoletta Calzolari, Khalid Choukri, Aldo Gangemi, Bente Maegaard, Joseph Mariani, Jan Odjik, and Daniel Tapias, editors, *Proceedings of the 5th International Conference on Language Resources and Evaluation, LREC 2006*, pages 1986–1991, Genoa, Italy, 22–28 May 2006. ELRA.

S. Sarawagi. Information extraction. *Foundations and Trends in Databases*, 1(3): 261–377, 2007.

Jacques Savoy. Statistical inference in retrieval effectiveness evaluation. *Inf. Process. Manage.*, 33(4):495–512, 1997.

Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. Extended Named Entity Hierarchy. In *Proceedings of the LREC-2002*, 2002.

Mark D. Smucker, James Allan, and Ben Carterette. A Comparison of Statistical Significance Tests for Information Retrieval Evaluation. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 623–632, New York, NY, USA, 2007. ACM.

Daniel Sonntag. Distributed NLP and Machine Learning for Question Answering Grid. In *Proceedings of the workshop on Semantic Intelligent Middleware for the Web and the Grid at the 16th European Conference on Artificial Intelligence (ECAI)*, 2004.

Martin M. Soubbotin. Patterns of Potential Answer Expressions as Clues to the Right Answers. In *TREC*, 2001.

Hristo Tanev and Bernardo Magnini. Weakly Supervised Approaches for Ontology Population. In *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*, 2006.

Marta Tatu and Dan Moldovan. COGEX at RTE 3. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 22–27, Prague, June 2007.

Marta Tatu, Brandon Iles, John Slavick, Adrian Novischi, and Dan Moldovan. COGEX at the Second Recognizing Textual Entailment Challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy*, 2006.

Alberto Téllez-Valero, Manuel Montes-Y-Gómez, and Luis Villaseñor-Pineda. A Supervised Learning Approach to Spanish Answer Validation. In Peters et al. (2008), pages 391–394.

Alberto Téllez-Valero, Manuel Montes-Y-Gómez, Luis Villaseñor-Pineda, and Anselmo Peñas. Improving Question Answering by Combining Multiple Systems Via Answer Validation. In Alexander F. Gelbukh, editor, *CICLing*, volume 4919 of *Lecture Notes in Computer Science*, pages 544–554. Springer, 2008.

Alberto Téllez-Valero, Antonio Juárez-González, Manuel Montes-Y-Gómez, and Luis Villaseñor-Pineda. Using Non-Overlap Features for Supervised Answer Validation. In Peters et al. (2009).

Masatsugu Tonoike, Takehito Utsuro, and Satoshi Sato. Answer Validation by Keyword Association. In *Proc. of the 3rd Workshop on Robust Methods in Analysis of Natural Language Data*, 2004.

Alberto Téllez-Valero. *Validación de Respuestas Reconociendo la Implicación Textual*. PhD thesis, Instituto Nacional de Astrofísica, Óptica y Electrónica, 2009.

Alessandro Vallin, Bernardo Magnini, Danilo Giampiccolo, Lili Aunimo, Christelle Ayache, Petya Osenova, Anselmo Peñas, Maarten de Rijke, Bogdan Sacaleanu, Diana Santos, and Richard F. E. Sutcliffe. Overview of the CLEF 2005 Multilingual Question Answering Track. In Peters et al. (2006), pages 307–331.

C. J. van Rijsbergen. *Information Retrieval*. Butterworth, 1979.

Lucy Vanderwende and William B. Dolan. What syntax can contribute in entailment task. In *MLCW 2005, LNAI 3944, pp. 205-216. J. Quinonero-Candela et al. (eds.). Springer-Verlag*, 2005.

José L. Vicedo. La Búsqueda de Respuestas: Estado Actual y Perspectivas de Futuro. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 22:37–56, 2003.

Ellen M. Voorhees. Overview of the TREC 2001 Question Answering Track. In *E. M. Voorhees, D. K. Harman, editors: Proceedings of the Tenth Text REtrieval Conference (TREC 2001). NIST Special Publication 500-250*, 2001a.

Ellen M. Voorhees. Overview of the TREC 2002 Question Answering Track. In *E.M. Voorhees, L. P. Buckland, editors: Proceedings of the Eleventh Text REtrieval Conference (TREC 2002). NIST Publication 500-251*, 2002.

Ellen M. Voorhees. The Philosophy of Information Retrieval Evaluation. In Carol Peters, Martin Braschler, Julio Gonzalo, and Michael Kluck, editors, *CLEF*, volume 2406 of *Lecture Notes in Computer Science*, pages 355–370. Springer, 2001b.

Ellen M. Voorhees. Overview of the TREC 2003 Question Answering Track. In *Proceedings of the Twelfth Text REtrieval Conference (TREC 2003)*, 2003.

Ellen M. Voorhees. Overview of the TREC 2004 Question Answering Track. In *Proceedings of the Thirteenth Text REtreival Conference (TREC 2004)*, 2004.

Ellen M. Voorhees and Chris Buckley. The effect of Topic Set Size on Retrieval Experiment Error. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 316–323, 2002.

Ellen M. Voorhees and Dawn M. Tice. The TREC-8 Question Answering Track Evaluation. In *Text Retrieval Conference TREC-8*, pages 83–105, 1999.

Rui Wang. *Textual Entailment Recognition: A Data-Driven Approach*. PhD thesis, Universität des Saarlandes, 2007.

Rui Wang and Günter Neumann. Using Recognizing Textual Entailment as a Core Engine for Answer Validation. In Peters et al. (2008), pages 387–390.

Rui Wang and Günter Neumann. Information Synthesis for Answer Validation. In Peters et al. (2009), pages 472–475.

William Webber, Alistair Moffat, and Justin Zobel. Statistical Power in Retrieval Experimentation. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 571–580, New York, NY, USA, 2008. ACM.

Fabio Massimo Zanzotto and Alessandro Moschitti. Experimenting a "General Purpose"Textual Entailment Learner in AVE. In Peters et al. (2007), pages 510–517.

Justin Zobel. How Reliable are the Results of Large-Scale Information Retrieval Experiments? In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 307–314, New York, NY, USA, 1998. ACM.

ANEXOS

Anexo A

Resultados Answer Validation Exercise

En este anexo se muestran los resultados de todos los participantes en las tres ediciones del Answer Validation Exercise.

A.1. Resultados AVE 2006

Cuadro A.1: Precisión, cobertura y medida F sobre las respuestas correctas en inglés en el AVE 2006

Grupo	Sistema	F	Precisión	Cobertura
LCC	Tatu	0.46	0.33	0.76
U. Rome	Zanzotto_2	0.41	0.28	0.74
ITC-irst	Kouylekov	0.39	0.31	0.54
U. Rome	Zanzotto_1	0.38	0.27	0.63
U. Alicante	Kozareva_2	0.37	0.25	0.74
U. Alicante	Ferrández_2	0.32	0.20	0.72
U. Alicante	Kozareva_1	0.32	0.21	0.64
U. Alicante	Ferrández_1	0.31	0.21	0.54
U. Twente	Bosma_1	0.30	0.33	0.28
U. Twente	Bosma_2	0.28	0.27	0.28
Baseline 100 % SI		0.27	0.16	1
Baseline 50 % SI		0.24	0.16	0.5
U. P. Valencia	Bisbal	0.08	0.21	0.05

Cuadro A.2: Precisión, cobertura y medida F sobre las respuestas correctas en francés en el AVE 2006.

