Índice

Introducción

- Búsqueda de Respuestas (BR); a.k.a. *Question Answering (QA)*
- **Objetivo:** devolver respuestas concretas a **preguntas precisas y arbitrarias** de los usuarios en base al contenido de una colección de documentos

"¿Dónde está el Museo del Louvre?"	En París, Francia
"¿Cuál es la moneda de China?"	El yuan
"¿Quién es el presidente de Francia?"	Nicolas Sarkozy
"¿Cuántos céntimos hay en un euro?"	100

- **Problemas:**
  - **Variación lingüística**, i.e. las diferencias entre cómo está formulada la pregunta y cómo aparece la respuesta en el texto

    `Pregunta:` *"¿Quién es el presidente francés?"*
    `Texto:` *"(...) El jefe de estado de la República Francesa, Nicolas Sarkozy (...)"*
  - Detectar **cuándo NO hay respuesta** (ej. si no aparece en los docs.)
- Combina técnicas de IR e IE:
  - IR: localiza documentos relacionados con el tema de la consulta, pero no extrae la información requerida
  - IE: extrae la información requerida, pero no permite procesar consultas arbitrarias (sistemas muy especializados dependientes del dominio)

Introducción (cont.)

- **Aplicaciones:** cq. ámbito en el que el usuario final necesita conocer un dato específico pero no puede/quiere leer toda la doc. referente a dicho tema de búsqueda. Ej.:
  - Sistemas de ayuda en línea
  - Sistemas. de consulta de procedimientos en grandes organizaciones
  - Interfaces de consulta de manuales técnicos
  - Sistemas de consulta de repositorios de documentos
- Ejemplos: sistemas QA on-line:
  - START (generalista): `http://start.csail.mit.edu/`
  - EAGLi (genómica): `http://eagl.unige.ch/EAGLi/`

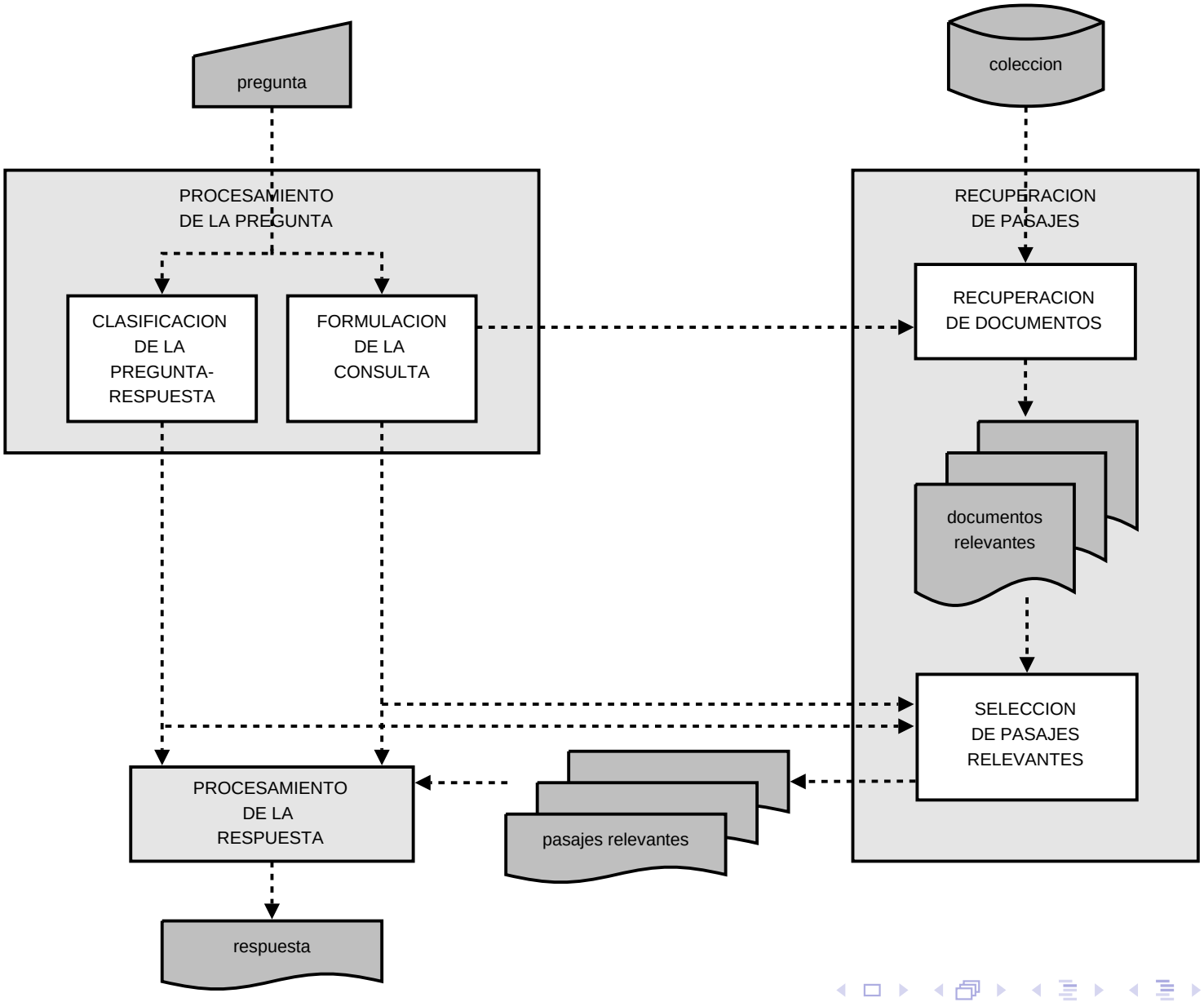
Tipos de Preguntas–Respuestas

- Complejidad del sistema dada por la complejidad de las preguntas y respuestas a procesar:

FACTOID:	*"¿Quién es el presidente francés?"*
DEFINITION:	*"¿Qué es el átomo?"*
	"¿Quién es Isabel Pantoja?"
TEMP. RESTR.:	*"¿Quién era el presidente de Uganda durante la guerra de Ruanda?"*
	"¿Qué equipo de Fórmula 1 ganó el G.P. de Hungría en 2004?"
LIST:	*"¿Cuáles son los 3 principales productos de exportación de Galicia?"*
EMBEDDED:	*"¿Cuándo murió el rey que sucedió a Alfonso X?"*
YES/NO:	*"¿Tenía hermanas Cervantes?"*
WHY-:	*"¿Por qué dimitió Nixon?"*
HOW-:	*"¿Cómo murió Alatriste?"*
...	...

- A partir de ahora nos restringiremos a **preguntas factuales (factoid questions)**: la respuesta es un hecho simple o una entidad (i.e. tipo *"Trivial Pursuit"*)

Arquitectura General



Arquitectura General (cont.)

I. **Procesamiento de la Pregunta** (*Question Processing*): procesa la pregunta para:

I.1. **Formulación de la Consulta** (*Query Formulation*): generar la consulta (*query*) con la que buscar en la colección los docs. relevantes para el tema de la pregunta (i.e. proceso de IR)

`Pregunta:` *"¿Quién es el presidente francés?"*
`Consulta:` presidente, francés

I.2. **Clasificación de la Pregunta–Respuesta** (*Question–Answer Type Classification*): identificar el tipo de respuesta esperada

- En *preguntas factuales* se refiere al **tipo de entidad** de la respuesta: persona, lugar, fecha, etc.

`Pregunta:` *"¿Quién es el presidente francés?"*
`Tipo respuesta:` persona (PER)

Arquitectura General (cont.)