Grupo	Sistema	F	Precisión	Cobertura
U. Alicante	Kozareva_2	0.47	0.34	0.74
U. Alicante	Kozareva_1	0.41	0.38	0.44
Baseline 100 % SI		0.37	0.23	1
Baseline 50 % SI		0.32	0.23	0.5
LIMSI	Grau	0.11	0.43	0.06
U. Twente	Bosma	0.09	0.46	0.05

Cuadro A.3: Precisión, cobertura y medida F sobre las respuestas correctas en español en el AVE 2006.

Grupo	Sistema	F	Precisión	Cobertura
LCC	Tatu	0.61	0.53	0.71
UNED	Herrera_1	0.57	0.47	0.72
UNED	Herrera_2	0.56	0.47	0.71
UNED	Rodrigo	0.53	0.44	0.68
U. Alicante	Kozareva_2	0.53	0.41	0.76
Proyecto R2D2	R2	0.49	0.44	0.56
U. Twente	Bosma_1	0.47	0.48	0.46
Baseline 100 % SI		0.45	0.29	1
U. Twente	Bosma_2	0.43	0.55	0.36
U. Alicante	Kozareva_1	0.43	0.47	0.39
Baseline 50 % SI		0.37	0.29	0.5

Cuadro A.4: Precisión, cobertura y medida F sobre las respuestas correctas en alemán en el AVE 2006.

Grupo	Sistema	F	Precisión	Cobertura
FUH	Glöckner_1	0.54	0.58	0.51
FUH	Glöckner_2	0.50	0.73	0.38
U. Alicante	Kozareva_2	0.47	0.36	0.68
Baseline 100 % SI		0.39	0.24	1
U. Alicante	Kozareva_1	0.39	0.40	0.38
Baseline 50 % SI		0.33	0.24	0.5
U. Twente	Bosma	0.14	0.4	0.09

Cuadro A.5: Precisión, cobertura y medida F sobre las respuestas correctas en holandés en el AVE 2006.

Grupo	Sistema	F	Precisión	Cobertura
U. Twente	Bosma_1	0.39	0.29	0.59
U. Alicante	Kozareva_1	0.30	0.19	0.68
U. Alicante	Kozareva_2	0.25	0.15	0.90
U. Twente	Bosma_2	0.22	0.2	0.25
Baseline 100 % SI		0.19	0.10	1
Baseline 50 % SI		0.17	0.10	0.5

Cuadro A.6: Precisión, cobertura y medida F sobre las respuestas correctas en portugués en el AVE 2006.

Grupo	Sistema	F	Precisión	Cobertura
Baseline 100 % SI		0.38	0.24	1
U. Twente	Bosma	0.35	0.58	0.26
Baseline 50 % SI		0.32	0.24	0.5
U. Alicante	Kozareva	0.15	0.19	0.13

Cuadro A.7: Precisión, cobertura y medida F sobre las respuestas correctas en italiano en el AVE 2006.

Grupo	Sistema	F	Precisión	Cobertura
U. Alicante	Kozareva_2	0.41	0.28	0.72
U. Alicante	Kozareva_1	0.35	0.22	0.89
Baseline 100 % SI		0.29	0.17	1
Baseline 50 % SI		0.26	0.17	0.5
U. Twente	Bosma	0.17	0.33	0.11

A.2. Resultados AVE 2007

Cuadro A.8: Precisión, cobertura y medida F sobre las respuestas correctas en español en el AVE 2007

Grupo	Sistema	F	Precisión	Cobertura
INAOE	Tellez_1	0.53	0.38	0.86
INAOE	Tellez_2	0.52	0.41	0.72
UNED	Rodrigo	0.47	0.33	0.82
U. Jaén	García_1	0.37	0.24	0.85
Baseline 100 % SI		0.37	0.23	1
Baseline 50 % SI		0.32	0.23	0.5
U. Jaén	García_2	0.19	0.4	0.13

Cuadro A.9: Precisión, cobertura y medida F sobre las respuestas correctas en alemán en el AVE 2007

Grupo	Sistema	F	Precisión	Cobertura
FUH	Glöckner_1	0.72	0.61	0.9
FUH	Glöckner_2	0.68	0.54	0.94
Baseline 100 % SI		0.4	0.25	1
Baseline 50 % SI		0.34	0.25	0.5

Cuadro A.10: Precisión, cobertura y medida F sobre las respuestas correctas en inglés en el AVE 2007

Grupo	Sistema	F	Precisión	Cobertura
DFKI	Wang_2	0.55	0.44	0.71
DFKI	Wang_1	0.46	0.37	0.62
U. Alicante	Ferrández_1	0.39	0.25	0.81
Proyecto Text-Mess	T-M_1	0.36	0.25	0.62
Iasi	Iftene	0.34	0.21	0.81
UNED	Rodrigo	0.34	0.22	0.71
Proyecto Text-Mess	T-M_2	0.34	0.25	0.52
U. Alicante	Ferrández_2	0.29	0.18	0.81
Baseline 100 % SI		0.19	0.11	1
Baseline 50 % SI		0.18	0.11	0.5

Cuadro A.11: Precisión, cobertura y medida F sobre las respuestas correctas en portugués en el AVE 2007

Grupo	Sistema	F	Precisión	Cobertura
U. Évora	Saias	0.68	0.91	0.55
Baseline 100 % SI		0.6	0.43	1
Baseline 50 % SI		0.46	0.43	0.5

Cuadro A.12: Comparación de sistemas de AV con sistemas de QA en español en el AVE 2007

Grupo	Sistema	Tipo de sistema	qa_accuracy (%_mejor_combinación)
Selección perfecta		AV	0.59 (100 %)
Priberam	prib_1	QA	0.49 (83.17 %)
INAOE	Tellez_1	AV	0.45 (75.25 %)
UNED	Rodrigo	AV	0.42 (70.3 %)
U. Jaén	García_1	AV	0.41 (68.32 %)
INAOE	inao_1	QA	0.38 (63.37 %)
INAOE	Tellez_2	AV	0.36 (61.39 %)
Baseline random		AV	0.25 (41.45 %)
MIRA	mira_1	QA	0.15 (25.74 %)
UPV	upv_1	QA	0.13 (21.78 %)
U. Jaén	García_2	AV	0.08 (13.86 %)
TALP	talp_1	QA	0.07 (11.88 %)

Cuadro A.13: Comparación de sistemas de AV con sistemas de QA en alemán en el AVE 2007

Grupo	Sistema	Tipo de sistema	qa_accuracy (%_mejor_combinación)
Selección perfecta		AV	0.54 (100 %)
FUH	Glöckner_2	AV	0.50 (93.44 %)
FUH	Glöckner_1	AV	0.48 (88.52 %)
DFKI	dfki_1	QA	0.35 (65.57 %)
FUH	fuha_1	QA	0.32 (59.02 %)
Baseline random		AV	0.28 (51.91 %)
DFKI	dfki_2	QA	0.25 (45.9 %)
FUH	fuha_2	QA	0.21 (39.34 %)
DFKI	dfki_3	QA	0.05 (9.84 %)