II. **Recuperación de Pasajes** (*Passage Retrieval* *): buscar **fragmentos (pasajes) relevantes** para el tema de la pregunta entre los documentos de la colección. 2 fases:

   II.1. **Recuperación de Documentos** (*Document Retrieval*): con la consulta (*query*) generada previamente, buscar los docs. de la colección relevantes para el tema de la pregunta (i.e. proceso de IR)

   II.2. **Selección de Pasajes Relevantes** (*"Passage Retrieval *"*): procesar los docs. devueltos para localizar los fragmentos (*pasajes*) susceptibles de contener la respuesta

(*) En la literatura suele emplearse la misma denominación tanto para el proceso completo como para el segundo subproceso

Arquitectura General (cont.)

III. **Procesamiento de la Respuesta** (*Answer Processing*): extraer la respuesta de los fragmentos (pasajes) candidatos obtenidos. 2 aproximaciones:

   i. Devolver la **respuesta exacta**.

   ii. Devolver una **ventana de texto** del documento conteniendo la respuesta (**snippet**).

I.1. Formulación de la Consulta

- a.k.a. *Query Formulation*
- **Objetivo: generar la consulta** (*query*) con la que buscar en la colección los docs. relevantes para el tema de la pregunta (i.e. proceso de IR)

  `Pregunta:` *"¿Quién es el presidente francés?"*
  `Consulta:` presidente, francés

- **Aproximaciones:**
  - Usar la pregunta tal cual (i.e. delegamos en el sistema de IR)
  - Eliminar el interrogativo (*"quién"*, *"qué"*, *"where"*, *"when"*...)
  - Eliminar *stopwords*
  - Sólo NPs
  - *Query expansion*, especialmente con colecciones reducidas: menor redundancia (difícil se usen expresiones similares)
  - Variantes morfológicas, theasurus (ej. WordNet), *relevance feedback*, etc.
  - Reformulación (*reformulation*): reformular/reescribir la pregunta en forma de patrones de posibles respuestas *(Lin, 2007)*

    `Where is A ?` $\longrightarrow$ `/A is located in/`

    - Especialmente con colecciones grandes (ej. web): mayor redundancia

I.2. Clasificación de la Pregunta–Respuesta

- a.k.a. *Question–Answer Type Classification*
- **Objetivo: identificar el tipo de respuesta** esperada
  - En *preguntas factuales* suele referirse al **tipo de entidad** de la respuesta: persona, lugar, fecha, etc.

    `Pregunta:` *"¿Quién es el presidente francés?"*
    `Tipo respuesta:` persona (PER)
  - Permite **restringir la búsqueda** a aquellos segmentos de texto que contengan entidades de ese tipo
  - Permite generar **patrones de respuesta** para formatear la salida

    `What is A ?` $\longrightarrow$ `A is...`
- **Clases:** según tipo de entidad considerada
  - Refinable en forma de taxonomía

    HUMAN:
    BIOGRAPHY: *"¿Quién fue Confucio?"*
    INDIVIDUAL: *"¿Quién fue el primer hombre en el espacio?"*

I.2. Clasificación de la Pregunta–Respuesta (cont.)

Implementación del clasificador:

- Gral. en base a una **"question head-word" /"answer type word"**:
  - Interrogativos (*"quién"*, *"where"*, *"when"*...)
  - NP/NE al que se refiere el interrogativo (*"qué"*, *"cuál"*...):

    "¿Cuál es la ciudad más poblada de África?"

    "Which is the largest city of Africa?"

    - mediante análisis
    - mediante heurísticas: ej. el primero tras el interrogativo

- **Aproximaciones:**
  - Mediante reglas/patrones. Ej.:

    `/who {is|was|are|were} <PERSON>/` → BIOGRAPHY
  - Mediante aprendizaje automático: entrenado sobre repositorios de preguntas clasificadas manualmente

    "¿Quién fue Confucio?" BIOGRAPHY

    "¿Quién fue el primer hombre en el espacio?" INDIVIDUAL

II.1. Recuperación de Documentos

- a.k.a. *Document Retrieval*
- **Objetivo:** con la consulta (*query*) generada previamente, buscar los docs. de la colección relevantes para el tema de la pregunta (i.e. proceso de IR)
- **Aproximaciones:**
  i. Sistema de **IR "estándard"**
  ii. Sistema de **IR basado en pasajes** *(Kaszkiel & Zobel, 2001)*: sistemas de IR operando sobre unidades de texto más reducidas (*pasajes*) en lugar del documento completo
  iii. (**Web**) Posibilidad de trabajar sobre:
    - El documento completo (obtenible a partir de la URL)
    - Los *snippets* devueltos por el buscador

II.2. Selección de Pasajes Relevantes

- a.k.a. *Passage Retrieval*
- **Problema** con *Recuperación de Documentos*: aunque el documento parezca relevante, no tiene porqué contener siquiera una posible respuesta (ej. contiene los términos de la consulta por casualidad).
- **Objetivo:** procesar los docs. devueltos para localizar los fragmentos (*pasajes*) susceptibles de contener la respuesta.
- **Def. "pasaje"**: no existe consenso
  - Dependiente del sistema: según requerimientos
  - Granularidad variable: secciones, párrafos, N oraciones/líneas/palabras...
  - S/N superpuestos. Ej. pasajes de 5 líneas:

p1:	l1..l5	p1:	l1..l5
p2:	l6..l10	p2:	l2..l6
p3:	l11..l15	p3:	l3..l7

II.2. Selección de Pasajes Relevantes (cont.)

Proceso:

(1) **Segmentación** del doc. en pasajes

(2) **Filtrado** de los pasajes según s/n contienen respuestas potenciales (según s/n contengan entidades del tipo buscado)

(3) **Puntuar y ordenar** los pasajes seleccionados:

- Mediante:
  - Reglas/patrones
  - Aprendizaje automático
- En base a:
  - N° de entidades del tipo buscado
  - N° de términos de la pregunta que contiene
  - *Longest matching* con el texto de la pregunta (palabras y/o n-gramas)
  - Ranking del doc. al que pertenece
  - Proximidad entre los términos de la consulta (ej. ventana más corta)
  - (Web) *snippets*

III. Procesamiento de la Respuesta

- a.k.a. *Answer Processing*
- **Objetivo:** extraer la respuesta de los pasajes candidatos. 2 aproximaciones:
  i. Devolver **respuesta exacta**.
  ii. Devolver ventana de texto conteniendo la respuesta (**snippet**).
- **Aproximaciones** (combinables):
  i. Devolver la **entidad del tipo deseado** presente en el pasaje
  ii. Mediante **patrones de extracción** construidos manual o automáticamente. Ej.

DEFINITION QUESTIONS: *"What is autism ?"*

`/<AP> such as <QP>/` $=\sim$ "*[developmental disorders]*$_{AP}$ `such as` *[autism]*$_{QP}$"

- Permite filtrar si hay varias entidades del mismo tipo en el pasaje.
- Ídem cuando el tipo de la entidad a devolver no está clara (p.ej. *definitions*)
- Patrones específicos para cada tipo de pregunta
- Existen algoritmos de aprendizaje (más adelante)

Procesamiento de la Respuesta (cont.)

- **Aproximaciones**: (cont.)
  iii. Combinar con otros factores y/o ordenar las respuestas candidatas en base a ellos:
    - Mecanismos de votación
    - S/N se corresponden con el tipo de respuesta buscado
    - Qué patrón de extracción de respuesta matchea
    - Distancia entre la respuesta y los términos de la consulta
    - *Longest matching* con la pregunta
    - Ranking del pasaje origen
    - Factores de ordenación de los pasajes

  iv. (Web) *n-gram tiling*: construir una respuesta concatenando n-gramas de palabras provenientes de los *snippets* devueltos

III. Procesamiento de la Respuesta (cont.): Algoritmo de Aprendizaje de Patrones

- **Objetivo:** aprender **patrones de extracción de respuestas** para un tipo de pregunta dado
  - Relaciones *pregunta–respuesta*. Ej. *nombre_persona* $\leftrightarrow$ *año_nacimiento*

- Similar a *algoritmo de bootstrapping* visto en IE.