Cuadro A.14: Comparación de sistemas de AV con sistemas de QA en inglés en el AVE 2007 en el AVE 2007

Grupo	Sistema	Tipo de sistema	qa_accuracy (%_mejor_combinación)
Selección perfecta		AV	0.3 (100 %)
DFKI	Wang_2	AV	0.21 (70 %)
Iasi	Iftene	AV	0.21 (70 %)
U. Alicante	Ferrández_2	AV	0.19 (65 %)
U. Indonesia	ui_1	QA	0.18 (60 %)
U. Alicante	Ferrández_1	AV	0.18 (60 %)
DFKI	Wang_1	AV	0.16 (55 %)
UNED	Rodrigo	AV	0.16 (55 %)
Proy. Text-Mess	T-M_1	AV	0.15 (50 %)
DFKI	dfk1_1	QA	0.13 (45 %)
Proy. Text-Mess	T-M_2	AV	0.12 (40 %)
Baseline random		AV	0.1 (35 %)
DFKI	dfki_2	QA	0.04 (15 %)
Macquarie	mqaf_1	QA	0 (0 %)
Macquarie	mqaf_2	QA	0 (0 %)

Cuadro A.15: Comparación de sistemas de AV con sistemas de QA en portugués en el AVE 2007

Grupo	Sistema	Tipo de sistema	qa_accuracy (%_mejor_combinación)
Selección perfecta		AV	0.74 (100 %)
Priberam	prib_1	QA	0.61 (82.73 %)
U. Évora	Saias	AV	0.44 (60 %)
Baseline random		AV	0.44 (60 %)
U. Évora	ue_1	QA	0.41 (55.45 %)
LCC	lcc_1	QA	0.3 (40 %)
U. Porto	up_1	QA	0.23 (30.91 %)
INESC	ines_1	QA	0.13 (17.27 %)
INESC	ines_2	QA	0.11 (15.45 %)
SINTEF	esfi_1	QA	0.07 (10 %)
SINTEF	esfi_2	QA	0.04 (5.45 %)

A.3. Resultados AVE 2008

Cuadro A.16: Precisión, cobertura y medida F sobre las respuestas correctas en alemán en el AVE 2008

Grupo	Sistema	F	Precisión	Cobertura
DFKI	Wang	0.61	0.54	0.71
FUH	Glöckner_1	0.39	0.33	0.49
FUH	Glöckner_2	0.29	0.25	0.34
Baseline 100 % SI		0.21	0,12	1
Baseline 50 % SI		0.19	0.12	0.5

Cuadro A.17: Precisión, cobertura y medida F sobre las respuestas correctas en español en el AVE 2008

Grupo	Sistema	F	Precisión	Cobertura
U. Alicante	Ferrández_2	0.44	0.32	0.67
INAOE	Tellez_2	0.39	0.30	0.59
U. Alicante	Ferrández_1	0.38	0.26	0.76
INAOE	Tellez_1	0.23	0,13	0.86
Baseline 100 % SI		0.18	0.10	1
Baseline 50 % SI		0.17	0.10	0.5
U. Jaén	García_1	0.06	0.15	0.04
U. Jaén	García_2	0.05	0.22	0.03

Cuadro A.18: Precisión, cobertura y medida F sobre las respuestas correctas en francés en el AVE 2008

Grupo	Sistema	F	Precisión	Cobertura
LIMSI	Moriceau_1	0.61	0.75	0.52
LIMSI	Moriceau_2	0.57	0.88	0.42
LINA	Jacquin	0.51	0.56	0.46
Baseline 100 % SI		0.45	0.29	1
Baseline 50 % SI		0.37	0.29	0.5
UJA	García_1	0.08	0.15	0.06
UJA	García_2	0.08	0.13	0.06

Cuadro A.19: Precisión, cobertura y medida F sobre las respuestas correctas en inglés en el AVE 2008

Grupo	Sistema	F	Precisión	Cobertura
DFKI	Wang	0.64	0.54	0.78
U. Alicante	Ferrández	0.49	0.35	0.86
UNC	Castillo_2	0.21	0.13	0.56
Iasi	Iftene_2	0.19	0.11	0.85
UNC	Castillo_1	0.17	0.09	0.94
Iasi	Iftene_1	0.17	0.09	0.76
Baseline 100 % SI		0.14	0.08	1
Baseline 50 % SI		0.13	0.08	0.5
U. Jaén	García_2	0.02	0.17	0.01
U. Jaén	García_1	0	0	0

Cuadro A.20: Precisión, cobertura y medida F sobre las respuestas correctas en rumano en el AVE 2008

Grupo	Sistema	F	Precisión	Cobertura
Iasi	Iftene_2	0.23	0.13	0.92
Iasi	Iftene_1	0.22	0.12	0.92
Baseline 100 % SI		0.20	0.11	1
Baseline 50 % SI		0.19	0.11	0.50

Cuadro A.21: Comparación de sistemas de AV con sistemas de QA en alemán en el AVE 2008: (1) estimated_qa_performance, (2) qa_accuracy (%_mejor_combinación), (3) qa_rej_accuracy, (4) qa_accuracy_max

Grupo	**Sistema**	**Tipo de sistema**	**(1)**	**(2)**	**(3)**	**(4)**
Selección perfecta		AV	0.77	0.52 (100 %)	0.48	1
DFKI	Wang	AV	0.52	0.43 (82.26 %)	0.21	0.64
DFKI	dfki_1	QA	0.38	0.38 (72.58 %)	0	0.38
DFKI	dfki_2	QA	0.37	0.37 (70.97 %)	0	0.37
FUH	Glöckner_1	AV	0.32	0.32 (61.29 %)	0	0.32
FUH	fuh_1	QA	0.24	0.24 (45.16 %)	0	0.24
FUH	Glöckner_2	AV	0.23	0.23 (43.55 %)	0	0.23
FUH	fuh_2	QA	0.22	0.22 (41.94 %)	0	0.22
UH-UK	log_1	QA	0.17	0.17 (32.26 %)	0	0.17
FUH	fuh_3	QA	0.16	0.16 (30.65 %)	0	0.16
FUH	fuh_4	QA	0.16	0.16 (30.65 %)	0	0.16
UH-UK	log_2	QA	0.15	0.15 (29.03 %)	0	0.15
DFKI	dfki_2	QA	0.14	0.14 (27.42 %)	0	0.14
FUH	fuh_5	QA	0.12	0.12 (22.58 %)	0	0.12
Baseline random		AV	0.11	0.11 (21.13 %)	0	0.11
FUH	fuh_6	QA	0.10	0.10 (19.35 %)	0	0.10

Cuadro A.22: Comparación de sistemas de AV con sistemas de QA en español en el AVE 2008: (1) estimated_qa_performance, (2) qa_accuracy (%_mejor_combinación), (3) qa_rej_accuracy, (4) qa_accuracy_max