  (1) Partimos de tuplas de relaciones ya confirmadas. Ej.:
  (Mozart, 1756)

  (2) Buscamos en la web documentos que contengan dichos términos.

  (3) Seleccionamos oraciones que contengan ambos términos.

  (4) Generamos a partir de ellas **patrones de extracción** en base a las palabras y signos de puntuación alrededor de los términos de la tupla. Ej.:

  "(...) Mozart (1756–1791) (...)" $\Rightarrow$ `/<NAME> (<BD>-<DD>)/`
  "(...) Mozart was born on 1756 (...)" $\Rightarrow$ `/<NAME> was born on <BD>/`

  (5) Los filtramos, conservando únicamente aquéllos de alta precisión

NLP & QA

- Primeros sistemas QA eran una evolución de sistemas IR:
  - Únicamente técnicas de IR
  - Buen rendimiento durante recuperación
  - Pobre durante extracción

- **Solución:** introducción de técnicas de NLP en tareas de precisión:
  i. Análisis de la pregunta
  ii. Extracción de las respuestas

  Clasificación sistemas QA (d.p.d.v. NLP):
  - Clase 0: no emplean NLP
  - Clase 1: nivel léxico-sintáctico
  - Clase 2: nivel semántico
  - Clase 3: nivel contextual

Clase 0: No NLP

- **Sólo técnicas de IR**
- **Objetivo:** recuperar **extractos del texto** que contengan la respuesta esperada (i.e. ventanas de texto/*snippets*)
- Buen rendimiento ventanas grandes (∼250 char), pobre con pequeñas (∼50 char)
- *Procesamiento de la Pregunta*: seleccionar palabras clave suponemos cercanas a la respuesta
  - Eliminación de *stopwords*
  - Seleccionar las de mayor valor discriminativo
- *Procesamiento de la Respuesta*: aproximaciones:
  1. Mediante **ventanas de texto/snippets**:
     - Dividir texto relevante (doc./pasaje) en ventanas ($\leq$ longitud máxima permitida)
     - Ordenar dichas ventanas en base a:
       — Cuántas de las palabras clave contiene
       — Valor discriminativo de las mismas
       — Si su orden de aparición es s/n similar al de la pregunta
       — Proximidad de las palabras clave restantes (no contenidas) a la ventana

Clase 0: No NLP (cont.)

- *Procesamiento de la Respuesta*: aproximaciones: (cont.)
  II. Mediante **patrones de extracción** *(Soubbotin & Soubbotin, 2001)*:
    - Asociados a determinados tipos de preguntas
    - Puntuación asociada (según grado de fiabilidad)
    - Generados manualmente a partir de sus "respuestas tipo". Ej. *fechas nacimiento y fallecimiento*:

`/<Palabra_mayus> (\d\d\d\d–\d\d\d\d)/` $=\sim$ "*Mozart (1756–1791)*"

Clase 1: Nivel Léxico–Sintáctico

- **Modelo dominante**
- *Recuperación de Pasajes*: seguimos empleando IR
- **NER: prácticamente todos lo emplean**
  - *Clasificación de la Pregunta–Respuesta* (según tipo de entidad buscada)
  - *Selección de Pasajes Relevantes*: conservando sólo aquéllos que contienen entidades del tipo buscado
  - *Procesamiento de la Respuesta* con *snippets*: ídem pero con los *snippets*
  - *(Brill et al., 2001)* basado en la redundancia en la Web:

    (1) Reformular la pregunta reordenando los términos (+ entidad respuesta) de todas las formas posibles a modo de patrones de respuesta. Ej.:

    "When was Kennedy born?" ⇒ `/Kennedy was born on <DATE>/`
    `/born Kennedy on <DATE> was/`
    ...

    (2) Lanzar los patrones contra un buscador web:
    — Los malformados no devolverán documentos
    — Los correctos sí (por la redundancia de la web)

    (3) Nos quedamos con los que contengan entidades del tipo buscado

    (opcional, 4) Localizar la respuesta (obtenida de la web) en la colección original

Clase 1: Nivel Léxico–Sintáctico (cont.)

- **Etiquetación morfosintáctica (POS-tagging): uso muy extendido**
  - Fase previa al análisis sintáctico
- Lematización: eliminación de la variación flexiva
- Información sintáctica:
  - Técnicas de **análisis sintáctico superficial**
  - Objetivo: obtener la estructura sintáctica de cara a fases posteriores. Ej.:
    - *Clasificación de la Pregunta–Respuesta*: a veces necesario analizar estructura sintáctica. Ej.:
      "Which is the largest city of Africa?"
      → *"which"* (interrogativo) de por sí no indica el tipo a buscar
      → *"city"* (núcleo del NP) indica el tipo a buscar
    - *Procesamiento de la Respuesta*: similitud entre estructuras sintácticas pregunta ↔ respuestas candidatas

Clase 1: Nivel Léxico–Sintáctico (cont.)

- Información léxico-semántica:
  - Empleo de algoritmos de distancia semántica/conceptual
  - Empleo de técnicas de desambiguación del sentido de las palabras (WSD)
  - Para **expansión de la consulta** (ej. sinónimos WordNet)
    - Aplicable en diferentes fases del proceso: generación de la consulta, selección de pasajes, extracción de respuestas, etc.
  - Para implementar o complementar NER
  - Para *Clasificación de la Pregunta–Respuesta*

Clase 2: Nivel Semántico

- **Uso marginal** por su excesivo coste y complejidad
- Obtención de la **representación semántica** del texto (como Formas Lógicas, FL) mediante **análisis semántico**
- QA como **demostración de teoremas**: proceso de comparación entre las representaciones semánticas de preguntas y respuestas candidatas
  - `WolframAlpha` ??? (`http://www.wolframalpha.com/`)

Clase 3: Nivel Contextual

Básicamente restringido a la **resolución de correferencias**:

- Durante *Procesamiento de la Pregunta*: para series de preguntas encadenadas

  "¿Quién es el presidente francés?. ¿Cuántos años tiene?..."

- Durante *Procesamiento de la Respuesta*: para las coreferencias existentes en los pasajes seleccionados
  - Menos frecuente (mucho más costoso)

QA sobre Web

- Ya hemos visto algunas aproximaciones/técnicas a lo largo del tema
- Colección estática vs. web
  - **Estática:** acceso total a la colección
  - **Web:** acceso más reducido y costoso
    - **Solución:** restringirse a los primeros documentos devueltos (**top**)
- **Modificar proceso IR** para emplear un **buscador**:
  - Empleando algún API o *toolkit* del buscador
  - Implementando un interfaz propio:
    - Entrada al buscador (`url` con consulta): construir `url` correspondiente al buscador concreto empleado
      — Sintaxis variable
    - Salida del buscador (página HTML): parsear página resultados
      — Identificar URLs de los documentos devueltos
      — Descargarlos para procesarlos
      — Procesar los propios *snippets*

Colecciones de Referencia/Evaluación

- Composición: 3 elementos
  1. Documentos
  2. Preguntas
  3. Respuestas de referencia
- Más importantes (asociadas a instituciones/congresos):
  - TREC (TREC QA)
  - CLEF (QA@CLEF)
- A partir de ahora **QA@CLEF 2003/04** como referencia

Colecciones de Referencia/Evaluación (cont.)

1. **Documentos:**
   - Colección sobre la que **buscar y extraer la respuesta**.
   - Inicialmente los mismos empleadas en las *tracks* de RI.

2. **Preguntas:** 2 campos
   I. **Identificador** de la pregunta (numérico)
   II. **Texto** de la pregunta

   ```
   M SPA 0030 ¿Cuántos habitantes tiene Sidney?
   ```

3. **Respuestas de referencia:** 3 campos
   I. **Identificador** de la pregunta a la que responde
   II. Copia del texto de la pregunta (para facilitar revisión)
   III. **Respuesta(s):** (no necesariamente exhaustivo) 2 subcampos
      i. **Identificador del documento** del que se extrajo dicha respuesta.
      ii. **Texto de la respuesta**.