Grupo	**Sistema**	**Tipo de sistema**	**(1)**	**(2)**	**(3)**	**(4)**
Selección perfecta		AV	0.85	0.62 (100 %)	0.38	1
Priberam	prib_1	QA	0.54	0.54 (88.10 %)	0	0.54
U. Alicante	Ferrández_1	AV	0.37	0.32 (52.38 %)	0.14	0.46
INAOE	Tellez_1	AV	0.34	0.32 (52.38 %)	0.06	0.38
U. Alicante	Ferrández_2	AV	0.33	0.27 (44.05 %)	0.21	0.48
INAOE	Tellez_2	AV	0.33	0.27 (44.05 %)	0.22	0.49
INAOE	inao_1	QA	0.25	0.25 (40.48 %)	0	0.25
INAOE	inao_2	QA	0.25	0.25 (40.48 %)	0	0.25
U. Alicante	ua_1	QA	0.22	0.22 (35.71 %)	0	0.22
MIRACLE	mira_1	QA	0.21	0.21 (33.33 %)	0	0.21
MIRACLE	mira_2	QA	0.18	0.18 (29.76 %)	0	0.18
U. Alicante	ua_2	QA	0.18	0.18 (28.57 %)	0	0.18
U. Alicante	ua_3	QA	0.13	0.13 (21.43 %)	0	0.13
U. Alicante	ua_4	QA	0.12	0.12 (19.05 %)	0	0.12
Baseline random		AV	0.11	0.11 (17.12 %)	0	0.11
MIRACLE	mira_3	QA	0.06	0.06 (9.52 %)	0	0.06
U. Jaén	García_1	AV	0.06	0.04 (7.14 %)	0.32	0.36
U. Jaén	García_1	AV	0.03	0.02 (3.57 %)	0.35	0.37

Cuadro A.23: Comparación de sistemas de AV con sistemas de QA en francés en el AVE 2008: (1) estimated_qa_performance, (2) qa_accuracy (%_mejor_combinación), (3) qa_rej_accuracy, (4) qa_accuracy_max

Grupo	**Sistema**	**Tipo de sistema**	**(1)**	**(2)**	**(3)**	**(4)**
Selección perfecta		AV	0.73	0.48 (100 %)	0.52	1
SYNA	syna_1	QA	0.47	0.47 (98.08 %)	0	0.47
Baseline random		AV	0.33	0.33 (68.80 %)	0	0.33
LIMSI	Moriceau_1	AV	0.32	0.23 (48.08 %)	0.39	0.62
LINA	Jacquin	AV	0.29	0.21 (44.23 %)	0.35	0.56
LIMSI	Moriceau_2	AV	0.29	0.19 (40.38 %)	0.48	0.67
SYNA	syna_2	QA	0.19	0.19 (40.38 %)	0	0.19
SYNA	syna_3	QA	0.17	0.17 (34.62 %)	0	0.17
U. Jaén	García_1	AV	0.04	0.03 (5.77 %)	0.41	0.44
U. Jaén	García_2	AV	0.04	0.03 (5.77 %)	0.41	0.44

Cuadro A.24: Comparación de sistemas de AV con sistemas de QA en inglés en el AVE 2008: (1) estimated_qa_performance, (2) qa_accuracy (%_mejor_combinación), (3) qa_rej_accuracy, (4) qa_accuracy_max

Grupo	**Sistema**	**Tipo de sistema**	**(1)**	**(2)**	**(3)**	**(4)**
Selección perfecta		AV	0.56	0.34 (100 %)	0.66	1
DFKI	Wang	AV	0.34	0.24 (70.37 %)	0.44	0.68
U. Alicante	Ferrández	AV	0.27	0.19 (57.41 %)	0.4	0.59
Iasi	Iftene_2	AV	0.24	0.24 (70.37 %)	0.01	0.25
WLVS	wlvs_1	QA	0.21	0.21 (62.96 %)	0	0.21
Iasi	Iftene_1	AV	0.19	0.19 (57.41 %)	0	0.19
UNC	Castillo_2	AV	0.17	0.16 (46.30 %)	0.1	0.26
DFKI	dfki_1	QA	0.17	0.17 (50 %)	0	0.17
UNC	Castillo_1	AV	0.16	0.16 (46.30 %)	0	0.16
DCU	dcu_1	QA	0.10	0.10 (29.63 %)	0	0.10
Baseline random		AV	0.09	0.09 (25.25 %)	0	0.09
NLE	nle_1	QA	0.06	0.06 (18.52 %)	0	0.06
NLE	nle_2	QA	0.05	0.05 (14.81 %)	0	0.05
ILK	ilk_1	QA	0.04	0.04 (12.96 %)	0	0.04
U. Jaén	García_2	AV	0.01	0.01 (1.85 %)	0.64	0.65
DCU	dcu_2	QA	0.01	0.01 (1.85 %)	0	0.01
U. Jaén	García_1	AV	0	0 (0 %)	0.63	0.63

Cuadro A.25: Comparación de sistemas de AV con sistemas de QA en rumano en el AVE 2008: (1) estimated_qa_performance, (2) qa_accuracy (%_mejor_combinación), (3) qa_rej_accuracy, (4) qa_accuracy_max

Grupo	**Sistema**	**Tipo de sistema**	**(1)**	**(2)**	**(3)**	**(4)**
Selección perfecta		AV	0.65	0.41 (100 %)	0.59	1
Iasi	Iftene_2	AV	0.25	0.24 (57.14 %)	0.05	0.29
Iasi	iasi_1	QA	0.22	0.22 (53.06 %)	0	0.22
Iasi	iasi_2	QA	0.19	0.19 (46.94 %)	0	0.19
Iasi	Iftene_1	AV	0.17	0.17 (40.82 %)	0	0.17
ICIA	icia_1	QA	0.17	0.17 (40.82 %)	0	0.17
Baseline random		AV	0.10	0.10 (24.66 %)	0	0.10
ICIA	icia_2	QA	0.08	0.08 (18.37 %)	0	0.08

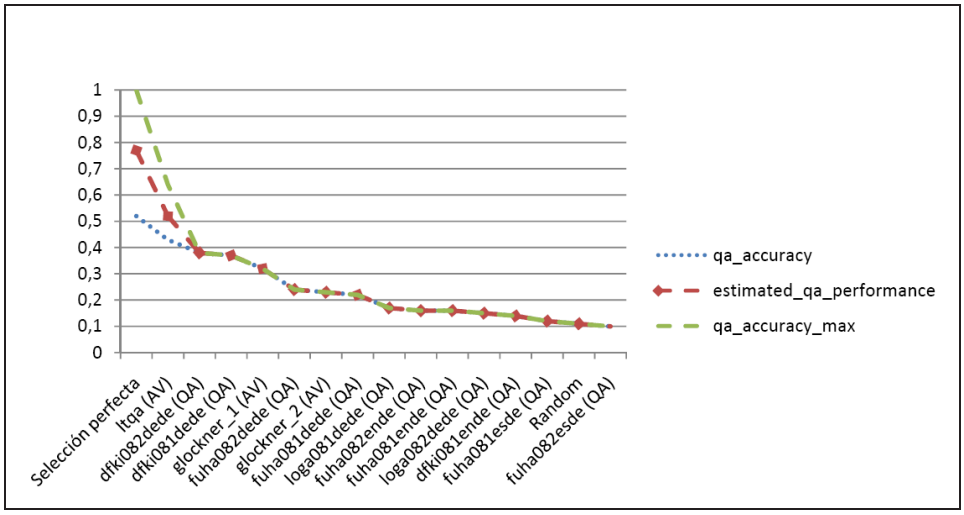

Figura A.1: Gráfico comparando el rendimiento de los sistemas del QA@CLEF y los del AVE 2008 en alemán

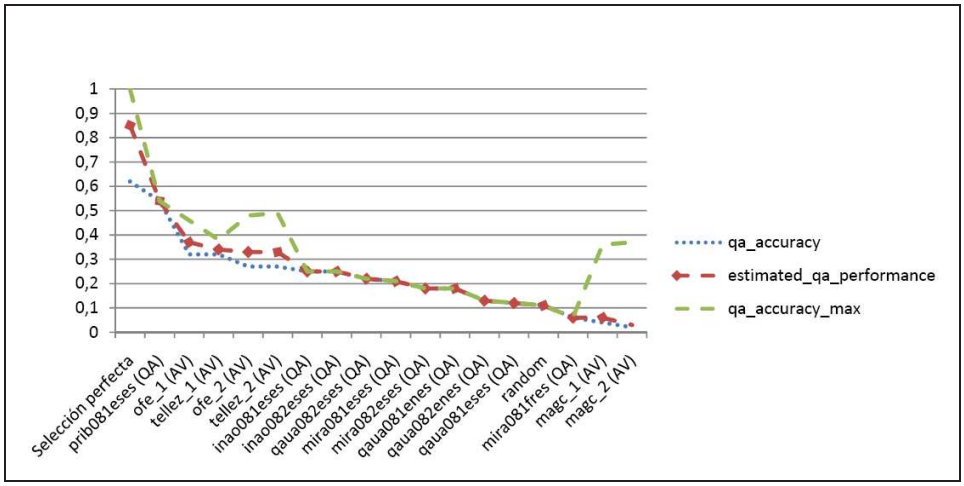

Figura A.2: Gráfico comparando el rendimiento de los sistemas del QA@CLEF y los del AVE 2008 en español

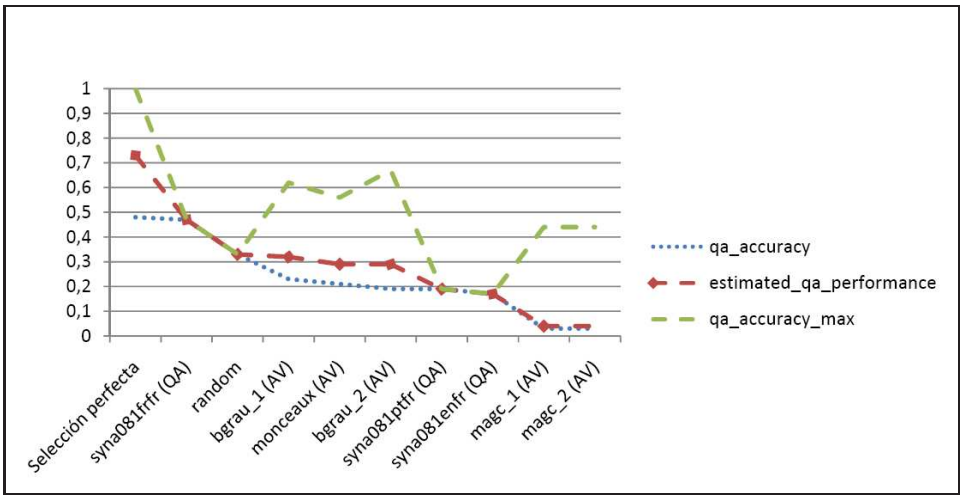

Figura A.3: Gráfico comparando el rendimiento de los sistemas del QA@CLEF y los del AVE 2008 en alemán

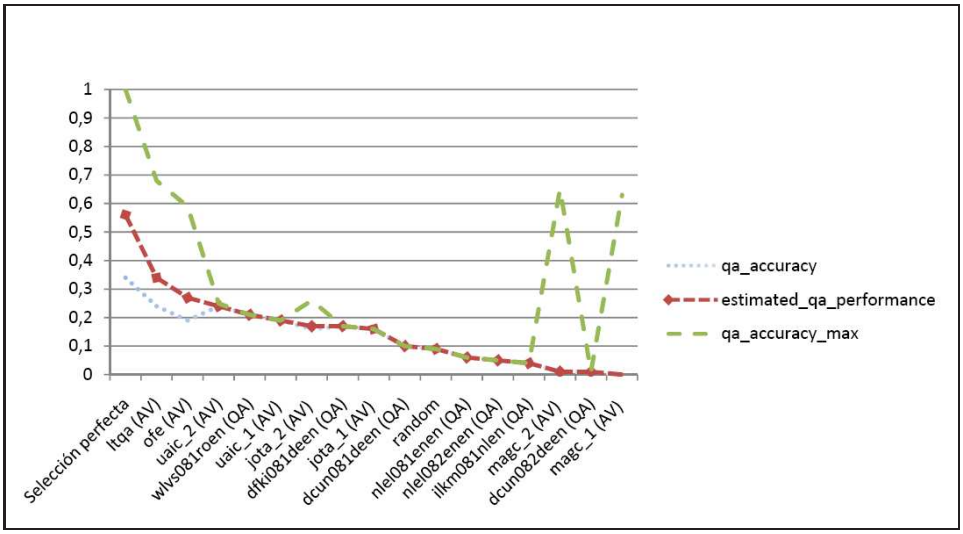

Figura A.4: Gráfico comparando el rendimiento de los sistemas del QA@CLEF y los del AVE 2008 en inglés

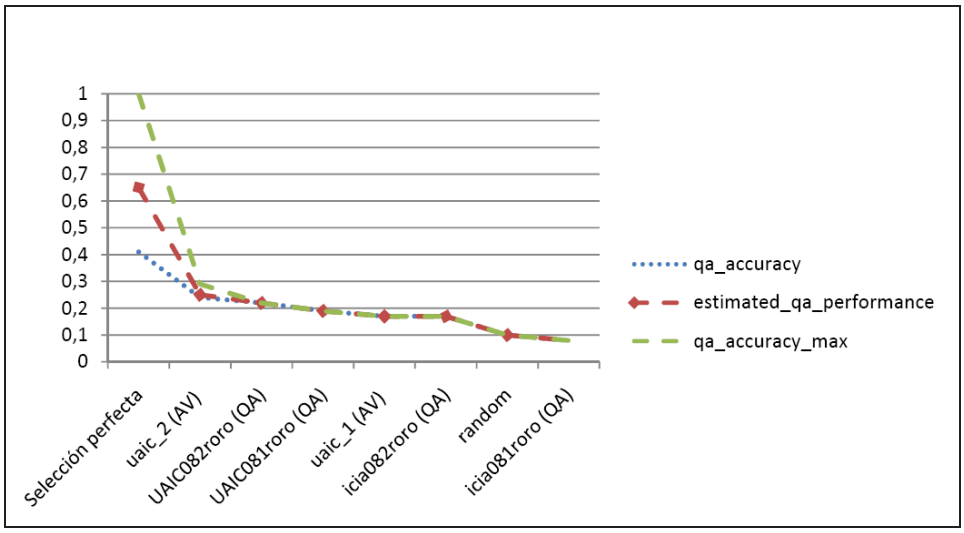

Figura A.5: Gráfico comparando el rendimiento de los sistemas del QA@CLEF y los del AVE 2008 en rumano

Anexo B

Impacto de la Tesis en la Comunidad Científica

En este anexo se muestra el impacto que ha tenido el trabajo desarrollado en esta tesis dentro de la comunidad científica. A la hora de describir este impacto se han seguido dos puntos de vista:

- La contribución que ha tenido esta tesis a los trabajos desarrollados por otros autores. El resultado de estas contribuciones se puede ver en las referencias realizadas al trabajo de esta tesis.
- Los artículos escritos por el autor en relación con esta tesis.

En los siguientes subapartados se recoge cada uno de estos puntos de vista.

B.1. Contribuciones a la Comunidad

El trabajo mostrado en esta tesis ha contribuido al desarrollo de las investigaciones realizadas por otros autores en varias líneas. A continuación se muestra cada una de estas líneas:

- Desarrollo y evaluación de sistemas de AV haciendo uso de las colecciones generadas en el AVE y que se mostraron en el Capítulo 5 (página 125) (Ferrández et al., 2008a; Ligozat et al., 2007).
- Desarrollo y evaluación de sistemas de RTE haciendo uso de las colecciones generadas en el AVE 2006 (Ferrés and Rodríguez, 2007; Moschitti and Zanzotto, 2007).
- Incorporación de módulos de AV en sistemas de QA para mejorar los resultados de estos últimos (Glöckner, 2008a; Téllez-Valero et al., 2009).

- Desarrollo de sistemas de QA multi-flujo que realizan la selección por medio de un módulo de AV (Glöckner et al., 2007; Hartrumpf et al., 2008; Téllez-Valero et al., 2008).

- Desarrollo de sistemas de AV que siguen el modelo basado en RTE propuesto en el Capítulo 3 (página 89) (Ferrández et al., 2008a; Iftene, 2008).

Por otro lado, ha habido también otras tesis doctorales donde se ha desarrollado trabajo relacionado con el de ésta. Algunos ejemplos se muestran a continuación con un pequeño comentario para cada una de ellas:

- En la tesis de Adrian Iftene (Iftene, 2009) se describe un sistema de RTE que logró obtener uno de los mejores resultados en los RTE-3 y 4. El sistema está basado en el reconocimiento de entidades nombradas y la implicación entre ellas y fue puesto a prueba como parte de un módulo de AV que participó en las ediciones de 2007 y 2008 del AVE. Además, el autor logró mejorar los resultados de un sistema de QA incorporando el módulo de AV que utilizaron en el AVE.

- La tesis de Óscar Ferrández (Ferrández, 2009) propone también el desarrollo de un sistema de RTE. Sin embargo, este sistema se desarrolla bajo la suposición de que esta tarea se debe de abordar desde distintos niveles lingüísticos. En concreto, el trabajo desarrollado se centra en los niveles léxico, sintáctico y semántico para detectar diversas relaciones de implicación entre el texto y la hipótesis. El funcionamiento del sistema propuesto se puso a prueba tanto en los RTE Challenges, como en las evaluaciones del AVE, donde consiguió estar entre los mejores sistemas.

- Otra tesis relacionada con este trabajo es la desarrollada por Alberto Téllez-Valero (Téllez-Valero, 2009). En esta tesis se aborda el desarrollo de un sistema de AV basado en RTE siguiendo la propuesta realizada en el Capítulo 3 (página 89). El sistema fue evaluado en el AVE 2007 y 2008, obteniendo uno de los mejores resultados en castellano, lo cuál motivó al autor para utilizar su sistema como parte de un sistema de QA, logrando mejorar los resultados de este último.

- La tesis llevada a cabo por Rui Wang (Wang, 2007) supone también otro aporte a la comunidad de RTE. Su tesis presenta un sistema que lleva a cabo un procesamiento basado en análisis de dependencias. En caso de que este procesamiento falle, la decisión de implicación se toma en función de dos procesamientos adicionales: uno basado en relaciones de dependencia locales y otro basado en bolsa de palabras. Además de ser probado en los RTE Challenges, el sistema propuesto participó en el AVE 2007 para comprobar la validez de su enfoque en AV, obteniendo los mejores resultados en inglés.

B.2. Publicaciones del Autor

A continuación se muestra la relación de artículos publicados por parte del autor y que están relacionados con el trabajo desarrollado en esta tesis:

- Anselmo Peñas, **Álvaro Rodrigo** and Felisa Verdejo. *SPARTE, a Test Suite for Recognising Textual Entailment in Spanish*. In Alexander F. Gelbukh, editor, CICLing, volume 3878 of Lecture Notes in Computer Science, pages 275–286. Springer, 2006. 7 referencias externas en `http://scholar.google.com/` a fecha 4 de febrero de 2010.

  En este artículo se describe el enfoque seguido para desarrollar la colección mostrada en la sección 3.2 (página 93), la cuál sirvió para probar el modelo de AV basado en RTE propuesto en el Capítulo 3 (página 89). En este artículo se describe el proceso que se siguió para construir esta colección y la evaluación que se realizó de la misma.

- Jesús Herrera, Anselmo Peñas, **Álvaro Rodrigo** and Felisa Verdejo. *UNED at PASCAL RTE-2 Challenge*. In Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy., 2006. 5 referencias externas en `http://scholar.google.com/` a fecha 4 de febrero de 2010.

  En este artículo se describe un sistema de RTE desarrollado, entre otros, por el autor de esta tesis y que fue evaluado en el RTE-2 Challenge. La principal contribución de este trabajo fue la inclusión del reconocimiento de implicación entre rangos numéricos y los primeros pasos para la detección de implicación entre entidades nombradas. En ambos casos, la información obtenida mostró ser útil para la mejora de los resultados del sistema propuesto. El sistema propuesto en este artículo constituye un primer paso para poder desarrollar un sistema de AV siguiendo el modelo propuesto en el Capítulo 3 (página 89).

- Anselmo Peñas, **Álvaro Rodrigo**, Valentín Sama and Felisa Verdejo. *Overview of the Answer Validation Exercise 2006*. In Carol Peters, Paul Clough, Fredric C. Gey, Jussi Karlgren, Bernardo Magnini, Douglas W. Oard, Maarten de Rijke and Maximilian Stempfhuber, editors, CLEF, volume 4730 of Lecture Notes in Computer Science, pages 257–264. Springer, 2006. 22 referencias externas en `http://scholar.google.com/` a fecha 4 de febrero de 2010.

  En este artículo se describe la primera edición del AVE (celebrada dentro del CLEF 2006), en la cuál se puso en práctica por primera vez la metodología de evaluación propuesta en el Capítulo 5 (página 125). En el artículo se muestra la metodología que se propuso a partir de la definición de la tarea de RTE, el proceso para generar las colecciones de evaluación y las medidas propuestas para la comparación de los sistemas participantes. Además, en el artículo se

muestran los resultados obtenidos por los participantes y se comparan los métodos empleados por cada uno de ellos.

- **Álvaro Rodrigo**, Anselmo Peñas, Jesús Herrera and Felisa Verdejo. *The Effect of Entity Recognition on Answer Validation*. In Carol Peters, Paul Clough, Fredric C. Gey, Jussi Karlgren, Bernardo Magnini, Douglas W. Oard, Maarten de Rijke and Maximilian Stempfhuber, editors, CLEF, volume 4730 of Lecture Notes in Computer Science, pages 483–489. Springer, 2006. 2 referencias externas en `http://scholar.google.com/` a fecha 4 de febrero de 2010.

  En este artículo se describe el sistema de AV propuesto por los autores para la evaluación realizada en el AVE 2006, el cuál sigue el modelo propuesto en el Capítulo 3 (página 89). En concreto, se pretendía probar la validez de modelo desde el punto de vista de un participante. El sistema se desarrolló en castellano y presentaba como principal innovación la incorporación del reconocimiento de implicación textual entre entidades en los pares texto-hipótesis. Los resultados obtenidos fueron prometedores, lo cuál sugirió que el reconocimiento de entidades nombradas suponía una información relevante para la tarea de AV. Esto sirvió de motivación para que otros grupos incorporasen este tipo de procesamiento a sus sistemas en las siguientes ediciones del AVE y para comprobar la validez del modelo propuesto.

- **Álvaro Rodrigo**, Anselmo Peñas, Jesús Herrera and Felisa Verdejo. *Experiments of UNED at the Third Recognising Textual Entailment Challenge*. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pages 89–94, Prague, June 2007. Association for Computational Linguistics. 5 referencias externas en `http://scholar.google.com/` a fecha 4 de febrero de 2010.

  En este artículo se describe el sistema de RTE desarrollado por los autores a partir del sistema presentado en el RTE-2 y presentado en el RTE-3 Challenge, y que se pretendía utilizar en AV. La principal novedad que ofrecía este sistema radica en que tras los resultados obtenidos por los autores durante su participación en el AVE 2006 (Rodrigo et al., 2007) y el RTE-2 (Herrera et al., 2006), se propuso el reconocimiento de implicación textual entre entidades cuando los pares de evaluación se obtenían de otras tareas distintas a las de QA (lo cuál fue probada en el AVE 2006). Los resultados mostraron que la implicación entre entidades lograba aportar mejores resultados en la tarea de QA, pero en las demás era necesario la utilización de información adicional. Por otro lado, en este artículo se probó a utilizar distinta información en función de la tarea a partir de la cuál se generaban los pares de evaluación. Mediante esta propuesta se mostró que se podían mejorar los resultados del sistema, lo cuál sugiere que se podrían mejorar los resultados en RTE si los pares de cada tarea que se pretende abordar (IE, IR, QA o SUM en los RTE Challenges) son tratados de manera diferente.

- Anselmo Peñas, **Álvaro Rodrigo**, Valentín Sama and Felisa Verdejo. *Testing the Reasoning for Question Answering Validation*. In Journal of Logic and Computation. 18(3), pages 459–474, 2008. 4 referencias externas en `http://scholar.google.com/` a fecha 4 de febrero de 2010.

  En este artículo se propone el modelo de sistemas de AV basados en RTE que se describió en el Capítulo 3 (página 89). Una vez descritas las principales características de este modelo, en este artículo se propone la evaluación de este tipo de sistemas describiendo las medidas a utilizar (las propuestas para evaluar la validación en el Capítulo 4 de la página 105) y un método para desarrollar colecciones de evaluación. Como caso de estudio de la metodología de evaluación propuesta se utilizó la evaluación realizada en el AVE 2006.

- Anselmo Peñas, **Álvaro Rodrigo** and Felisa Verdejo. *Overview of the Answer Validation Exercise 2007*. In Carol Peters, Valentin Jijkoun, Thomas Mandl, Henning Müller, Douglas W. Oard, Anselmo Peñas, Vivien Petras, and Diana Santos, editors, CLEF, volume 5152 of Lecture Notes in Computer Science, pages 237–248. Springer, 2007. 15 referencias externas en `http://scholar.google.com/` a fecha 4 de febrero de 2010.

  En este artículo se describe la tarea desarrollada en el AVE 2007, en la cuál se presenta la metodología del Capítulo 5 (pagina 125) con las mejoras introducidas despues de la experiencia obtenida en el AVE 2006. Estas modificaciones hacen referencia, principalmente, a la omisión de hipotesis ya construidas en las colecciones de evaluación y a la incorporación de medidas para evaluar la selección de respuestas (en esta edición solamente se incluyeron las medidas enfocadas a evaluar la correcta selección, las cuáles fueron descritas en la sección 4.2.1 de la pagina 109). Además, en el artículo se muestran los resultados de los sistemas participantes, y tras la inclusión de las nuevas medidas de evaluación, se realiza una comparación de los sistemas participantes con los de QA que participaron en el CLEF. Los resultados de esta comparación permiten observar que se podrían mejorar los resultados de los sistemas de QA mediante la incorporación de módulos de AV.

- **Álvaro Rodrigo**, Anselmo Peñas and Felisa Verdejo. *UNED at Answer Validation Exercise 2007*. In Carol Peters, Valentin Jijkoun, Thomas Mandl, Henning Müller, Douglas W. Oard, Anselmo Peñas, Vivien Petras and Diana Santos, editors, CLEF, volume 5152 of Lecture Notes in Computer Science, pages 237–248. Springer, 2007. 7 referencias externas a fecha 4 de febrero de 2010 en `http://scholar.google.com/`.

  En este artículo se describe el sistema de AV presentado por los autores al AVE 2007. El sistema propuesto deriva del presentado en el AVE 2006 (Rodrigo et al., 2007). El artículo muestra un enfoque para aplicar dicho sistema a las situaciones en las cuáles en vez de pares texto-hipotesis para realizar la validación se dispone de tripletas {pregunta, respuesta, texto soporte} (tal y

como se planteó la tarea en el AVE 2007). Los resultados obtenidos sugieren que la solución adoptada es adecuada y que en este contexto las entidades nombradas siguen siendo importantes en la tarea de AV.

- **Álvaro Rodrigo**, Anselmo Peñas and Felisa Verdejo. *Towards an Entity-based recognition of Textual Entailment*. In Text Analysis Conference (TAC) 2008 Workshop. Maryland, USA, 2008. 2 referencias externas en `http://scholar.google.com/` a fecha 4 de febrero de 2010.

  En este artículo se describe el sistema propuesto por los autores para el RTE-4 y los resultados obtenidos por el mismo. El sistema hace uso de algunos de los módulos empleados por los autores en los sistemas que utilizaron en ediciones anteriores de los RTE Challenges, pero cambiando el enfoque con el cual se afronta la tarea. En concreto, en este artículo se presenta un enfoque que presta mas atención a la estructura y las relaciones dentro del texto y de la hipotesis con el objetivo de realizar un procesamiento más profundo y obtener mejores resultados tanto en esta tarea como en AV.

- **Álvaro Rodrigo**, Anselmo Peñas and Felisa Verdejo. *Evaluating Answer Validation in multi-stream Question Answering*. In The Second International Workshop on Evaluating Information Access (EVIA2008), A Satellite Workshop of NTCIR-7, 2008.

  En este artículo se presenta la parte de la metodología expuesta en el Capítulo 5 (pagina 125) correspondiente a la evaluación de sistemas de AV que realizan la selección en un sistema multi-flujo de QA. Con este propósito, en el artículo se presentan una serie de medidas enfocadas a esta tarea y se propone por primera vez el uso de una medida que tiene en cuenta la capacidad de los sistemas de AV para detectar preguntas para las cuáles no se ha dado ninguna respuesta correcta. Para realizar los experimentos de este artículo se hizo uso de los datos generados por los participantes del AVE 2007.

- **Álvaro Rodrigo**, Anselmo Peñas and Felisa Verdejo. *Overview of the Answer Validation Exercise 2008*. In Carol Peters, Thomas Deselaers, Nicola Ferro, Julio Gonzalo, Gareth J. F. Jones, Mikko Kurimo, Thomas Mandl, Anselmo Peñas, and Vivien Petras, editors, CLEF, volume 5706 of Lecture Notes in Computer Science, pages 296–313. Springer, 2008. 11 referencias externas en `http://scholar.google.com/` a fecha 4 de febrero de 2010.

  En este artículo se describe la ultima edición del AVE, mostrándose la versión final de la metodología del Capítulo 5 (pagina 125). La principal novedad de esta edición consiste en la inclusión de todo el conjunto de medidas propuestas en el Capítulo 4 (pagina 105). Además, los resultados de los distintos sistemas participantes permitieron conocer las cualidades de cada una de estas medidas, mostrando el escenario para el cuál es más adecuado utilizar cada una de ellas. Finalmente, los resultados obtenidos aportaron nuevas

evidencias de la mejora que se podría obtener en QA mediante la inclusión de módulos de AV, incluyendo como novedad en esta edición la detección de preguntas sin respuestas correctas.

- **Álvaro Rodrigo**, Anselmo Peñas and Felisa Verdejo. *Comparación de Enfoques para Evaluar la Validación de Respuestas*. Procesamiento del Lenguaje Natural, 43:277–285, 2009.

  En este artículo se describe la comparación realizada en el Capítulo 4 (pagina 105) entre el uso de la *medida-F* y una medida basada en el análisis del espacio ROC (medida denominada AUC) para evaluar la validación de respuestas. Ambas medidas se comparan en cuanto su estabilidad, su poder de discriminación y su adecuación para la evaluación en AV. Como resultado de la comparación se concluye que la *medida-F* es más adecuada cuando se pretende evaluar un sistema de AV cuyo objetivo es mejorar los resultados en QA.

- Anselmo Peñas, Pamela Forner, Richard Sutcliffe, **Álvaro Rodrigo**, Corina Forascu, Iñaki Alegria, Danilo Giampiccolo, Nicolas Moreau and Petya Osenova. *Overview of ResPubliQA 2009: Question Answering Evaluation over European Legislation*. In CLEF 2009, LNCS, to appear.

  El artículo describe la tarea desarrollada en el ResPubliQA 2009, la cuál proponía un escenario similar al propuesto en el Capítulo 6 (pagina 161) y hacia uso de la medida propuesta en ese capítulo. En el artículo se muestra el proceso seguido para desarrollar las colecciones de evaluación utilizadas en la tarea, así como la evaluación realizada y los resultados obtenidos por los sistemas participantes. Además, en el artículo se compara el uso de *accuracy* y de *c@1*, observándose la idoneidad del uso de *c@1* para detectar los enfoques más prometedores en el escenario que se plantea.

- **Álvaro Rodrigo**, Joaquín Pérez-Iglesias, Anselmo Peñas, Guillermo Garrido and Lourdes Araujo. *Approaching Question Answering by means of Paragraph Validation*. In CLEF 2009, LNCS, to appear.

  En este artículo se describe el sistema propuesto por los autores para la evaluación realizada dentro del ResPubliQA 2009. El sistema propuesto está formado principalmente por una etapa de IR enfocada a mejorar los resultados de un sistema de QA y por un módulo de AV. Dado que la evaluación realizada en el ResPubliQA 2009 utilizaba la medida de evaluación propuesta en el Capítulo 6 (pagina 161), en esta tarea se permitía dada una pregunta decidir no responder a la misma en caso de que el sistema no estuviese seguro de poder encontrar una respuesta correcta. Es por ello que el sistema de AV aprovechaba la experiencia de los autores como organizadores del AVE y desarrolladores de sistemas de AV para tomar la decisión sobre si alguna de las respuestas candidatas era o no correcta. En caso de que hubiese

mas de una respuesta correcta, el sistema decidía qué respuesta devolver haciendo uso de medidas de solapamiento considerando n-gramas. El sistema propuesto consiguió los mejores resultados en ingles y los segundos mejores en castellano.

- Joaquín Pérez-Iglesias, Guillermo Garrido, **Álvaro Rodrigo**, Lourdes Araujo and Anselmo Peñas. *Information Retrieval Baselines for the ResPubliQA Task*. In CLEF 2009, LNCS, to appear.

  En este artículo los autores proponen un sistema de IR desarrollado para participar en el ResPubliQA 2009. El sistema está orientado a mejorar los resultados de un sistema de QA y fue presentado en todos los idiomas disponibles como sistema baseline. El sistema consiguió resultados muy similares a los de los mejores sistemas de cada idioma, por lo que muestra la utilidad del enfoque propuesto para mejorar los resultados en QA.

- **Álvaro Rodrigo**, Anselmo Peñas and Felisa Verdejo. *Answer Validation Exercises at CLEF*. In LREC 2010. To appear.

  En este artículo se realiza un resumen de las principales características de la evaluación desarrollada en las tres ediciones del AVE. En concreto, el artículo muestra la motivación para proponer la tarea, los objetivos que se plantearon al inicio de la misma, las características de los recursos desarrollados, así como las medidas de evaluación propuestas. Finalmente, en el artículo se discuten también los resultados de los participantes y las principales conclusiones que se obtuvieron.

- Pamela Forner, Danilo Giampiccolo, Bernardo Magnini, Anselmo Peñas, **Álvaro Rodrigo** and Richard Sutcliffe. *Evaluating Multilingual Question Answering Systems at CLEF*. In LREC 2010. To appear.

  Este artículo resume las evaluaciones de sistemas de QA que se han celebrado en el marco del CLEF desde su primera edición en 2003. El artículo describe el proceso de organización de la tarea (la cuál ha estado enfocada a la evaluación tanto de sistemas monolingües como multilingües) en términos de creación de recursos, idiomas en los cuáles se propuso, evaluación llevada a cabo (la cuál incluyó en su última edición el uso de *c@1*) y resultados de los participantes.