   ```
   0030|¿Cuántos habitantes tiene Sidney?|
      (EFE19940109-03287;Cuatro millones de habitantes),
      (EFE19940113-05570;3,4 millones de habitantes)
   ```

Respuestas a Devolver

- Par [texto_de_respuesta, docid]:
  - texto_de_respuesta: 2 aproximaciones:
    i. Devolver la **respuesta exacta**: más preciso, más complejo
       — ¿Cuál es la Capital de España?
       — Madrid
    ii. Devolver una **ventana de texto** del documento conteniendo la respuesta (**snippet**).
       — y también capital de España es Madrid, la cual
  - docid: **identificador del documento** del que se ha extraído la respuesta.
    - **Debe contener y justificar la respuesta** (i.e. al leerlo se desprende que ésa es la respuesta) y así evitar acertar *"por casualidad"*
- Devolver NIL si el sistema no encuentra respuesta
- A veces se permiten varias respuestas ordenadas por preferencia

Respuestas a Devolver: Ejemplos

```
M SPA 0045 ¿Cuál es la capital de España?
M SPA 0046 ¿Quién disparó a JR?
```

- **EJEMPLO 1** (respuesta exacta, 3 respuestas/pregunta):

```
45 plnnex031ms 1 3456 EFE19940115-75571 Madrid
45 plnnex031ms 2 678 EFE19940723-13794 Nápoles
45 plnnex031ms 3 500 EFE19950503-24532 París
46 plnnex031ms 1 7854 NIL
```

- **EJEMPLO 2** (respuesta exacta, 1 respuesta/pregunta):

```
45 plnnex031ms 1 3456 EFE19940115-75571 Madrid
46 plnnex031ms 1 7854 NIL
```

- **EJEMPLO 3** (ventana de texto, 1 respuesta/pregunta):

```
45 plnnst031ms 1 482.78 EFE19950308-23832 y también capital de España es Madrid, la cual
46 plnnex031ms 1 377 NIL
```

Comprobación de Respuestas

- **Tarea manual**: un evaluador humano examina las respuestas devueltas por el sistema y los documentos que las justifican
- **Casuística:** 4 valoraciones posibles:

1. **Incorrecta (W):** el `texto_de_respuesta` contiene una respuesta incorrecta o incompleta
2. **No justificada (U):** el `texto_de_respuesta` sí contiene una respuesta correcta, pero el documento indicado (`docid`) no la justifica
3. **Inexacta (X):** (sólo aplicable en el caso de sistemas de respuestas exacta) el `texto_de_respuesta` sí contiene una respuesta correcta y el documento indicado la justifica, pero en dicho *string* falta o sobra texto; es decir, la respuesta no exacta 100%

   — `¿Cuál es la Capital de España?`
   — `bella Madrid*`
   — `adrid*`
4. **Correcta (R):** el `texto_de_respuesta` es la respuesta exacta (en sistemas de respuesta exacta) o bien contiene la respuesta (en sistemas de ventanas) y el documento es justificativo.

Comprobación de Respuestas (cont.)

4. **Correcta (R):** (cont.) CASOS PARTICULARES:

   - Si un documento dice que Badajoz es la capital de España (lo cual sabemos que es incorrecto), y devolvemos Badajoz como respuesta justificándola con dicho documento, entonces dicha respuesta deberá igualmente ser evaluada como **Correcta (R)**.
   - Una **respuesta** `NIL` será considerada **Correcta (R)** sólo si no hay respuesta para dicha pregunta en la colección. Es decir, si el sistema ha devuelto `NIL` pero existen documentos donde aparece una respuesta, entonces se considerará **Incorrecta (W)**.

Permisividad

- Aún teniendo en cuenta lo dicho anteriormente, se pueden considerar 2 escenarios de evaluación:
  1. **"Estricto"** (por defecto): sólo se evalúan como respuestas correctas las respuestas tipo R (i.e. las correctas y justificadas).
  2. **"Permisivo"**: se aceptan también las respuestas tipo U (i.e. las correctas pero NO justificadas)
- Afectará a (ver más abajo):
  - *MRR*
  - Número de preguntas para las que se ha devuelto la respuesta correcta

Medidas de Evaluación: *MRR*

- Aplicable sólo si se permite devolver más de una respuesta.
- **Def. Reciprocal Rank (*RR*): inverso de la posición** en la que ha sido devuelta la respuesta correcta (i.e. si ha devuelto la respuesta en la posición n, su *RR* es $\frac{1}{n}$).
- **Def. Mean Reciprocal Rank (*MRR*):** media de los *RR* para un conjunto de preguntas.
  - Ejemplo: Dadas tres preguntas —q_1, q_2 y q_3—, para las dos primeras el sistema devuelve su respuesta correcta en segunda y primera posición, respectivamente, y para la última no obtenemos ninguna respuesta correcta.

$$RR_1 = \frac{1}{2} = 0.5 \quad RR_2 = \frac{1}{1} = 1 \quad RR_3 = 0 \quad MRR = \frac{0.5 + 1 + 0}{3} = 0.5$$

Medidas de Evaluación: Estadísticas

(QA@CLEF 2003: Se permitía devolver 3 respuestas ordenadas por pregunta)

- *MRR* calculada para el caso "estricto".
- Ídem para el caso "permisivo".
- Número de preguntas para las que se ha devuelto la respuesta correcta para el caso "estricto" (independientemente de su posición).
- Ídem para el caso "permisivo".
- Número de preguntas para las que se ha devuelto `NIL` como respuesta.
- Número de preguntas para las que se ha devuelto `NIL` como respuesta y ésta era además la respuesta correcta.

Referencias

- [CLEF, n.d.] Cross-Language Evaluation Forum. Site: `http://www.clef-campaign.org`
- [TREC, n.d.] Text REtrieval Conference. Site: `http://trec.nist.gov`
- [Brill et al., 2001] Brill, E., Lin, J., Banko, M. & Dumais, S. (2001). Data-intensive Question Answering. In *NIST Special Publication 500-250: The Tenth Text Retrieval Conference (TREC 10)*.
- [Jurafsky & Martin, 2009] Jurafsky, D. & Martin, J.H. (2009). Chapter 23: Question Answering and Summarization. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (2nd ed.)*. Pearson–Prentice Hall.
- [Kaszkiel & Zobel, 2001] Kaszkiel, M. & Zobel, J. (2001). Effective ranking with arbitrary passages. In *Journal of the American Society of Information Science*, 52(4):344–364
- [Lin, 2007] Lin, J. (2007). An exploration of the principles underlying redundancy-based factoid question answering. In *ACM Transactions on Information Systems (TOIS)*, 25(2):6.

Referencias (cont.)

- [Magini et al., 2003] Magnini, B., Romagnoli, S., Vallin, A., Herrera, J., Peñas, A., Peinado, V., Verdejo, M., de Rijke, M. & Vallin, R. (2003). The Multiple Language Question Answering Track at CLEF 2003. In *Results of the CLEF 2003 Cross-Language System Evaluation Campaign, Working Notes for the CLEF 2003 Workshop*. Disponible en `http://www.clef-campaign.org/`.

- [Pașca, 2003] Pașca, M. (2003). Chapter 7: Answer Extraction from Web Documents. *Open-domain question answering from large text collections*. CSLI Publications.

- [Soubbotin & Soubbotin, 2001] Soubbotin, M. & Soubbotin, S. (2001). Patterns of potential answer expressions as clues to the right answers. In *NIST Special Publication 500-250: The Tenth Text Retrieval Conference (TREC 10)*.

- [Vicedo, 2003] Vicedo, J.L. (2003). *Recuperación de información de alta precisión: los sistemas de búsqueda de respuestas*. Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN).