Question answering de dominio abierto y de dominio cerrado

Julián Peller

Julio 2016

index.tex

¿Qué es question answering?

Question Answering

Es el proceso automatizado de generación de respuestas concretas para preguntas formuladas en lenguaje natural.

Question

¿Quién desarrolló la teoría de la relatividad?

Answer

Albert Einstein.

¿Qué es question answering?

Question Answering

Es el proceso automatizado de generación de respuestas concretas para preguntas formuladas en lenguaje natural.

Question

¿Quién desarrolló la teoría de la relatividad?

Answer

Albert Einstein.

Ejemplo IR: esto no es QA

Figura: Google como sistema de Information Retrieval

IR es: "retornar *información relevante* para una *consulta* (*information need*) a partir de una *base de conocimiento*"

Ejemplo QA: esto es QA

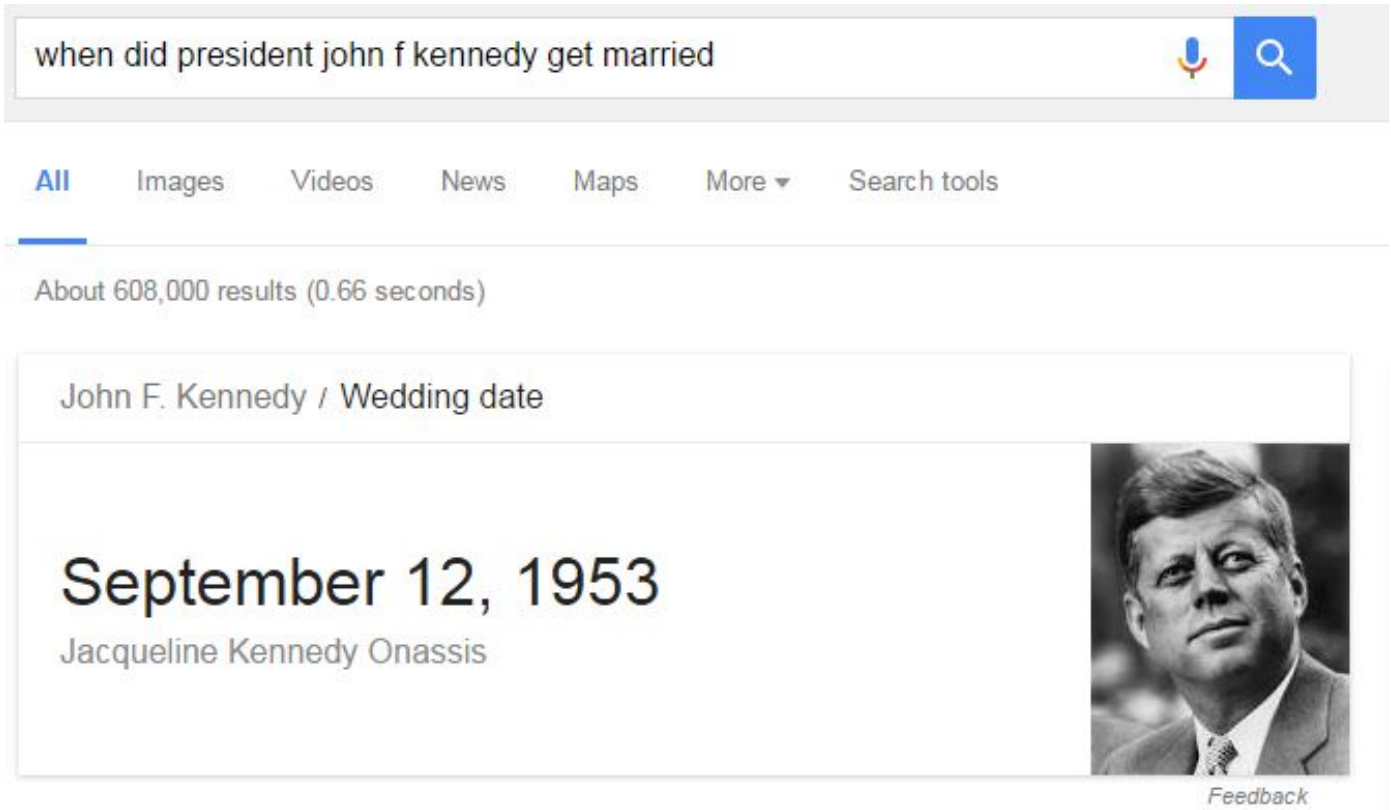


Figura: Google como sistema de Question Answering
QA es IR, pero más específico en qué significa *consulta* y qué significa *información relevante*

Ejes de clasificación

Generalidad del dominio

Dominio abierto o dominio cerrado

- Enciclopedias, Internet
- Restaurants, clima, transportes.

Tipo de datos

- Estructurado / no estructurado.
  - Bases de datos, texto plano, texto formateado.

Qué es esta tesis

Brevemente

Implementamos dos sistemas de question answering
(no, no hay hipótesis; es práctica)

Dominio cerrado con soporte para inglés / Popescu World

- Estructurado - Base de datos relacional: countries, cities, languages
- Modelo teórico de *tratabilidad semántica*
- Traduce preguntas a consultas SQL

Dominio abierto con soporte multilingüe / Multilingual Qanus

- No estructurado - Wikipedia(s) como base de conocimientos.
- Framework Qanus adaptado para Freeling
- Basado en IR + NLP + heurísticas

Qué es esta tesis

Brevemente

Implementamos dos sistemas de question answering
(no, no hay hipótesis; es práctica)

Dominio cerrado con soporte para inglés / Popescu World

- Estructurado - Base de datos relacional: countries, cities, languages
- Modelo teórico de *tratabilidad semántica*
- Traduce preguntas a consultas SQL

Dominio abierto con soporte multilingüe / Multilingual Qanus

- No estructurado - Wikipedia(s) como base de conocimientos.
- Framework Qanus adaptado para Freeling
- Basado en IR + NLP + heurísticas

Qué es esta tesis

Brevemente

Implementamos dos sistemas de question answering
(no, no hay hipótesis; es práctica)

Dominio cerrado con soporte para inglés / Popescu World

- Estructurado - Base de datos relacional: countries, cities, languages
- Modelo teórico de *tratabilidad semántica*
- Traduce preguntas a consultas SQL

Dominio abierto con soporte multilingüe / Multilingual Qanus

- No estructurado - Wikipedia(s) como base de conocimientos.
- Framework Qanus adaptado para Freeling
- Basado en IR + NLP + heurísticas

Qué es esta tesis

Brevemente

Implementamos dos sistemas de question answering
(no, no hay hipótesis; es práctica)

Dominio cerrado con soporte para inglés / Popescu World

- Estructurado - Base de datos relacional: countries, cities, languages
- Modelo teórico de *tratabilidad semántica*
- Traduce preguntas a consultas SQL

Dominio abierto con soporte multilingüe / Multilingual Qanus

- No estructurado - Wikipedia(s) como base de conocimientos.
- Framework Qanus adaptado para Freeling
- Basado en IR + NLP + heurísticas

Qué es esta tesis

Brevemente

Implementamos dos sistemas de question answering
(no, no hay hipótesis; es práctica)

Dominio cerrado con soporte para inglés / Popescu World

- Estructurado - Base de datos relacional: countries, cities, languages
- Modelo teórico de *tratabilidad semántica*
- Traduce preguntas a consultas SQL

Dominio abierto con soporte multilingüe / Multilingual Qanus

- No estructurado - Wikipedia(s) como base de conocimientos.
- Framework Qanus adaptado para Freeling
- Basado en IR + NLP + heurísticas

Qué es esta tesis

Brevemente

Implementamos dos sistemas de question answering
(no, no hay hipótesis; es práctica)

Dominio cerrado con soporte para inglés / Popescu World

- Estructurado - Base de datos relacional: countries, cities, languages
- Modelo teórico de *tratabilidad semántica*
- Traduce preguntas a consultas SQL

Dominio abierto con soporte multilingüe / Multilingual Qanus

- No estructurado - Wikipedia(s) como base de conocimientos.
- Framework Qanus adaptado para Freeling
- Basado en IR + NLP + heurísticas

Qué es esta tesis

Brevemente

Implementamos dos sistemas de question answering
(no, no hay hipótesis; es práctica)

Dominio cerrado con soporte para inglés / Popescu World

- Estructurado - Base de datos relacional: countries, cities, languages
- Modelo teórico de *tratabilidad semántica*
- Traduce preguntas a consultas SQL

Dominio abierto con soporte multilingüe / Multilingual Qanus

- No estructurado - Wikipedia(s) como base de conocimientos.
- Framework Qanus adaptado para Freeling
- Basado en IR + NLP + heurísticas

Qué es esta tesis

Brevemente

Implementamos dos sistemas de question answering
(no, no hay hipótesis; es práctica)

Dominio cerrado con soporte para inglés / Popescu World

- Estructurado - Base de datos relacional: countries, cities, languages
- Modelo teórico de *tratabilidad semántica*
- Traduce preguntas a consultas SQL

Dominio abierto con soporte multilingüe / Multilingual Qanus

- No estructurado - Wikipedia(s) como base de conocimientos.
- Framework Qanus adaptado para Freeling
- Basado en IR + NLP + heurísticas

Qué es esta tesis

Brevemente

Implementamos dos sistemas de question answering
(no, no hay hipótesis; es práctica)

Dominio cerrado con soporte para inglés / Popescu World

- Estructurado - Base de datos relacional: countries, cities, languages
- Modelo teórico de *tratabilidad semántica*
- Traduce preguntas a consultas SQL

Dominio abierto con soporte multilingüe / Multilingual Qanus

- No estructurado - Wikipedia(s) como base de conocimientos.
- Framework Qanus adaptado para Freeling
- Basado en IR + NLP + heurísticas

index.tex

Introducción

Modelo teórico: tratabilidad semántica (Popescu et al.)

- QADB: Question answering como interfaz para una base de datos
- **Tratabilidad semántica de una pregunta q en el contexto de una DB d.**
- Funcionalidad acotada y soporte solo para el inglés.

Pregunta

When was Albert Einstein born?

Consulta SQL

```
SELECT birth_date
FROM scientists
WHERE name = 'Albert Einstein'
```


Respuesta

March 14th, 1879

Introducción

Modelo teórico: tratabilidad semántica (Popescu et al.)

- QADB: Question answering como interfaz para una base de datos
- **Tratabilidad semántica de una pregunta q en el contexto de una DB d.**
- Funcionalidad acotada y soporte solo para el inglés.

Pregunta

When was Albert Einstein born?

Consulta SQL

```
SELECT birth_date
FROM scientists
WHERE name = 'Albert Einstein'
```


Respuesta

March 14th, 1879

Introducción

Modelo teórico: tratabilidad semántica (Popescu et al.)

- QADB: Question answering como interfaz para una base de datos
- **Tratabilidad semántica de una pregunta q en el contexto de una DB d.**
- Funcionalidad acotada y soporte solo para el inglés.

Pregunta

When was Albert Einstein born?

Consulta SQL

```
SELECT birth_date
FROM scientists
WHERE name = 'Albert Einstein'
```

Respuesta

March 14th, 1879

Introducción

Modelo teórico: tratabilidad semántica (Popescu et al.)

- QADB: Question answering como interfaz para una base de datos
- **Tratabilidad semántica de una pregunta q en el contexto de una DB d.**
- Funcionalidad acotada y soporte solo para el inglés.

Pregunta

When was Albert Einstein born?

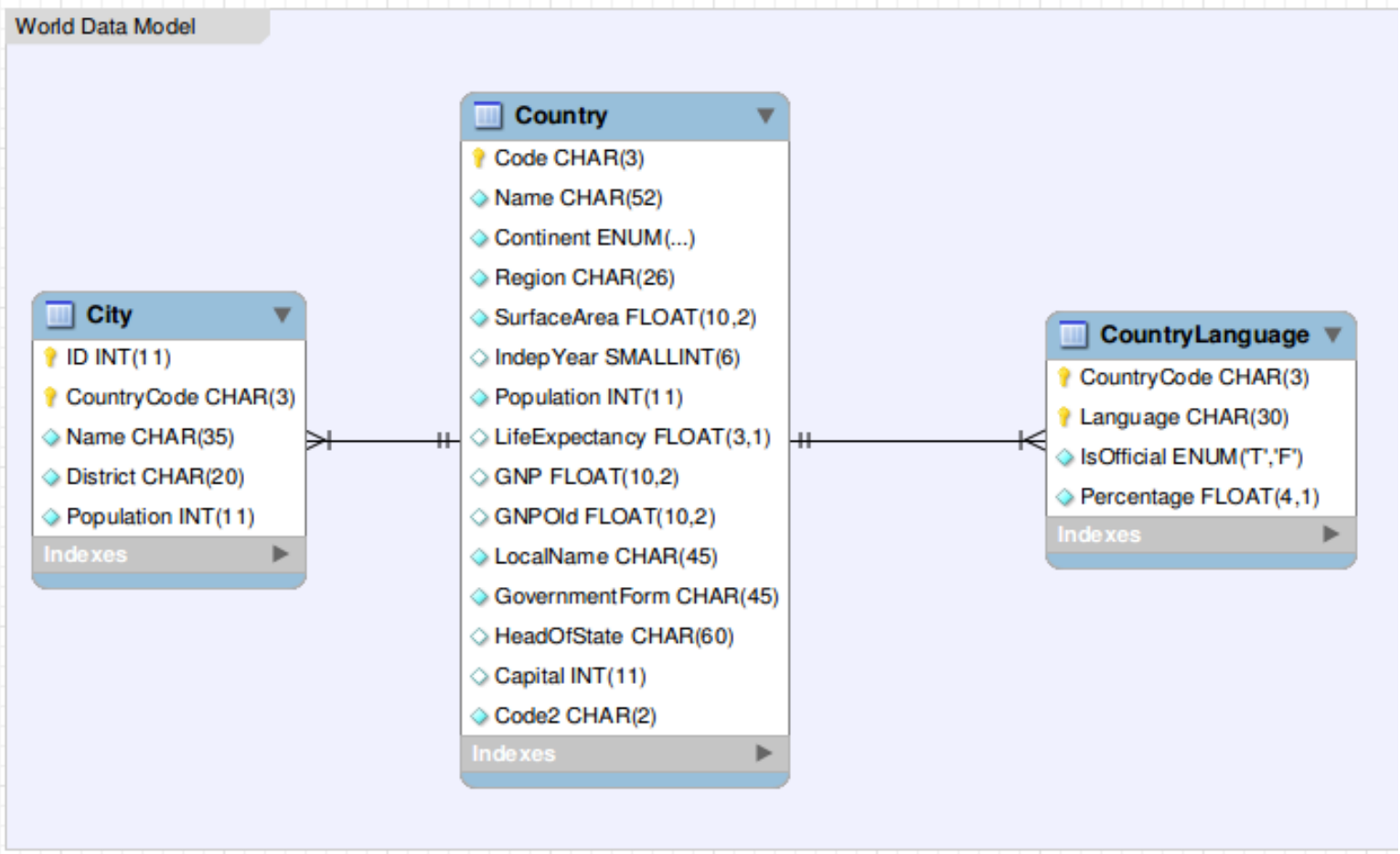
Consulta SQL

```
SELECT birth_date
FROM scientists
WHERE name = 'Albert Einstein'
```

Respuesta

March 14th, 1879

Base de datos: World (Country, City y CountryLanguage)



Tratabilidad semántica / motivación

- La complejidad de las preguntas en lenguaje natural es arbitraria.
- Distinguir un subconjunto 1) *tratable* (simple) y 2) *abarcativo*
  - La **tratabilidad semántica** formaliza esta clase
- Rechazar y pedir reformulación de las no tratables
  - Es mejor no dar respuesta a dar una mala.
  - Conservar la confianza en el sistema.

Tratable

La tratabilidad está dada por cualidades estructurales pero también por el contenido

- atributo = valor (forma de gobierno = monarquía)
- valor suelto - atributo implícito

Tratabilidad semántica / motivación

- La complejidad de las preguntas en lenguaje natural es arbitraria.
- Distinguir un subconjunto 1) *tratable* (simple) y 2) *abarcativo*
  - La **tratabilidad semántica** formaliza esta clase
- Rechazar y pedir reformulación de las no tratables
  - Es mejor no dar respuesta a dar una mala.
  - Conservar la confianza en el sistema.

Tratable

La tratabilidad está dada por cualidades estructurales pero también por el contenido

- atributo = valor (forma de gobierno = monarquía)
- valor suelto - atributo implícito

Tratabilidad semántica / motivación

- La complejidad de las preguntas en lenguaje natural es arbitraria.
- Distinguir un subconjunto 1) *tratable* (simple) y 2) *abarcativo*
  - La **tratabilidad semántica** formaliza esta clase
- Rechazar y pedir reformulación de las no tratables
  - Es mejor no dar respuesta a dar una mala.
  - Conservar la confianza en el sistema.

Tratable

La tratabilidad está dada por cualidades estructurales pero también por el contenido

- atributo = valor (forma de gobierno = monarquía)
- valor suelto - atributo implícito

Tratabilidad semántica / motivación

- La complejidad de las preguntas en lenguaje natural es arbitraria.
- Distinguir un subconjunto 1) *tratable* (simple) y 2) *abarcativo*
  - La **tratabilidad semántica** formaliza esta clase
- Rechazar y pedir reformulación de las no tratables
  - Es mejor no dar respuesta a dar una mala.
  - Conservar la confianza en el sistema.

Tratable

La tratabilidad está dada por cualidades estructurales pero también por el contenido

- atributo = valor (forma de gobierno = monarquía)
- valor suelto - atributo implícito

Tratabilidad semántica / motivación

- La complejidad de las preguntas en lenguaje natural es arbitraria.
- Distinguir un subconjunto 1) *tratable* (simple) y 2) *abarcativo*
  - La **tratabilidad semántica** formaliza esta clase
- Rechazar y pedir reformulación de las no tratables
  - Es mejor no dar respuesta a dar una mala.
  - Conservar la confianza en el sistema.

Tratable

La tratabilidad está dada por cualidades estructurales pero también por el contenido

- atributo = valor (forma de gobierno = monarquía)
- valor suelto - atributo implícito

Tratabilidad semántica / motivación

- La complejidad de las preguntas en lenguaje natural es arbitraria.
- Distinguir un subconjunto 1) *tratable* (simple) y 2) *abarcativo*
  - La **tratabilidad semántica** formaliza esta clase
- Rechazar y pedir reformulación de las no tratables
  - Es mejor no dar respuesta a dar una mala.
  - Conservar la confianza en el sistema.

Tratable

La tratabilidad está dada por cualidades estructurales pero también por el contenido

- atributo = valor (forma de gobierno = monarquía)
- valor suelto - atributo implícito

Tratabilidad semántica / motivación

- La complejidad de las preguntas en lenguaje natural es arbitraria.
- Distinguir un subconjunto 1) *tratable* (simple) y 2) *abarcativo*
  - La **tratabilidad semántica** formaliza esta clase
- Rechazar y pedir reformulación de las no tratables
  - Es mejor no dar respuesta a dar una mala.
  - Conservar la confianza en el sistema.

Tratable

La tratabilidad está dada por cualidades estructurales pero también por el contenido

- atributo = valor (forma de gobierno = monarquía)
- valor suelto - atributo implícito

Tratabilidad semántica / motivación

- La complejidad de las preguntas en lenguaje natural es arbitraria.
- Distinguir un subconjunto 1) *tratable* (simple) y 2) *abarcativo*
  - La **tratabilidad semántica** formaliza esta clase
- Rechazar y pedir reformulación de las no tratables
  - Es mejor no dar respuesta a dar una mala.
  - Conservar la confianza en el sistema.

Tratable

La tratabilidad está dada por cualidades estructurales pero también por el contenido

- atributo = valor (forma de gobierno = monarquía)
- valor suelto - atributo implícito

Tratabilidad semántica / motivación

- La complejidad de las preguntas en lenguaje natural es arbitraria.
- Distinguir un subconjunto 1) *tratable* (simple) y 2) *abarcativo*
  - La **tratabilidad semántica** formaliza esta clase
- Rechazar y pedir reformulación de las no tratables
  - Es mejor no dar respuesta a dar una mala.
  - Conservar la confianza en el sistema.

Tratable

La tratabilidad está dada por cualidades estructurales pero también por el contenido

- atributo = valor (forma de gobierno = monarquía)
- valor suelto - atributo implícito

Tratabilidad semántica / ejemplos

Ejemplos

1. ¿Qué países asiáticos son monarquías? → Sí
2. ¿Qué países de Asia no desarrollaron formas de gobierno basadas en las ideas de la Ilustración? → No
3. ¿Qué países del continente asiático tienen como forma de gobierno una monarquía? → Sí
4. ¿Qué países del continente que está al este de Europa y contiene a Rusia tiene esa forma de gobierno en la que manda una sola persona? → No
5. ¿Quiénes son los líderes de las monarquías asiáticas? → Sí

Tratabilidad semántica / ejemplos

Ejemplos

1. ¿Qué países asiáticos son monarquías? → Sí
2. ¿Qué países de Asia no desarrollaron formas de gobierno basadas en las ideas de la Ilustración? → No
3. ¿Qué países del continente asiático tienen como forma de gobierno una monarquía? → Sí
4. ¿Qué países del continente que está al este de Europa y contiene a Rusia tiene esa forma de gobierno en la que manda una sola persona? → No
5. ¿Quiénes son los líderes de las monarquías asiáticas? → Sí

Tratabilidad semántica / ejemplos

Ejemplos

1. ¿Qué países asiáticos son monarquías? → Sí
2. ¿Qué países de Asia no desarrollaron formas de gobierno basadas en las ideas de la Ilustración? → No
3. ¿Qué países del continente asiático tienen como forma de gobierno una monarquía? → Sí
4. ¿Qué países del continente que está al este de Europa y contiene a Rusia tiene esa forma de gobierno en la que manda una sola persona? → No
5. ¿Quiénes son los líderes de las monarquías asiáticas? → Sí

Tratabilidad semántica / ejemplos

Ejemplos

1. ¿Qué países asiáticos son monarquías? → Sí
2. ¿Qué países de Asia no desarrollaron formas de gobierno basadas en las ideas de la Ilustración? → No
3. ¿Qué países del continente asiático tienen como forma de gobierno una monarquía? → Sí
4. ¿Qué países del continente que está al este de Europa y contiene a Rusia tiene esa forma de gobierno en la que manda una sola persona? → No
5. ¿Quiénes son los líderes de las monarquías asiáticas? → Sí

Tratabilidad semántica / ejemplos

Ejemplos

1. ¿Qué países asiáticos son monarquías? → Sí
2. ¿Qué países de Asia no desarrollaron formas de gobierno basadas en las ideas de la Ilustración? → No
3. ¿Qué países del continente asiático tienen como forma de gobierno una monarquía? → Sí
4. ¿Qué países del continente que está al este de Europa y contiene a Rusia tiene esa forma de gobierno en la que manda una sola persona? → No
5. ¿Quiénes son los líderes de las monarquías asiáticas? → Sí

Tratabilidad semántica / ejemplos

Ejemplos

1. ¿Qué países asiáticos son monarquías? → Sí
2. ¿Qué países de Asia no desarrollaron formas de gobierno basadas en las ideas de la Ilustración? → No
3. ¿Qué países del continente asiático tienen como forma de gobierno una monarquía? → Sí
4. ¿Qué países del continente que está al este de Europa y contiene a Rusia tiene esa forma de gobierno en la que manda una sola persona? → No
5. ¿Quiénes son los líderes de las monarquías asiáticas? → Sí

Tratabilidad semántica / intuición

Una pregunta semánticamente tratable tiene:

- Atributos y Valores apareados
  - ¿Qué países tienen como forma de gobierno una monarquía?
- Valores sueltos (atributo implícito)
  - ¿Qué países son monarquías?
- Una Q-word (Qué, quién, cuándo) apareada con un atributo o suelta (atributo implícito)
  - ¿Quiénes son los líderes de las monarquías asiáticas?
  - ¿Qué países son monarquías?
- Marcadores sintácticos
- Menciones sueltas a relaciones

Tratabilidad semántica / intuición

Una pregunta semánticamente tratable tiene:

- Atributos y Valores apareados
  - ¿Qué países tienen como forma de gobierno una monarquía?
- Valores sueltos (atributo implícito)
  - ¿Qué países son monarquías?
- Una Q-word (Qué, quién, cuándo) apareada con un atributo o suelta (atributo implícito)
  - ¿Quiénes son los líderes de las monarquías asiáticas?
  - ¿Qué países son monarquías?
- Marcadores sintácticos
- Menciones sueltas a relaciones

Tratabilidad semántica / intuición

Una pregunta semánticamente tratable tiene:

- Atributos y Valores apareados
  - ¿Qué países tienen como forma de gobierno una monarquía?
- Valores sueltos (atributo implícito)
  - ¿Qué países son monarquías?
- Una Q-word (Qué, quién, cuándo) apareada con un atributo o suelta (atributo implícito)
  - ¿Quiénes son los líderes de las monarquías asiáticas?
  - ¿Qué países son monarquías?
- Marcadores sintácticos
- Menciones sueltas a relaciones

Tratabilidad semántica / intuición

Una pregunta semánticamente tratable tiene:

- Atributos y Valores apareados
  - ¿Qué países tienen como forma de gobierno una monarquía?
- Valores sueltos (atributo implícito)
  - ¿Qué países son monarquías?
- Una Q-word (Qué, quién, cuándo) apareada con un atributo o suelta (atributo implícito)
  - ¿Quiénes son los líderes de las monarquías asiáticas?
  - ¿Qué países son monarquías?
- Marcadores sintácticos
- Menciones sueltas a relaciones

Tratabilidad semántica / intuición

Una pregunta semánticamente tratable tiene:

- Atributos y Valores apareados
  - ¿Qué países tienen como forma de gobierno una monarquía?
- Valores sueltos (atributo implícito)
  - ¿Qué países son monarquías?
- Una Q-word (Qué, quién, cuándo) apareada con un atributo o suelta (atributo implícito)
  - ¿Quiénes son los líderes de las monarquías asiáticas?
  - ¿Qué países son monarquías?
- Marcadores sintácticos
- Menciones sueltas a relaciones

Tratabilidad semántica / intuición

Una pregunta semánticamente tratable tiene:

- Atributos y Valores apareados
  - ¿Qué países tienen como forma de gobierno una monarquía?
- Valores sueltos (atributo implícito)
  - ¿Qué países son monarquías?
- Una Q-word (Qué, quién, cuándo) apareada con un atributo o suelta (atributo implícito)
  - ¿Quiénes son los líderes de las monarquías asiáticas?
  - ¿Qué países son monarquías?
- Marcadores sintácticos
- Menciones sueltas a relaciones

Tratabilidad semántica / intuición

Una pregunta semánticamente tratable tiene:

- Atributos y Valores apareados
  - ¿Qué países tienen como forma de gobierno una monarquía?
- Valores sueltos (atributo implícito)
  - ¿Qué países son monarquías?
- Una Q-word (Qué, quién, cuándo) apareada con un atributo o suelta (atributo implícito)
  - ¿Quiénes son los líderes de las monarquías asiáticas?
  - ¿Qué países son monarquías?
- Marcadores sintácticos
- Menciones sueltas a relaciones

Tratabilidad semántica / intuición

Una pregunta semánticamente tratable tiene:

- Atributos y Valores apareados
  - ¿Qué países tienen como forma de gobierno una monarquía?
- Valores sueltos (atributo implícito)
  - ¿Qué países son monarquías?
- Una Q-word (Qué, quién, cuándo) apareada con un atributo o suelta (atributo implícito)
  - ¿Quiénes son los líderes de las monarquías asiáticas?
  - ¿Qué países son monarquías?
- Marcadores sintácticos
- Menciones sueltas a relaciones

Tratabilidad semántica / intuición

Una pregunta semánticamente tratable tiene:

- Atributos y Valores apareados
  - ¿Qué países tienen como forma de gobierno una monarquía?
- Valores sueltos (atributo implícito)
  - ¿Qué países son monarquías?
- Una Q-word (Qué, quién, cuándo) apareada con un atributo o suelta (atributo implícito)
  - ¿Quiénes son los líderes de las monarquías asiáticas?
  - ¿Qué países son monarquías?
- Marcadores sintácticos
- Menciones sueltas a relaciones

Tratabilidad semántica / intuición

Una pregunta semánticamente tratable tiene:

- Atributos y Valores apareados
  - ¿Qué países tienen como forma de gobierno una monarquía?
- Valores sueltos (atributo implícito)
  - ¿Qué países son monarquías?
- Una Q-word (Qué, quién, cuándo) apareada con un atributo o suelta (atributo implícito)
  - ¿Quiénes son los líderes de las monarquías asiáticas?
  - ¿Qué países son monarquías?
- Marcadores sintácticos
- Menciones sueltas a relaciones

Tratabilidad semántica / intuición

Abarcativo e intuitivamente traducible a una consulta SQL

- ¿Qué países del continente asiático tienen como forma de gobierno una monarquía?
  - Debería ser algo como: SELECT Country.Name FROM Country WHERE Continent=**Asia** AND GovernmentForm=**Monarchy**
- ¿Qué países asiáticos son monarquías?
  - Misma query, pero los atributos implícitos y deben "descubrirse"
- ¿Quiénes son los líderes de las monarquías asiáticas?
  - Q-word apareada explícitamente con atributo: SELECT Country.HeadOfState FROM Country WHERE Continent=**Asia** AND GovernmentForm=**Monarchy**

Tratabilidad semántica / intuición

Abarcativo e intuitivamente traducible a una consulta SQL

- ¿Qué países del continente asiático tienen como forma de gobierno una monarquía?
  - Debería ser algo como: SELECT Country.Name FROM Country WHERE Continent=**Asia** AND GovernmentForm=**Monarchy**
- ¿Qué países asiáticos son monarquías?
  - Misma query, pero los atributos implícitos y deben "descubrirse"
- ¿Quiénes son los líderes de las monarquías asiáticas?
  - Q-word apareada explícitamente con atributo: SELECT Country.HeadOfState FROM Country WHERE Continent=**Asia** AND GovernmentForm=**Monarchy**

Tratabilidad semántica / intuición

Abarcativo e intuitivamente traducible a una consulta SQL

- ¿Qué países del continente asiático tienen como forma de gobierno una monarquía?
  - Debería ser algo como: SELECT Country.Name FROM Country WHERE Continent=**Asia** AND GovernmentForm=**Monarchy**
- ¿Qué países asiáticos son monarquías?
  - Misma query, pero los atributos implícitos y deben "descubrirse"
- ¿Quiénes son los líderes de las monarquías asiáticas?
  - Q-word apareada explícitamente con atributo: SELECT Country.HeadOfState FROM Country WHERE Continent=**Asia** AND GovernmentForm=**Monarchy**

Tratabilidad semántica / intuición

Abarcativo e intuitivamente traducible a una consulta SQL

- ¿Qué países del continente asiático tienen como forma de gobierno una monarquía?
  - Debería ser algo como: SELECT Country.Name FROM Country WHERE Continent=**Asia** AND GovernmentForm=**Monarchy**
- ¿Qué países asiáticos son monarquías?
  - Misma query, pero los atributos implícitos y deben "descubrirse"
- ¿Quiénes son los líderes de las monarquías asiáticas?
  - Q-word apareada explícitamente con atributo: SELECT Country.HeadOfState FROM Country WHERE Continent=**Asia** AND GovernmentForm=**Monarchy**

Tratabilidad semántica / intuición

Abarcativo e intuitivamente traducible a una consulta SQL

- ¿Qué países del continente asiático tienen como forma de gobierno una monarquía?
  - Debería ser algo como: SELECT Country.Name FROM Country WHERE Continent=**Asia** AND GovernmentForm=**Monarchy**
- ¿Qué países asiáticos son monarquías?
  - Misma query, pero los atributos implícitos y deben "descubrirse"
- ¿Quiénes son los líderes de las monarquías asiáticas?
  - Q-word apareada explícitamente con atributo: SELECT Country.HeadOfState FROM Country WHERE Continent=**Asia** AND GovernmentForm=**Monarchy**

Tratabilidad semántica / intuición

Abarcativo e intuitivamente traducible a una consulta SQL

- ¿Qué países del continente asiático tienen como forma de gobierno una monarquía?
  - Debería ser algo como: SELECT Country.Name FROM Country WHERE Continent=**Asia** AND GovernmentForm=**Monarchy**
- ¿Qué países asiáticos son monarquías?
  - Misma query, pero los atributos implícitos y deben "descubrirse"
- ¿Quiénes son los líderes de las monarquías asiáticas?
  - Q-word apareada explicitamente con atributo: SELECT Country.HeadOfState FROM Country WHERE Continent=**Asia** AND GovernmentForm=**Monarchy**

Tratabilidad semántica / intuición

Abarcativo e intuitivamente traducible a una consulta SQL

- ¿Qué países del continente asiático tienen como forma de gobierno una monarquía?
  - Debería ser algo como: SELECT Country.Name FROM Country WHERE Continent=**Asia** AND GovernmentForm=**Monarchy**
- ¿Qué países asiáticos son monarquías?
  - Misma query, pero los atributos implícitos y deben "descubrirse"
- ¿Quiénes son los líderes de las monarquías asiáticas?
  - Q-word apareada explicitamente con atributo: SELECT Country.HeadOfState FROM Country WHERE Continent=**Asia** AND GovernmentForm=**Monarchy**

Módulos principales

Lexicón: dominio semántico

- Elementos de la DB → Wordnet → Lexicón (sinónimos)
- Todas las palabras que el sistema entiende están en el lexicón (tokens)
- Salto de las palabras a los elementos de la DB
  - region, part, area, neighborhood, realm → Atributo Country.Region
  - city, metropolis, urban center → Relación (tabla) City

Tokenizer: pregunta → tokens del lexicón

- Pregunta q → Marcadores sintácticos (se tiran) + términos del lexicón
- Si no se puede, la pregunta ya no es tratable
- Propone particiones de q en items del lexicón (tokenizaciones completas)

Matcher: (tokens → elementos) + apareo + desambiguación

- Filtra tokenizaciones válidas ("traducciones") con max flow (grafos)
- Descubre atributos implícitos
- Genera pares de atributos y valores y un atributo con la qword
- Desambigua sobreyecciones token → elemento al maximizar el flujo

Módulos principales

Lexicón: dominio semántico

- Elementos de la DB → Wordnet → Lexicón (sinónimos)
- Todas las palabras que el sistema entiende están en el lexicón (tokens)
- Salto de las palabras a los elementos de la DB
  - region, part, area, neighborhood, realm → Atributo Country.Region
  - city, metropolis, urban center → Relación (tabla) City

Tokenizer: pregunta → tokens del lexicón

- Pregunta q → Marcadores sintácticos (se tiran) + términos del lexicón
- Si no se puede, la pregunta ya no es tratable
- Propone particiones de q en items del lexicón (tokenizaciones completas)

Matcher: (tokens → elementos) + apareo + desambiguación

- Filtra tokenizaciones válidas ("traducciones") con max flow (grafos)
- Descubre atributos implícitos
- Genera pares de atributos y valores y un atributo con la qword
- Desambigua sobreyecciones token → elemento al maximizar el flujo

Módulos principales

Lexicón: dominio semántico

- Elementos de la DB → Wordnet → Lexicón (sinónimos)
- Todas las palabras que el sistema entiende están en el lexicón (tokens)
- Salto de las palabras a los elementos de la DB
  - region, part, area, neighborhood, realm → Atributo Country.Region
  - city, metropolis, urban center → Relación (tabla) City

Tokenizer: pregunta → tokens del lexicón

- Pregunta q → Marcadores sintácticos (se tiran) + términos del lexicón
- Si no se puede, la pregunta ya no es tratable
- Propone particiones de q en ítems del lexicón (tokenizaciones completas)

Matcher: (tokens → elementos) + apareo + desambiguación

- Filtra tokenizaciones válidas ("traducciones") con max flow (grafos)
- Descubre atributos implícitos
- Genera pares de atributos y valores y un atributo con la qword
- Desambigua sobreyecciones token → elemento al maximizar el flujo

Módulos principales

Lexicón: dominio semántico

- Elementos de la DB → Wordnet → Lexicón (sinónimos)
- Todas las palabras que el sistema entiende están en el lexicón (tokens)
- Salto de las palabras a los elementos de la DB
  - region, part, area, neighborhood, realm → Atributo Country.Region
  - city, metropolis, urban center → Relación (tabla) City

Tokenizer: pregunta → tokens del lexicón

- Pregunta q → Marcadores sintácticos (se tiran) + términos del lexicón
- Si no se puede, la pregunta ya no es tratable
- Propone particiones de q en items del lexicón (tokenizaciones completas)

Matcher: (tokens → elementos) + apareo + desambiguación

- Filtra tokenizaciones válidas ("traducciones") con max flow (grafos)
- Descubre atributos implícitos
- Genera pares de atributos y valores y un atributo con la qword
- Desambigua sobreyecciones token → elemento al maximizar el flujo

Módulos principales

Lexicón: dominio semántico

- Elementos de la DB → Wordnet → Lexicón (sinónimos)
- Todas las palabras que el sistema entiende están en el lexicón (tokens)
- Salto de las palabras a los elementos de la DB
  - region, part, area, neighborhood, realm → Atributo Country.Region
  - city, metropolis, urban center → Relación (tabla) City

Tokenizer: pregunta → tokens del lexicón

- Pregunta q → Marcadores sintácticos (se tiran) + términos del lexicón
- Si no se puede, la pregunta ya no es tratable
- Propone particiones de q en items del lexicón (tokenizaciones completas)

Matcher: (tokens → elementos) + apareo + desambiguación

- Filtra tokenizaciones válidas ("traducciones") con max flow (grafos)
- Descubre atributos implícitos
- Genera pares de atributos y valores y un atributo con la qword
- Desambigua sobreyecciones token → elemento al maximizar el flujo

Módulos principales

Lexicón: dominio semántico

- Elementos de la DB → Wordnet → Lexicón (sinónimos)
- Todas las palabras que el sistema entiende están en el lexicón (tokens)
- Salto de las palabras a los elementos de la DB
  - region, part, area, neighborhood, realm → Atributo Country.Region
  - city, metropolis, urban center → Relación (tabla) City

Tokenizer: pregunta → tokens del lexicón

- Pregunta q → Marcadores sintácticos (se tiran) + términos del lexicón
- Si no se puede, la pregunta ya no es tratable
- Propone particiones de q en items del lexicón (tokenizaciones completas)

Matcher: (tokens → elementos) + apareo + desambiguación

- Filtra tokenizaciones válidas ("traducciones") con max flow (grafos)
- Descubre atributos implícitos
- Genera pares de atributos y valores y un atributo con la qword
- Desambigua sobreyecciones token → elemento al maximizar el flujo

Módulos principales

Lexicón: dominio semántico

- Elementos de la DB → Wordnet → Lexicón (sinónimos)
- Todas las palabras que el sistema entiende están en el lexicón (tokens)
- Salto de las palabras a los elementos de la DB
  - region, part, area, neighborhood, realm → Atributo Country.Region
  - city, metropolis, urban center → Relación (tabla) City

Tokenizer: pregunta → tokens del lexicón

- Pregunta q → Marcadores sintácticos (se tiran) + términos del lexicón
- Si no se puede, la pregunta ya no es tratable
- Propone particiones de q en items del lexicón (tokenizaciones completas)

Matcher: (tokens → elementos) + apareo + desambiguación

- Filtra tokenizaciones válidas ("traducciones") con max flow (grafos)
- Descubre atributos implícitos
- Genera pares de atributos y valores y un atributo con la qword
- Desambigua sobreyecciones token → elemento al maximizar el flujo

Módulos principales

Lexicón: dominio semántico

- Elementos de la DB → Wordnet → Lexicón (sinónimos)
- Todas las palabras que el sistema entiende están en el lexicón (tokens)
- Salto de las palabras a los elementos de la DB
  - region, part, area, neighborhood, realm → Atributo Country.Region
  - city, metropolis, urban center → Relación (tabla) City

Tokenizer: pregunta → tokens del lexicón

- Pregunta q → Marcadores sintácticos (se tiran) + términos del lexicón
- Si no se puede, la pregunta ya no es tratable
- Propone particiones de q en items del lexicón (tokenizaciones completas)

Matcher: (tokens → elementos) + apareo + desambiguación

- Filtra tokenizaciones válidas ("traducciones") con max flow (grafos)
- Descubre atributos implícitos
- Genera pares de atributos y valores y un atributo con la qword
- Desambigua sobreyecciones token → elemento al maximizar el flujo

Módulos principales

Lexicón: dominio semántico

- Elementos de la DB → Wordnet → Lexicón (sinónimos)
- Todas las palabras que el sistema entiende están en el lexicón (tokens)
- Salto de las palabras a los elementos de la DB
  - region, part, area, neighborhood, realm → Atributo Country.Region
  - city, metropolis, urban center → Relación (tabla) City

Tokenizer: pregunta → tokens del lexicón

- Pregunta q → Marcadores sintácticos (se tiran) + términos del lexicón
- Si no se puede, la pregunta ya no es tratable
- Propone particiones de q en items del lexicón (tokenizaciones completas)

Matcher: (tokens → elementos) + apareo + desambiguación

- Filtra tokenizaciones válidas ("traducciones") con max flow (grafos)
- Descubre atributos implícitos
- Genera pares de atributos y valores y un atributo con la qword
- Desambigua sobreyecciones token → elemento al maximizar el flujo

Módulos principales

Lexicón: dominio semántico

- Elementos de la DB → Wordnet → Lexicón (sinónimos)
- Todas las palabras que el sistema entiende están en el lexicón (tokens)
- Salto de las palabras a los elementos de la DB
  - region, part, area, neighborhood, realm → Atributo Country.Region
  - city, metropolis, urban center → Relación (tabla) City

Tokenizer: pregunta → tokens del lexicón

- Pregunta q → Marcadores sintácticos (se tiran) + términos del lexicón
- Si no se puede, la pregunta ya no es tratable
- Propone particiones de q en items del lexicón (tokenizaciones completas)

Matcher: (tokens → elementos) + apareo + desambiguación

- Filtra tokenizaciones válidas ("traducciones") con max flow (grafos)
- Descubre atributos implícitos
- Genera pares de atributos y valores y un atributo con la qword
- Desambigua sobreyecciones token → elemento al maximizar el flujo

Módulos principales

Lexicón: dominio semántico

- Elementos de la DB → Wordnet → Lexicón (sinónimos)
- Todas las palabras que el sistema entiende están en el lexicón (tokens)
- Salto de las palabras a los elementos de la DB
  - region, part, area, neighborhood, realm → Atributo Country.Region
  - city, metropolis, urban center → Relación (tabla) City

Tokenizer: pregunta → tokens del lexicón

- Pregunta q → Marcadores sintácticos (se tiran) + términos del lexicón
- Si no se puede, la pregunta ya no es tratable
- Propone particiones de q en items del lexicón (tokenizaciones completas)

Matcher: (tokens → elementos) + apareo + desambiguación

- Filtra tokenizaciones válidas ("traducciones") con max flow (grafos)
- Descubre atributos implícitos
- Genera pares de atributos y valores y un atributo con la qword
- Desambigua sobreyecciones token → elemento al maximizar el flujo

Módulos principales

Lexicón: dominio semántico

- Elementos de la DB → Wordnet → Lexicón (sinónimos)
- Todas las palabras que el sistema entiende están en el lexicón (tokens)
- Salto de las palabras a los elementos de la DB
  - region, part, area, neighborhood, realm → Atributo Country.Region
  - city, metropolis, urban center → Relación (tabla) City

Tokenizer: pregunta → tokens del lexicón

- Pregunta q → Marcadores sintácticos (se tiran) + términos del lexicón
- Si no se puede, la pregunta ya no es tratable
- Propone particiones de q en items del lexicón (tokenizaciones completas)

Matcher: (tokens → elementos) + apareo + desambiguación

- Filtra tokenizaciones válidas ("traducciones") con max flow (grafos)
- Descubre atributos implícitos
- Genera pares de atributos y valores y un atributo con la qword
- Desambigua sobreyecciones token → elemento al maximizar el flujo

Módulos principales

Lexicón: dominio semántico

- Elementos de la DB → Wordnet → Lexicón (sinónimos)
- Todas las palabras que el sistema entiende están en el lexicón (tokens)
- Salto de las palabras a los elementos de la DB
  - region, part, area, neighborhood, realm → Atributo Country.Region
  - city, metropolis, urban center → Relación (tabla) City

Tokenizer: pregunta → tokens del lexicón

- Pregunta q → Marcadores sintácticos (se tiran) + términos del lexicón
- Si no se puede, la pregunta ya no es tratable
- Propone particiones de q en items del lexicón (tokenizaciones completas)

Matcher: (tokens → elementos) + apareo + desambiguación

- Filtra tokenizaciones válidas ("traducciones") con max flow (grafos)
- Descubre atributos implícitos
- Genera pares de atributos y valores y un atributo con la qword
- Desambigua sobreyecciones token → elemento al maximizar el flujo

Módulos principales

Lexicón: dominio semántico

- Elementos de la DB → Wordnet → Lexicón (sinónimos)
- Todas las palabras que el sistema entiende están en el lexicón (tokens)
- Salto de las palabras a los elementos de la DB
  - region, part, area, neighborhood, realm → Atributo Country.Region
  - city, metropolis, urban center → Relación (tabla) City

Tokenizer: pregunta → tokens del lexicón

- Pregunta q → Marcadores sintácticos (se tiran) + términos del lexicón
- Si no se puede, la pregunta ya no es tratable
- Propone particiones de q en items del lexicón (tokenizaciones completas)

Matcher: (tokens → elementos) + apareo + desambiguación

- Filtra tokenizaciones válidas ("traducciones") con max flow (grafos)
- Descubre atributos implícitos
- Genera pares de atributos y valores y un atributo con la qword
- Desambigua sobreyecciones token → elemento al maximizar el flujo

Módulos principales

Lexicón: dominio semántico

- Elementos de la DB → Wordnet → Lexicón (sinónimos)
- Todas las palabras que el sistema entiende están en el lexicón (tokens)
- Salto de las palabras a los elementos de la DB
  - region, part, area, neighborhood, realm → Atributo Country.Region
  - city, metropolis, urban center → Relación (tabla) City

Tokenizer: pregunta → tokens del lexicón

- Pregunta q → Marcadores sintácticos (se tiran) + términos del lexicón
- Si no se puede, la pregunta ya no es tratable
- Propone particiones de q en items del lexicón (tokenizaciones completas)

Matcher: (tokens → elementos) + apareo + desambiguación

- Filtra tokenizaciones válidas ("traducciones") con max flow (grafos)
- Descubre atributos implícitos
- Genera pares de atributos y valores y un atributo con la qword
- Desambigua sobreyecciones token → elemento al maximizar el flujo

Traducción válida / Query

Traducción válida

Cada máximo flujo genera pares atributo-valor y un par atributo-qword. Si sus tokens de atributo y de valor explícitos están sintácticamente asociados, entonces decimos que es una **traducción válida** de q

Semánticamente tratable

Una pregunta es **semánticamente tratable** si tiene una q-word y al menos una traducción válida

Una traducción válida de q se traduce trivialmente a una consulta SQL

SELECT	Atributo apareado con la qword
WHERE	Pares de atributos y valores apareados (implícitos y explícitos)
FROM	Todas las relaciones mencionadas

Traducción válida / Query

Traducción válida

Cada máximo flujo genera pares atributo-valor y un par atributo-qword. Si sus tokens de atributo y de valor explícitos están sintácticamente asociados, entonces decimos que es una **traducción válida** de q

Semánticamente tratable

Una pregunta es **semánticamente tratable** si tiene una q-word y al menos una traducción válida

Una traducción válida de q se traduce trivialmente a una consulta SQL

SELECT	Atributo apareado con la qword
WHERE	Pares de atributos y valores apareados (implícitos y explícitos)
FROM	Todas las relaciones mencionadas

Traducción válida / Query

Traducción válida

Cada máximo flujo genera pares atributo-valor y un par atributo-qword. Si sus tokens de atributo y de valor explícitos están sintácticamente asociados, entonces decimos que es una **traducción válida** de q

Semánticamente tratable

Una pregunta es **semánticamente tratable** si tiene una q-word y al menos una traducción válida

Una traducción válida de q se traduce trivialmente a una consulta SQL

SELECT	Atributo apareado con la qword
WHERE	Pares de atributos y valores apareados (implícitos y explícitos)
FROM	Todas las relaciones mencionadas

Ejemplo 1: “Who is the head of state of Zimbabwe?”

Ejemplo 1:

“Who is the head of state of Zimbabwe?”

Ejemplo 1: Tokenizer

1. Separar, sacar puntuaciones, pasar a lower case, tirar stopwords:

{who, ~~is~~, ~~the~~, head, ~~of~~, state, ~~of~~, zimbabwe}

Ejemplo 1: Tokenizer

1. Separar, sacar puntuaciones, pasar a lower case, tirar stopwords:

$\{\text{who}, \text{head}, \text{state}, \text{zimbabwe}\}$

Ejemplo 1: Tokenizer - tokens para who

2. Obtener tokens para cada lema (items del lexicón, sinónimo de 1+ elementos de la DB)

$$getTokens(who) \rightarrow \{\text{'who'}\}$$

Ejemplo 1: Tokenizer - tokens para head

2. Obtener tokens para cada lema

$getTokens(head) \rightarrow$ { 'head country', 'head teacher body politic', 'head body politic', 'head teacher land', 'read / write head dos', 'heading provincial', 'heading state', 'heading commonwealth', 'read / write head body politic', 'head word land', 'head state of matter', 'read / write head provincial', 'read / write head state department', 'head province', 'heading res publica', 'read / write head nation', (51 más)...}

Ejemplo 1: Tokenizer - tokens para state

2. Obtener tokens para cada lema

$getTokens(state) \rightarrow$ {'heart eastern united states', 'centre eastern united states', 'central eastern united states', 'centrical eastern united states', 'midsection eastern united states', 'midriff eastern united states', 'centric eastern united states', 'center eastern united states', 'north western united states', (702 más)...}

Ejemplo 1: Tokenizer - tokens para Zimbabwe

2. Obtener tokens para cada lema

$getTokens(zimbabwe) \rightarrow$ {'zimbabwe', 'republic of zimbabwe', 'capital of zimbabwe'}

Ejemplo 1: Tokenizer - intersección

3. Filtrar solo tokens cuyos lemas estén en la pregunta

$getTokens(zimbabwe) \rightarrow \{$'zimbabwe', ~~'republic of zimbabwe'~~, ~~'capital of zimbabwe'~~$\}$

Ejemplo 1: Tokenizer - intersección

3. Filtrar solo tokens cuyos lemas estén en la pregunta

$$getTokens(zimbabwe) \rightarrow \{\text{'zimbabwe'}\}$$
$$getTokens(who) \rightarrow \{\text{'who'}\}$$
$$getTokens(head) \rightarrow \{\text{'head state'}\}$$
$$getTokens(state) \rightarrow \{\text{'state','head state'}\}$$

Ejemplo 1: "Who is the head of state of Zimbabwe?"

5. Generar el conjunto de partes de todos contra todos (*tokenizaciones*).

{'state', 'zimbabwe'},
{ 'who', 'state', 'head state', 'zimbabwe'},
{ 'who', 'head state', 'zimbabwe'},
...

Tokenización completa

Quedarse solo con los que:

- Cubren la palabras no stopwords de la pregunta
- No tienen palabras repetidas

Ejemplo 1: Tokenizaciones completas

$q =$ "Who is the head of state of Zimbabwe?"

$CompleteTokenizations(q) = \{\{\text{who, head state, zimbabwe}\}\}$

Ejemplo 1: Lexicón - Tokens → Elementos

Obtener elementos de la DB para cada token

who → *Who* (Qword compatible con HeadOfState)

head state → *HeadOfState* (Atributo de Country)

zimbabwe → *Zimbabwe* (Valor de Country.Name)

Ejemplo 1: Matcher - Grafo de atributos y valores

Figura: 1. Nodos Fuente (S) y Sumidero (T)

Ejemplo 1: Matcher - Grafo de atributos y valores

Figura: 2. Tokens de valor (palabras), desde S, con $f = 1$

Ejemplo 1: Matcher - Grafo de atributos y valores

Figura: 3. Valores (DB) asociados a los tokens, con $f = 1$

Ejemplo 1: Matcher - Grafo de atributos y valores

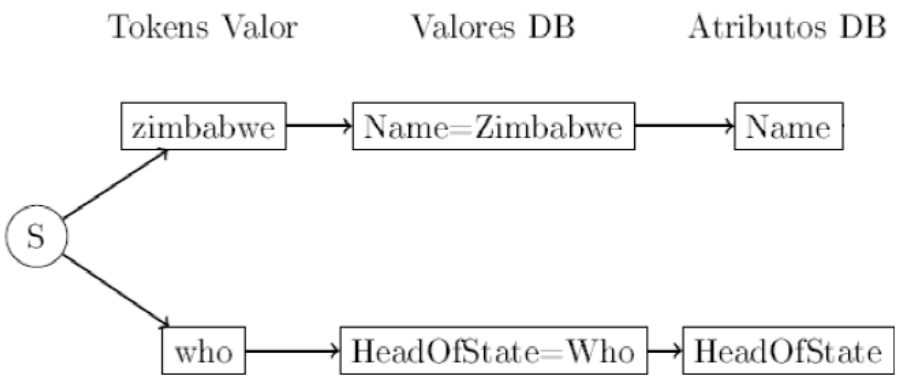


Figura: 4. Atributos (DB) asociados a los Valores, con $f = 1$

Ejemplo 1: Matcher - Grafo de atributos y valores

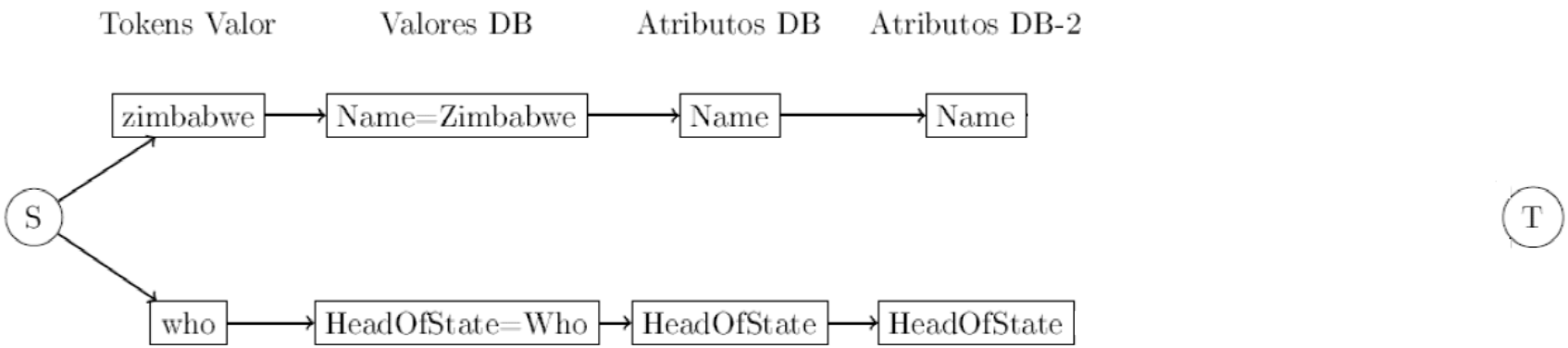


Figura: 5. Repite Atributos, para forzar $f = 1$ por Atributo

Ejemplo 1: Matcher - Grafo de atributos y valores

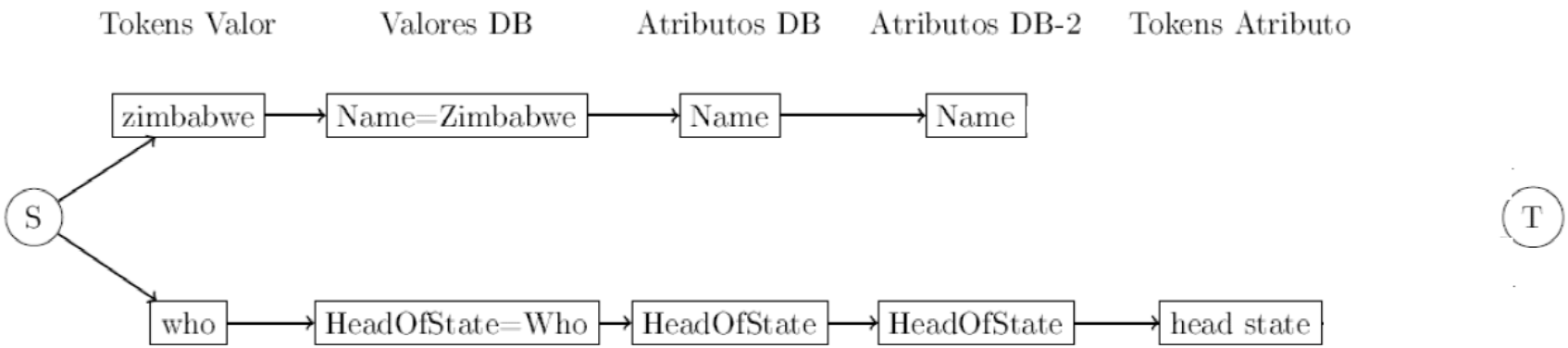


Figura: 6. Tokens de atributo, linkeados a Atributos con $f = 1$

Ejemplo 1: Matcher - Grafo de atributos y valores

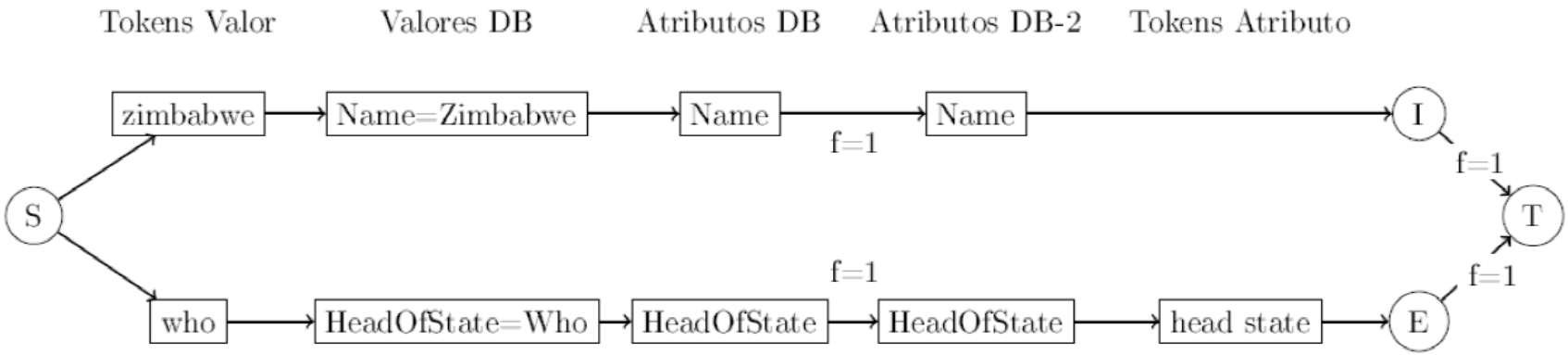


Figura: 6. Nodos I y E, con $f = tokens_{attr}$ y $f = tokens_{val} - tokens_{attr}$

Ejemplo 1: Matcher - notas

"Who ~~is~~ ~~the~~ head ~~of~~ state ~~of~~ Zimbabwe?"

- head state es el Atributo Country.HeadOfState
- Who es un Valor y está apareado con Country.HeadOfState
- Zimbabwe es un Valor y está apareado con Country.Name (implícito)
- No desambigua nada (los tokens tiene un solo elemento asociado)

Se chequea que los tokens de atributo y de valor explícitos apareados ('who' y 'head state') estén sintácticamente asociados

Ejemplo 1: Generador de Queries

Who ~~is~~ ~~the~~ head ~~of~~ state ~~of~~ Zimbabwe?

SELECT HeadOfState
FROM Country
WHERE Country.Name= 'Zimbabwe'

"Robert G. Mugabe"

Ejemplo 2: What caribbean countries are also considered north american?

Tokenizacion

- "What caribbean countries are also considered north american?"
- Tokenizaciones completas: {'north american', 'what', 'caribbean', 'country'}

Se obtienen todos los elementos que *corresponden* a cada token:

- caribbean → Caribbean (Valor de Country.Region **y** Valor de Country.Continent)
- north america → North America (Valor de Country.Continent)
- country → Country (Relación)
- what → What (Q-valor *compatible* con un montón de Atributos)

Ejemplo 2: What caribbean countries are also considered north american?

Tokenizacion

- "What caribbean countries are also considered north american?"
- Tokenizaciones completas: {'north american', 'what', 'caribbean', 'country'}

Se obtienen todos los elementos que *corresponden* a cada token:

- caribbean → Caribbean (Valor de Country.Region **y** Valor de Country.Continent)
- north america → North America (Valor de Country.Continent)
- country → Country (Relación)
- what → What (Q-valor *compatible* con un montón de Atributos)

Ejemplo 2: What caribbean countries are also considered north american?

Tokenizacion

- "What caribbean countries are also considered north american?"
- Tokenizaciones completas: {'north american', 'what', 'caribbean', 'country'}

Se obtienen todos los elementos que *corresponden* a cada token:

- caribbean → Caribbean (Valor de Country.Region **y** Valor de Country.Continent)
- north america → North America (Valor de Country.Continent)
- country → Country (Relación)
- what → What (Q-valor *compatible* con un montón de Atributos)

Ejemplo 2: What caribbean countries are also considered north american?

Tokenizacion

- "What caribbean countries are also considered north american?"
- Tokenizaciones completas: {'north american', 'what', 'caribbean', 'country'}

Se obtienen todos los elementos que *corresponden* a cada token:

- caribbean $\rightarrow$ Caribbean (Valor de Country.Region **y** Valor de Country.Continent)
- north america $\rightarrow$ North America (Valor de Country.Continent)
- country $\rightarrow$ Country (Relación)
- what $\rightarrow$ What (Q-valor *compatible* con un montón de Atributos)

Ejemplo 2: What caribbean countries are also considered north american?

Tokenizacion

- "What caribbean countries are also considered north american?"
- Tokenizaciones completas: {'north american', 'what', 'caribbean', 'country'}

Se obtienen todos los elementos que *corresponden* a cada token:

- caribbean → Caribbean (Valor de Country.Region **y** Valor de Country.Continent)
- north america → North America (Valor de Country.Continent)
- country → Country (Relación)
- what → What (Q-valor *compatible* con un montón de Atributos)

Ejemplo 2: What caribbean countries are also considered north american?

Tokenizacion

- "What caribbean countries are also considered north american?"
- Tokenizaciones completas: {'north american', 'what', 'caribbean', 'country'}

Se obtienen todos los elementos que *corresponden* a cada token:

- caribbean → Caribbean (Valor de Country.Region **y** Valor de Country.Continent)
- north america → North America (Valor de Country.Continent)
- country → Country (Relación)
- what → What (Q-valor *compatible* con un montón de Atributos)

Matcher: Grafo atributo-valor

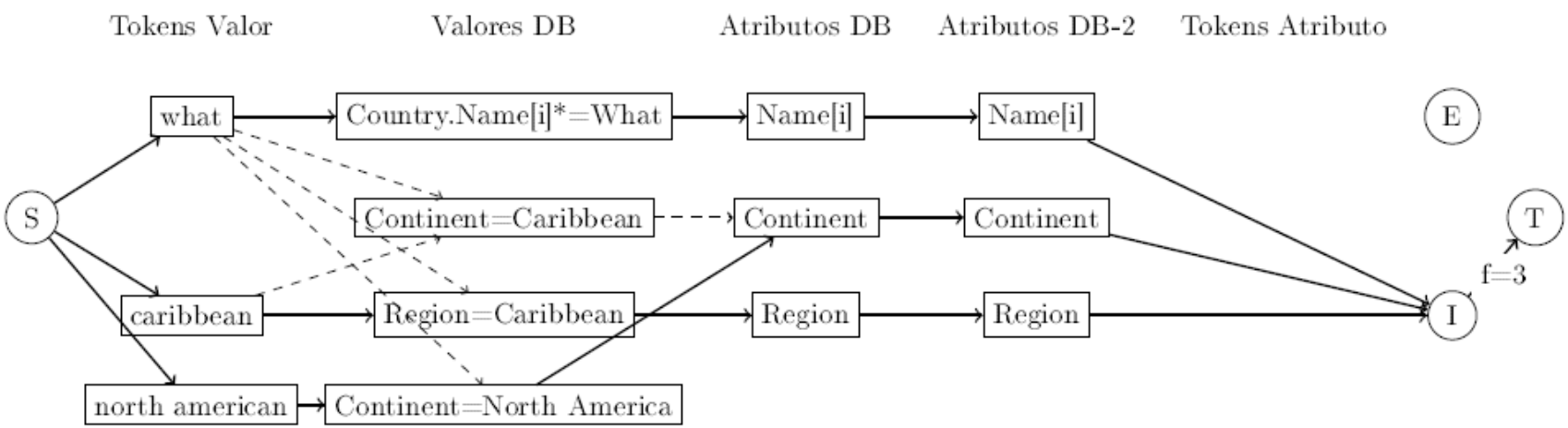


Figura: Grafo con desambiguación

Notar:

- Caribbean es ambigüo (What también)
- Como "north american" solo puede ser Continent, "caribbean" va a ser Region (para maximizar el flujo)

Ejemplo 2: Query

Query final:

```
SELECT DISTINCT Country.Name
FROM Country
WHERE Country.Continent = 'North America'
AND Country.Region = 'Caribbean'
```

Conclusiones y limitaciones: dominio cerrado

Sobre modelo dominio cerrado:

- Hipótesis fecunda del modelo: subset de preguntas. Tratabilidad semántica.
  - *Zonas* computables del lenguaje
- Modelo potente pero subespecificado (sinónimos, stopwords, attachment)
- La implementación requiere desarrollo humano dedicado y tuneo
- Killer combo con speech-to-text

Mejoras / trabajo futuro

- Generación del Lexicón
- Desambiguador de sentidos para los sinónimos de los tokens
- Generación de Stopwords
- CharniakParseTree: evaluar con un lingüista profesional las reglas
- Soportar diferentes paths para (*joinear*) tablas (solo soportamos uno canónico)
- Interfaz con el usuario: SQL $\rightarrow$ respuesta

index.tex

Introducción

- Sistema QA de dominio abierto con soporte multilingüe
- Basado en Qanus, adaptado para usar Freeling.
  - Pipeline - IR + NLP + Heurísticas
- Ejercicios de CLEF'07 monolingüe para español (es-es) y portugués (pt-pt)
  - Wikipedia como base de conocimiento (~~newsletters~~)
  - Factoids (~~lists~~, ~~definition~~)
  - Respuesta exacta. Nulas. Tópicos por grupo con co-referencia.
- Agregamos:
  - Heurísticas de generación de queries (de Lasso y de *topics*)
  - Heurística de ranking de pasajes
- Experimentamos y sacamos conclusiones.

Introducción

- Sistema QA de dominio abierto con soporte multilingüe
- Basado en Qanus, adaptado para usar Freeling.
  - Pipeline - IR + NLP + Heurísticas
- Ejercicios de CLEF'07 monolingüe para español (es-es) y portugués (pt-pt)
  - Wikipedia como base de conocimiento (~~newsletters~~)
  - Factoids (~~lists~~, ~~definition~~)
  - Respuesta exacta. Nulas. Tópicos por grupo con co-referencia.
- Agregamos:
  - Heurísticas de generación de queries (de Lasso y de *topics*)
  - Heurística de ranking de pasajes
- Experimentamos y sacamos conclusiones.

Introducción

- Sistema QA de dominio abierto con soporte multilingüe
- Basado en Qanus, adaptado para usar Freeling.
  - Pipeline - IR + NLP + Heurísticas
- Ejercicios de CLEF'07 monolingüe para español (es-es) y portugués (pt-pt)
  - Wikipedia como base de conocimiento (~~newsletters~~)
  - Factoids (~~lists~~, ~~definition~~)
  - Respuesta exacta. Nulas. Tópicos por grupo con co-referencia.
- Agregamos:
  - Heurísticas de generación de queries (de Lasso y de *topics*)
  - Heurística de ranking de pasajes
- Experimentamos y sacamos conclusiones.

Introducción

- Sistema QA de dominio abierto con soporte multilingüe
- Basado en Qanus, adaptado para usar Freeling.
  - Pipeline - IR + NLP + Heurísticas
- Ejercicios de CLEF'07 monolingüe para español (es-es) y portugués (pt-pt)
  - Wikipedia como base de conocimiento (~~newsletters~~)
  - Factoids (~~lists~~, ~~definition~~)
  - Respuesta exacta. Nulas. Tópicos por grupo con co-referencia.
- Agregamos:
  - Heurísticas de generación de queries (de Lasso y de *topics*)
  - Heurística de ranking de pasajes
- Experimentamos y sacamos conclusiones.

Introducción

- Sistema QA de dominio abierto con soporte multilingüe
- Basado en Qanus, adaptado para usar Freeling.
  - Pipeline - IR + NLP + Heurísticas
- Ejercicios de CLEF'07 monolingüe para español (es-es) y portugués (pt-pt)
  - Wikipedia como base de conocimiento (~~newsletters~~)
  - Factoids (~~lists~~, ~~definition~~)
  - Respuesta exacta. Nulas. Tópicos por grupo con co-referencia.
- Agregamos:
  - Heurísticas de generación de queries (de Lasso y de *topics*)
  - Heurística de ranking de pasajes
- Experimentamos y sacamos conclusiones.

Introducción

- Sistema QA de dominio abierto con soporte multilingüe
- Basado en Qanus, adaptado para usar Freeling.
  - Pipeline - IR + NLP + Heurísticas
- Ejercicios de CLEF'07 monolingüe para español (es-es) y portugués (pt-pt)
  - Wikipedia como base de conocimiento (~~newsletters~~)
  - Factoids (~~lists~~, ~~definition~~)
  - Respuesta exacta. Nulas. Tópicos por grupo con co-referencia.
- Agregamos:
  - Heurísticas de generación de queries (de Lasso y de *topics*)
  - Heurística de ranking de pasajes
- Experimentamos y sacamos conclusiones.

Introducción

- Sistema QA de dominio abierto con soporte multilingüe
- Basado en Qanus, adaptado para usar Freeling.
  - Pipeline - IR + NLP + Heurísticas
- Ejercicios de CLEF'07 monolingüe para español (es-es) y portugués (pt-pt)
  - Wikipedia como base de conocimiento (~~newsletters~~)
  - Factoids (~~lists~~, ~~definition~~)
  - Respuesta exacta. Nulas. Tópicos por grupo con co-referencia.
- Agregamos:
  - Heurísticas de generación de queries (de Lasso y de *topics*)
  - Heurística de ranking de pasajes
- Experimentamos y sacamos conclusiones.

Introducción

- Sistema QA de dominio abierto con soporte multilingüe
- Basado en Qanus, adaptado para usar Freeling.
  - Pipeline - IR + NLP + Heurísticas
- Ejercicios de CLEF'07 monolingüe para español (es-es) y portugués (pt-pt)
  - Wikipedia como base de conocimiento (~~newsletters~~)
  - Factoids (~~lists~~, ~~definition~~)
  - Respuesta exacta. Nulas. Tópicos por grupo con co-referencia.
- Agregamos:
  - Heurísticas de generación de queries (de Lasso y de *topics*)
  - Heurística de ranking de pasajes
- Experimentamos y sacamos conclusiones.

Introducción

- Sistema QA de dominio abierto con soporte multilingüe
- Basado en Qanus, adaptado para usar Freeling.
  - Pipeline - IR + NLP + Heurísticas
- Ejercicios de CLEF'07 monolingüe para español (es-es) y portugués (pt-pt)
  - Wikipedia como base de conocimiento (~~newsletters~~)
  - Factoids (~~lists~~, ~~definition~~)
  - Respuesta exacta. Nulas. Tópicos por grupo con co-referencia.
- Agregamos:
  - Heurísticas de generación de queries (de Lasso y de *topics*)
  - Heurística de ranking de pasajes
- Experimentamos y sacamos conclusiones.

Introducción

- Sistema QA de dominio abierto con soporte multilingüe
- Basado en Qanus, adaptado para usar Freeling.
  - Pipeline - IR + NLP + Heurísticas
- Ejercicios de CLEF'07 monolingüe para español (es-es) y portugués (pt-pt)
  - Wikipedia como base de conocimiento (~~newsletters~~)
  - Factoids (~~lists~~, ~~definition~~)
  - Respuesta exacta. Nulas. Tópicos por grupo con co-referencia.
- Agregamos:
  - Heurísticas de generación de queries (de Lasso y de *topics*)
  - Heurística de ranking de pasajes
- Experimentamos y sacamos conclusiones.

Introducción

- Sistema QA de dominio abierto con soporte multilingüe
- Basado en Qanus, adaptado para usar Freeling.
  - Pipeline - IR + NLP + Heurísticas
- Ejercicios de CLEF'07 monolingüe para español (es-es) y portugués (pt-pt)
  - Wikipedia como base de conocimiento (~~newsletters~~)
  - Factoids (~~lists~~, ~~definition~~)
  - Respuesta exacta. Nulas. Tópicos por grupo con co-referencia.
- Agregamos:
  - Heurísticas de generación de queries (de Lasso y de *topics*)
  - Heurística de ranking de pasajes
- Experimentamos y sacamos conclusiones.

Preguntas

Subset	Idioma	Factoid	Definition	List	Total
Todo	es	158	32	10	200
	pt	159	31	10	200
Wiki	es	**122**	24	9	163
	pt	**104**	18	8	130

Cuadro: Totales por tipo de pregunta

Ejemplo de grupo con tema y correferencia (primer cluster de preguntas para es-es):

- ¿En qué colegio estudia Harry Potter?
- ¿Cuál es el lema del colegio?
- ¿En qué casas está dividido?
- ¿Quién es el director del colegio?

Marco teórico / arquitectura de pipeline

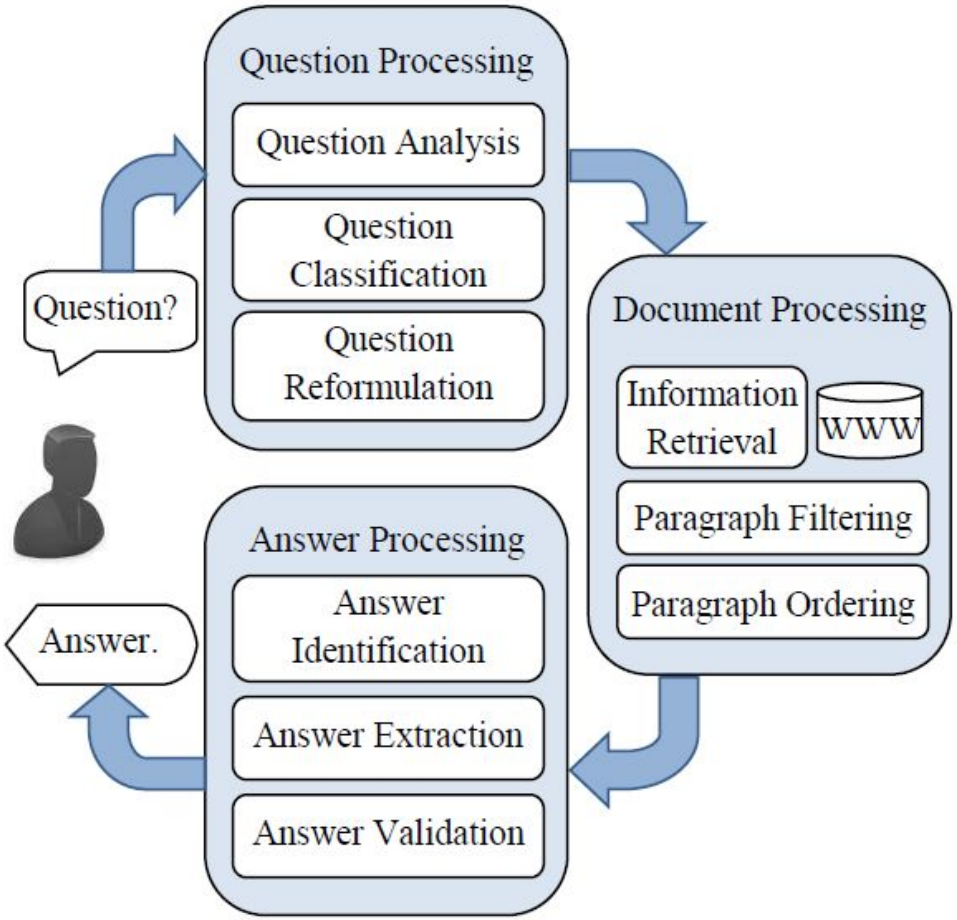


Figura: Módulos y submódulos de un sistema de QA (ver prox slide)

Marco teórico / arquitectura de pipeline

Módulo de procesamiento de la pregunta

- Submódulo principal: Question Classifier. Question Type
- Anotaciones de la pregunta (NER, POS)
- Creación de consulta para IR (Heurística de Lampert, Moldovan)

Módulo de procesamiento de documentos

- Submódulo principal: índice invertido (estructuras de IR)
- División en parágrafos
- Filtrado grueso
- Re-ranking de parágrafos

Módulo de procesamiento de la respuesta

- Heurísticas específicas basadas en casos.
- Identificación de respuestas candidatas y re rankeo
- Extracción de respuestas

Marco teórico / arquitectura de pipeline

Módulo de procesamiento de la pregunta

- Submódulo principal: Question Classifier. Question Type
- Anotaciones de la pregunta (NER, POS)
- Creación de consulta para IR (Heurística de Lampert, Moldovan)

Módulo de procesamiento de documentos

- Submódulo principal: índice invertido (estructuras de IR)
- División en parágrafos
- Filtrado grueso
- Re-ranking de parágrafos

Módulo de procesamiento de la respuesta

- Heurísticas específicas basadas en casos.
- Identificación de respuestas candidatas y re rankeo
- Extracción de respuestas

Marco teórico / arquitectura de pipeline

Módulo de procesamiento de la pregunta

- Submódulo principal: Question Classifier. Question Type
- Anotaciones de la pregunta (NER, POS)
- Creación de consulta para IR (Heurística de Lampert, Moldovan)

Módulo de procesamiento de documentos

- Submódulo principal: índice invertido (estructuras de IR)
- División en parágrafos
- Filtrado grueso
- Re-ranking de parágrafos

Módulo de procesamiento de la respuesta

- Heurísticas específicas basadas en casos.
- Identificación de respuestas candidatas y re rankeo
- Extracción de respuestas

Marco teórico / arquitectura de pipeline

Módulo de procesamiento de la pregunta

- Submódulo principal: Question Classifier. Question Type
- Anotaciones de la pregunta (NER, POS)
- Creación de consulta para IR (Heurística de Lampert, Moldovan)

Módulo de procesamiento de documentos

- Submódulo principal: índice invertido (estructuras de IR)
- División en parágrafos
- Filtrado grueso
- Re-ranking de parágrafos

Módulo de procesamiento de la respuesta

- Heurísticas específicas basadas en casos.
- Identificación de respuestas candidatas y re rankeo
- Extracción de respuestas

Marco teórico / arquitectura de pipeline

Módulo de procesamiento de la pregunta

- Submódulo principal: Question Classifier. Question Type
- Anotaciones de la pregunta (NER, POS)
- Creación de consulta para IR (Heurística de Lampert, Moldovan)

Módulo de procesamiento de documentos

- Submódulo principal: índice invertido (estructuras de IR)
- División en parágrafos
- Filtrado grueso
- Re-ranking de parágrafos

Módulo de procesamiento de la respuesta

- Heurísticas específicas basadas en casos.
- Identificación de respuestas candidatas y re rankeo
- Extracción de respuestas

Marco teórico / arquitectura de pipeline

Módulo de procesamiento de la pregunta

- Submódulo principal: Question Classifier. Question Type
- Anotaciones de la pregunta (NER, POS)
- Creación de consulta para IR (Heurística de Lampert, Moldovan)

Módulo de procesamiento de documentos

- Submódulo principal: índice invertido (estructuras de IR)
- División en parágrafos
- Filtrado grueso
- Re-ranking de parágrafos

Módulo de procesamiento de la respuesta

- Heurísticas específicas basadas en casos.
- Identificación de respuestas candidatas y re rankeo
- Extracción de respuestas

Marco teórico / arquitectura de pipeline

Módulo de procesamiento de la pregunta

- Submódulo principal: Question Classifier. Question Type
- Anotaciones de la pregunta (NER, POS)
- Creación de consulta para IR (Heurística de Lampert, Moldovan)

Módulo de procesamiento de documentos

- Submódulo principal: índice invertido (estructuras de IR)
- División en parágrafos
- Filtrado grueso
- Re-ranking de parágrafos

Módulo de procesamiento de la respuesta

- Heurísticas específicas basadas en casos.
- Identificación de respuestas candidatas y re rankeo
- Extracción de respuestas

Marco teórico / arquitectura de pipeline

Módulo de procesamiento de la pregunta

- Submódulo principal: Question Classifier. Question Type
- Anotaciones de la pregunta (NER, POS)
- Creación de consulta para IR (Heurística de Lampert, Moldovan)

Módulo de procesamiento de documentos

- Submódulo principal: índice invertido (estructuras de IR)
- División en parágrafos
- Filtrado grueso
- Re-ranking de parágrafos

Módulo de procesamiento de la respuesta

- Heurísticas específicas basadas en casos.
- Identificación de respuestas candidatas y re rankeo
- Extracción de respuestas

Marco teórico / arquitectura de pipeline

Módulo de procesamiento de la pregunta

- Submódulo principal: Question Classifier. Question Type
- Anotaciones de la pregunta (NER, POS)
- Creación de consulta para IR (Heurística de Lampert, Moldovan)

Módulo de procesamiento de documentos

- Submódulo principal: índice invertido (estructuras de IR)
- División en parágrafos
- Filtrado grueso
- Re-ranking de parágrafos

Módulo de procesamiento de la respuesta

- Heurísticas específicas basadas en casos.
- Identificación de respuestas candidatas y re rankeo
- Extracción de respuestas

Marco teórico / arquitectura de pipeline

Módulo de procesamiento de la pregunta

- Submódulo principal: Question Classifier. Question Type
- Anotaciones de la pregunta (NER, POS)
- Creación de consulta para IR (Heurística de Lampert, Moldovan)

Módulo de procesamiento de documentos

- Submódulo principal: índice invertido (estructuras de IR)
- División en parágrafos
- Filtrado grueso
- Re-ranking de parágrafos

Módulo de procesamiento de la respuesta

- Heurísticas específicas basadas en casos.
- Identificación de respuestas candidatas y re rankeo
- Extracción de respuestas

Marco teórico / arquitectura de pipeline

Módulo de procesamiento de la pregunta

- Submódulo principal: Question Classifier. Question Type
- Anotaciones de la pregunta (NER, POS)
- Creación de consulta para IR (Heurística de Lampert, Moldovan)

Módulo de procesamiento de documentos

- Submódulo principal: índice invertido (estructuras de IR)
- División en parágrafos
- Filtrado grueso
- Re-ranking de parágrafos

Módulo de procesamiento de la respuesta

- Heurísticas específicas basadas en casos.
- Identificación de respuestas candidatas y re rankeo
- Extracción de respuestas

Marco teórico / arquitectura de pipeline

Módulo de procesamiento de la pregunta

- Submódulo principal: Question Classifier. Question Type
- Anotaciones de la pregunta (NER, POS)
- Creación de consulta para IR (Heurística de Lampert, Moldovan)

Módulo de procesamiento de documentos

- Submódulo principal: índice invertido (estructuras de IR)
- División en parágrafos
- Filtrado grueso
- Re-ranking de parágrafos

Módulo de procesamiento de la respuesta

- Heurísticas específicas basadas en casos.
- Identificación de respuestas candidatas y re rankeo
- Extracción de respuestas

Marco teórico / arquitectura de pipeline

Módulo de procesamiento de la pregunta

- Submódulo principal: Question Classifier. Question Type
- Anotaciones de la pregunta (NER, POS)
- Creación de consulta para IR (Heurística de Lampert, Moldovan)

Módulo de procesamiento de documentos

- Submódulo principal: índice invertido (estructuras de IR)
- División en parágrafos
- Filtrado grueso
- Re-ranking de parágrafos

Módulo de procesamiento de la respuesta

- Heurísticas específicas basadas en casos.
- Identificación de respuestas candidatas y re rankeo
- Extracción de respuestas

Indice Invertido

ID	Text
1	Baseball is played during summer months.
2	Summer is the time for picnics here.
3	Months later we found out why.
4	Why is summer so hot here

↑

Documents

Inverse Index →

Term	Freq	Document ids
baseball	1	[1]
during	1	[1]
found	1	[3]
here	2	[2], [4]
hot	1	[4]
is	3	[1], [2], [4]
months	2	[1], [3]
summer	3	[1], [2], [4]
the	1	[2]
why	2	[3], [4]

Figura: Indice invertido

POS - part-of-speech tagging - etiquetado gramatical

Categoría Gramatical	Ejemplos
Determinante	aquel, este, mi, sus, nuestras
Pronombre	yo, tú, él, mí, nos, aquéllos, suyos
Preposición	a, ante, bajo, con, contra
Conjunción	y, aunque, pero, incluso
Interjección	ah, eh, ejem
Sustantivo	chicos, tesis, Pedro, cortapapeles
Verbo	cantamos, corrió, bailarán
Adjetivo	alegre, bonita, pésimos, desnuda
Adverbio	despacio, hábilmente, posteriormente

POS - "El hombre bajó la escalera."

Forma	Etiqueta	Descripción
El	DA0MS0	Determinante, Artículo, Masculino, Singular
hombre	NCMS000	Nombre, Común, Masculino, Singular
bajó	VMIS3S0	Verbo, Principal, Indicativo, Pasado, Tercera Persona, Singular
la	DA0FS0	Determinante, Artículo, Femenino, Singular
escalera	NCFS000	Nombre, Común, Femenino, Singular
.	Fp	Punto final

NER - Named entities recognition - Reconocimiento de Entidades Nombradas

Reconoce: Personas, Organizaciones, Lugares.

Lionel Messi nació el 24 de junio de 1987 en la ciudad de Rosario, en la provincia de Santa Fe. Es hijo de Jorge Horacio Messi, trabajador de una fábrica, y de Celia María Cuccittini, una limpiadora de medio tiempo (...).

Con apenas cinco años, Messi dio sus primeros pasos en Grandoli, un club de barrio al sur de Rosario, a pocas cuadras de su casa. El club estaba dirigido por su padre. Posteriormente jugó algunos partidos con el Central Córdoba. En 1995, comenzó a entrenarse en las divisiones inferiores de Newell's Old Boys.

NER - Named entities recognition - Reconocimiento de Entidades Nombradas

Reconoce: Personas, Organizaciones, Lugares.

Lionel Messi nació el 24 de junio de 1987 en la ciudad de Rosario, en la provincia de Santa Fe. Es hijo de Jorge Horacio Messi, trabajador de una fábrica, y de Celia María Cuccittini, una limpiadora de medio tiempo (...).

Con apenas cinco años, Messi dio sus primeros pasos en Grandoli, un club de barrio al sur de Rosario, a pocas cuadras de su casa. El club estaba dirigido por su padre. Posteriormente jugó algunos partidos con el Central Córdoba. En 1995, comenzó a entrenarse en las divisiones inferiores de Newell's Old Boys.

QC - Clases de preguntas

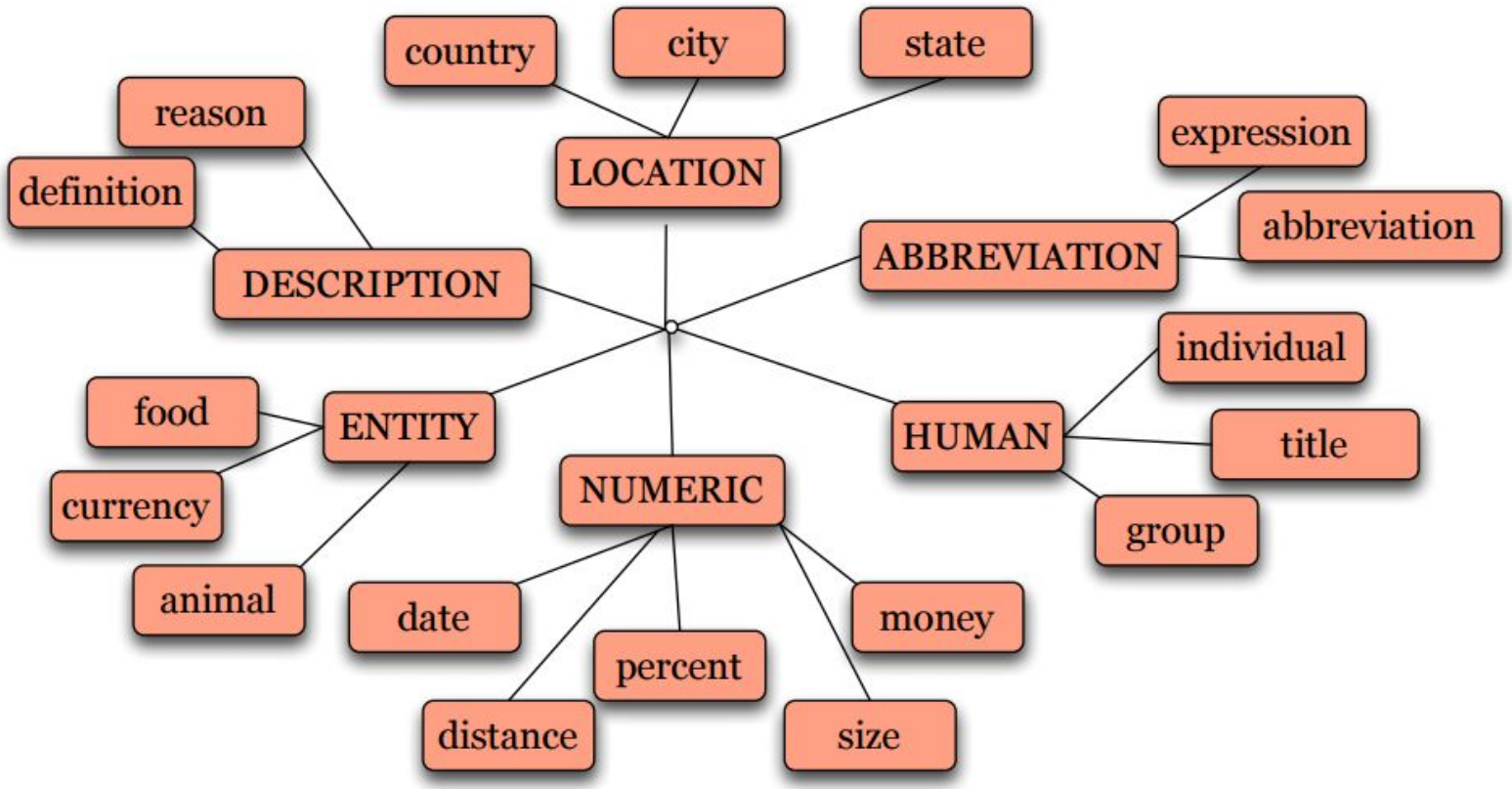


Figura: Clases de Question Type para el QC de Stanford

Qanus

Question-Answering @ National University of Singapore

- Un framework de question answering basado en information retrieval
- QA-sys: un sistema de QA funcional simple construido sobre este framework.
- Herramientas de NLP de Stanford (POS, NER y QC, para inglés) y Lucene.
- Estructura de Pipeline. Orientado para TREC 07 (Aquaint)

Qanus

Question-Answering @ National University of Singapore

- Un framework de question answering basado en information retrieval
- QA-sys: un sistema de QA funcional simple construido sobre este framework.
- Herramientas de NLP de Stanford (POS, NER y QC, para inglés) y Lucene.
- Estructura de Pipeline. Orientado para TREC 07 (Aquaint)

Qanus

Question-Answering @ National University of Singapore

- Un framework de question answering basado en information retrieval
- QA-sys: un sistema de QA funcional simple construido sobre este framework.
- Herramientas de NLP de Stanford (POS, NER y QC, para inglés) y Lucene.
- Estructura de Pipeline. Orientado para TREC 07 (Aquaint)

Qanus

Question-Answering @ National University of Singapore

- Un framework de question answering basado en information retrieval
- QA-sys: un sistema de QA funcional simple construido sobre este framework.
- Herramientas de NLP de Stanford (POS, NER y QC, para inglés) y Lucene.
- Estructura de Pipeline. Orientado para TREC 07 (Aquaint)

Qanus

Question-Answering @ National University of Singapore

- Un framework de question answering basado en information retrieval
- QA-sys: un sistema de QA funcional simple construido sobre este framework.
- Herramientas de NLP de Stanford (POS, NER y QC, para inglés) y Lucene.
- Estructura de Pipeline. Orientado para TREC 07 (Aquaint)

Qanus

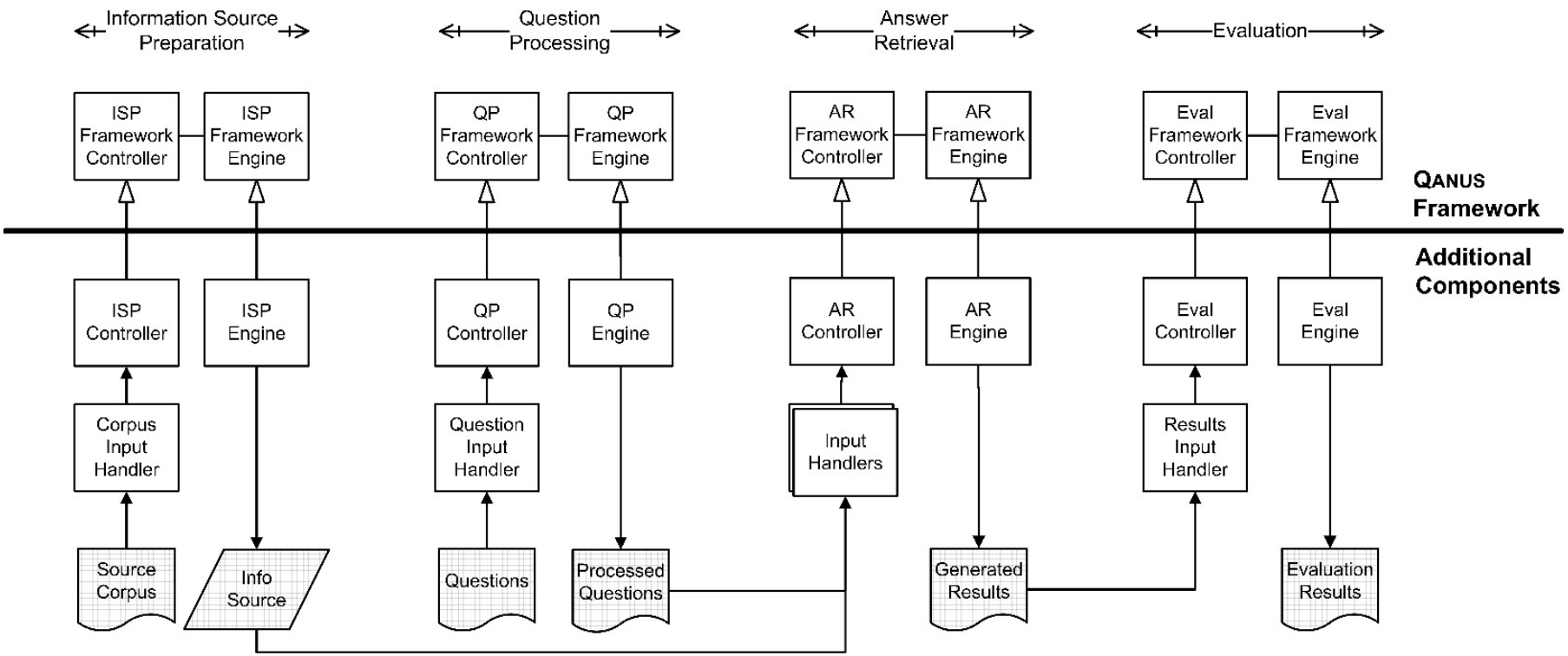


Figura: El framework Quanus y la implementación QA-sys

QA-Sys - Qanus @ TREC 07

Run Tag	Accuracy
LymbaPA07	0.706
LCCFerret	0.494
lsv2007c	0.289
UofL	0.258
QASCU1	0.256
FDUQAT16A	0.236
pronto07run3	0.222
ILQUA1	0.222
Ephyra3	0.208
QUANTA	0.206

Figura: Resultados de Trec 07

TREC 07

- Monolingue, Inglés.
- 200 preguntas
- Evaluada por humanos
- *Precisión*: respuestas exactas sobre respuestas totales
- LymbaPA $\rightarrow$ 0.706 (70 %)
- QA-Sys $\rightarrow$ 0.119 (12 %)

Baseline: QA-Sys

1. Preparación de la fuente de información: Preprocesa XML de Aquaint de manera offline en un índice Lucene.

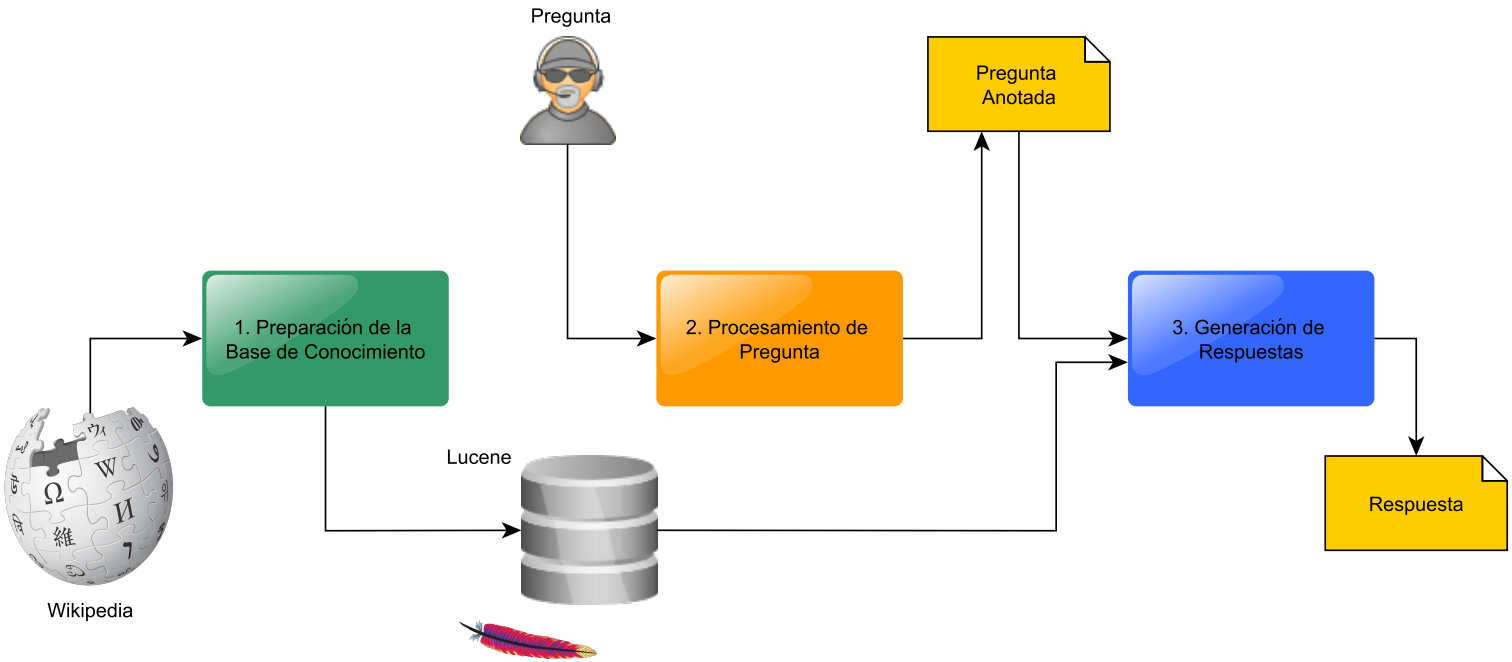
2. Análisis de la pregunta:
Anota la pregunta con POS, NER y QC.

3. Generación de respuestas: Busca la pregunta en el índice, separa los documentos en pasajes y los rankea con métricas heurísticas y luego, dependiendo del tipo de QC, aplica heurísticas basadas en NER, POS y QC sobre los pasajes para extraer una respuesta.

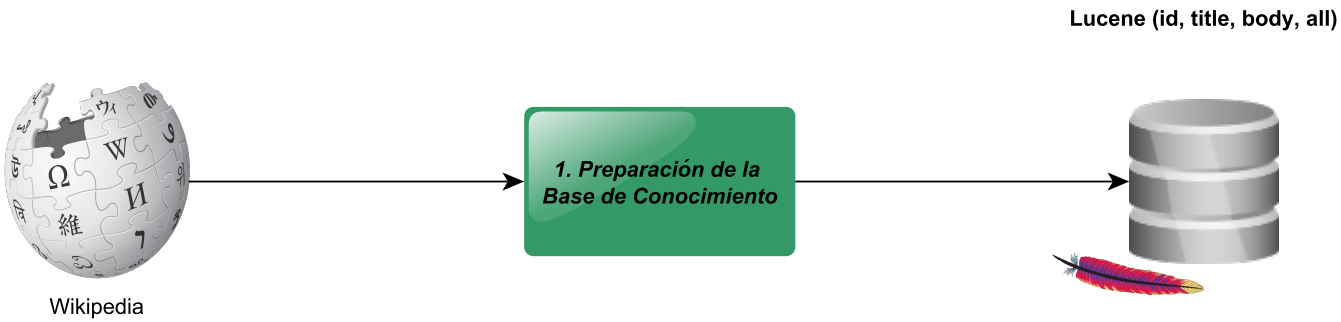
ML QA-Sys

- Soporte multi idioma no simultaneo.
  - Soporta varios idiomas, pero de a uno a la vez.
- Indice invertido de Wikipedia
- Reemplazamos las herramientas NLP por Freeling y sacamos reglas dependiente al idioma.

Pipeline


Pregunta
Pregunta Anotada
1. Preparación de la Base de Conocimiento
2. Procesamiento de Pregunta
3. Generación de Respuestas
Lucene
Respuesta
Wikipedia

1. Preparación de la base de conocimiento



Wikipedia	# Entradas	Nulas	Redirects	Filtradas	**Válidas**
es-06	**233750**	52	62805	34947	**135946**
pt-07	**498039**	80	210983	43390	**243586**

Cuadro: Wikipedias: cantidad de entradas válidas e inválidas

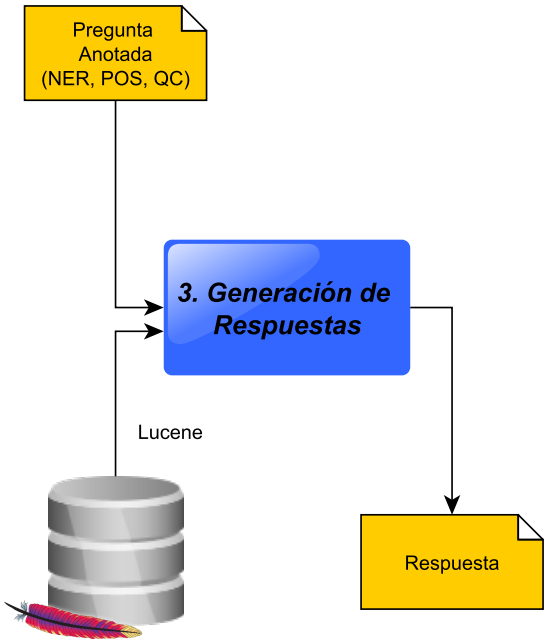
2. Procesamiento de la pregunta

- NER → Freeling
- POS → Freeling
- QC → No hay en ES ni PT
  - Aprovechando que estaban activas las tareas en-es y en-pt
  - Hack (truchada) → pasamos QC sobre preguntas en inglés

Clase	# ES	# PT
HUM	62	53
NUM	53	52
ENTY	34	28
DESC	25	30
LOC	24	36
ABBR	2	1
Total	200	200

Cuadro: Clases de las preguntas para ES y PT, Stanford Classifier (Li & Roth, Stanford)

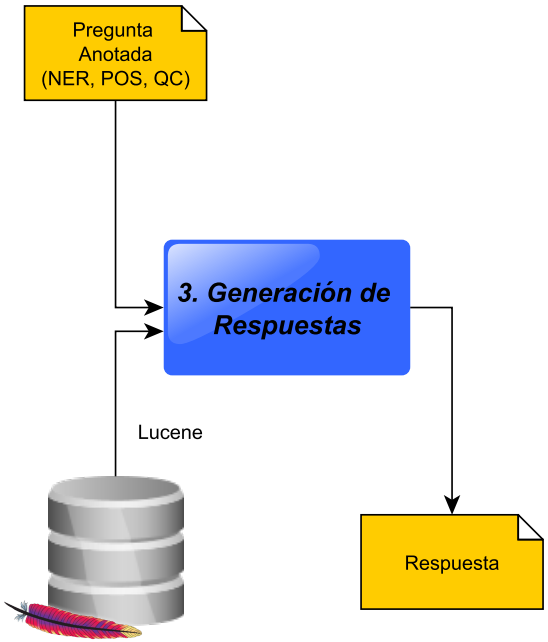
3. Generación de Respuestas



El algoritmo tiene los siguientes pasos:

1. Generación de Queries
2. Information Retrieval (documentos)
3. Extracción y rankeo de oraciones
4. POS tagging de las mejores oraciones
5. Heurísticas de AR basadas en el QC
   - Detectar respuestas candidatas
   - Rankear las respuestas candidatas
   - Extraer respuesta

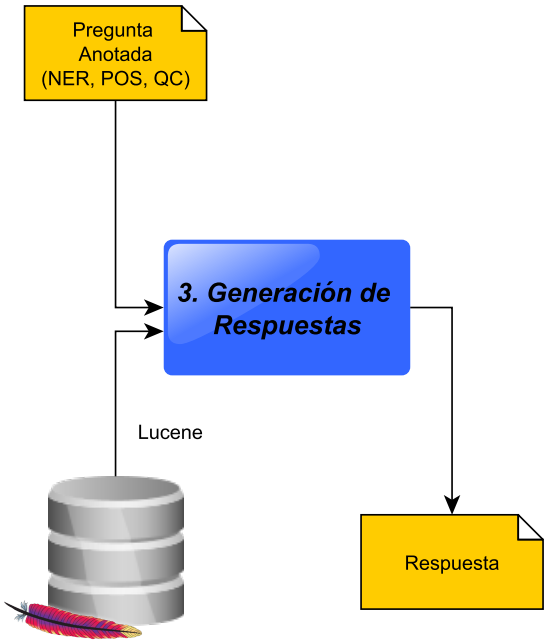
3. Generación de Respuestas



El algoritmo tiene los siguientes pasos:

1. Generación de Queries
2. Information Retrieval (documentos)
3. Extracción y rankeo de oraciones
4. POS tagging de las mejores oraciones
5. Heurísticas de AR basadas en el QC
   - Detectar respuestas candidatas
   - Rankear las candidatas
   - Extraer respuesta

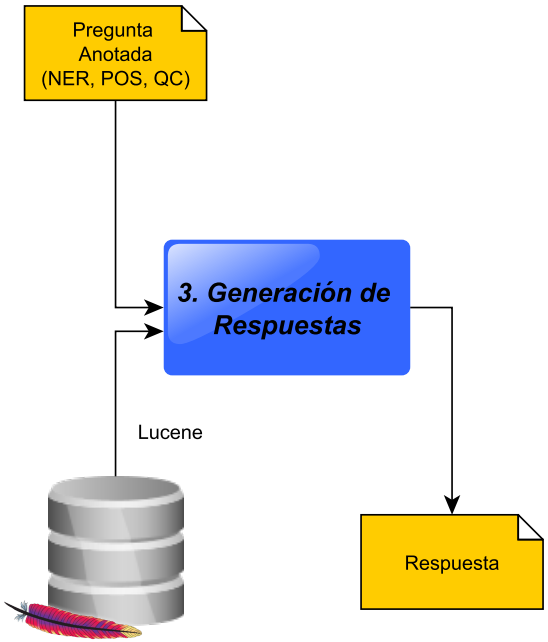
3. Generación de Respuestas



El algoritmo tiene los siguientes pasos:

1. Generación de Queries
2. Information Retrieval (documentos)
3. Extracción y rankeo de oraciones
4. POS tagging de las mejores oraciones
5. Heurísticas de AR basadas en el QC
   - Detectar respuestas candidatas
   - Re rankearlas según lógica específica
   - Extraer respuesta

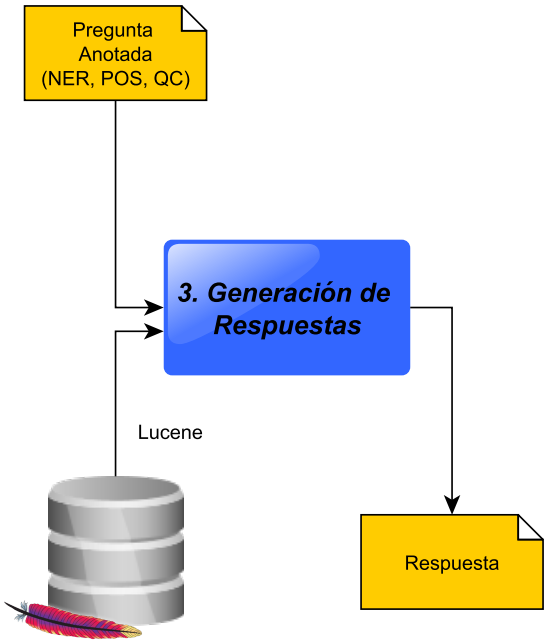
3. Generación de Respuestas



El algoritmo tiene los siguientes pasos:

1. Generación de Queries
2. Information Retrieval (documentos)
3. Extracción y rankeo de oraciones
4. POS tagging de las mejores oraciones
5. Heurísticas de AR basadas en el QC
   - Detectar respuestas candidatas
   - Re rankearlas según lógica específica
   - Extraer respuesta

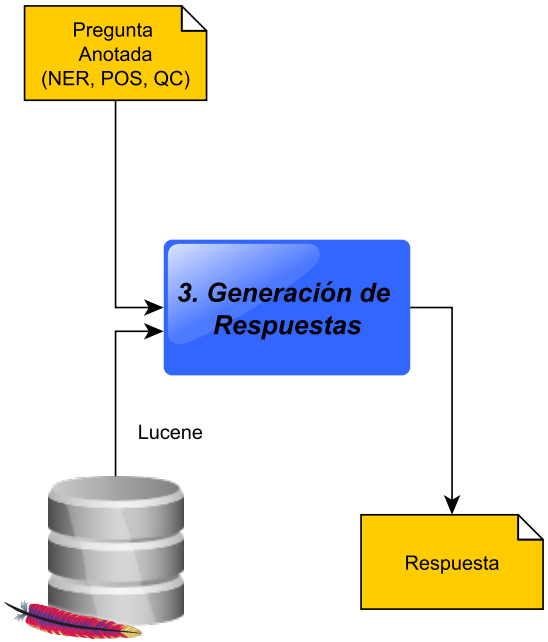
3. Generación de Respuestas



El algoritmo tiene los siguientes pasos:

1. Generación de Queries
2. Information Retrieval (documentos)
3. Extracción y rankeo de oraciones
4. POS tagging de las mejores oraciones
5. Heurísticas de AR basadas en el QC
   - Detectar respuestas candidatas
   - Re rankearlas según lógica específica
   - Extraer respuesta

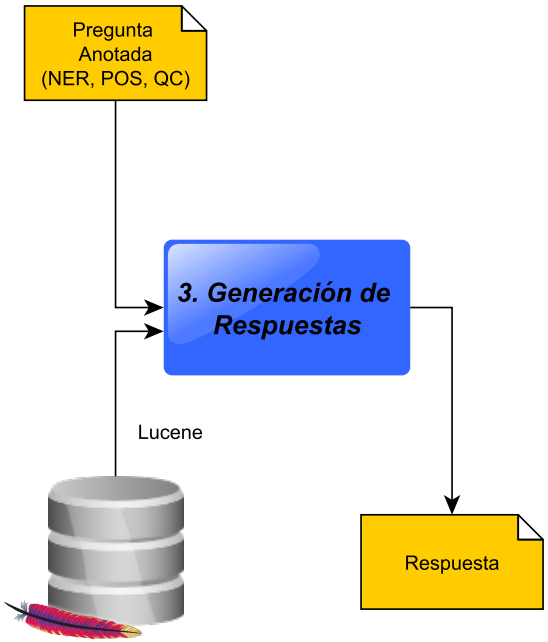
3. Generación de Respuestas



El algoritmo tiene los siguientes pasos:

1. Generación de Queries
2. Information Retrieval (documentos)
3. Extracción y rankeo de oraciones
4. POS tagging de las mejores oraciones
5. Heurísticas de AR basadas en el QC
   - Detectar respuestas candidatas
   - Re rankearlas según lógica específica
   - Extraer respuesta

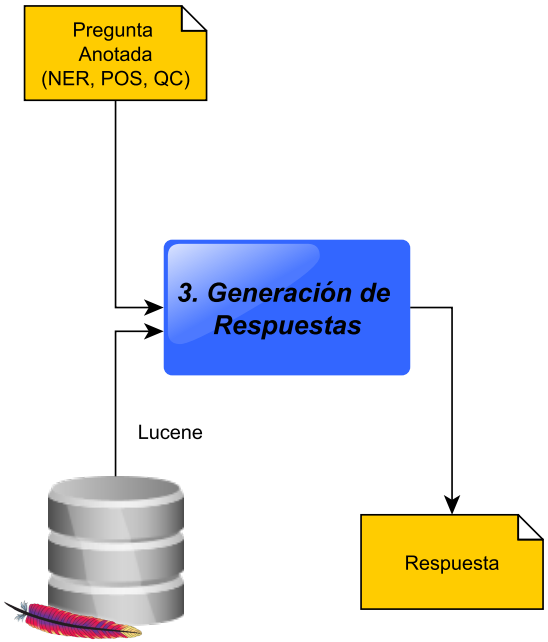
3. Generación de Respuestas



El algoritmo tiene los siguientes pasos:

1. Generación de Queries
2. Information Retrieval (documentos)
3. Extracción y rankeo de oraciones
4. POS tagging de las mejores oraciones
5. Heurísticas de AR basadas en el QC
   - Detectar respuestas candidatas
   - Re rankearlas según lógica específica
   - Extraer respuesta

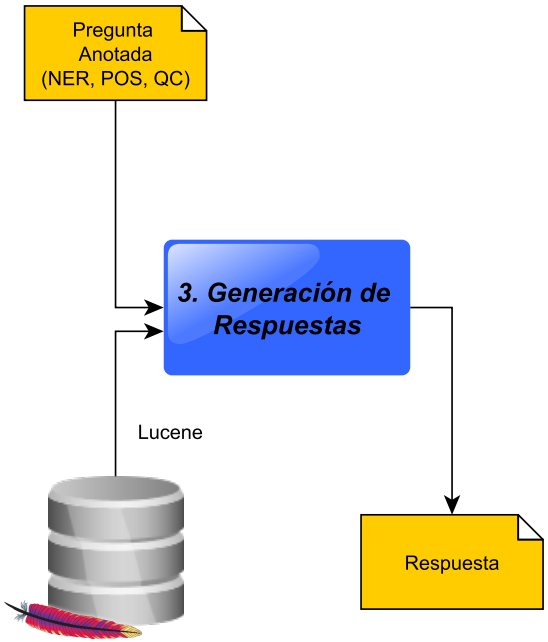
3. Generación de Respuestas



El algoritmo tiene los siguientes pasos:

1. Generación de Queries
2. Information Retrieval (documentos)
3. Extracción y rankeo de oraciones
4. POS tagging de las mejores oraciones
5. Heurísticas de AR basadas en el QC
   - Detectar respuestas candidatas
   - Re rankearlas según lógica específica
   - Extraer respuesta

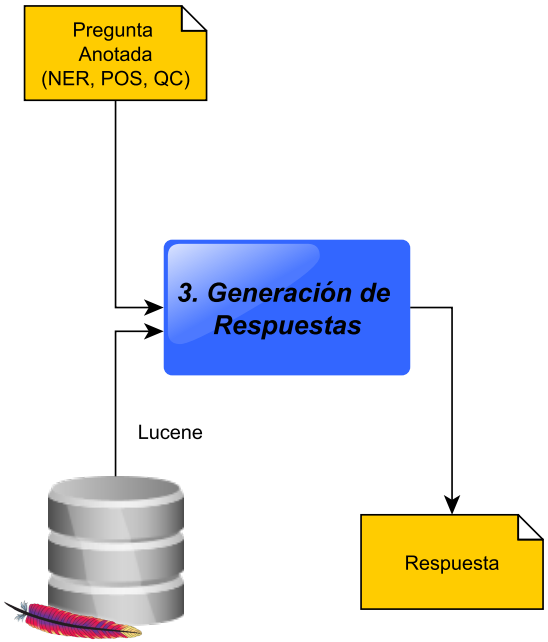
3. Generación de Respuestas



El algoritmo tiene los siguientes pasos:

1. Generación de Queries
2. Information Retrieval (documentos)
3. Extracción y rankeo de oraciones
4. POS tagging de las mejores oraciones
5. Heurísticas de AR basadas en el QC
   - Detectar respuestas candidatas
   - Re rankearlas según lógica específica
   - Extraer respuesta

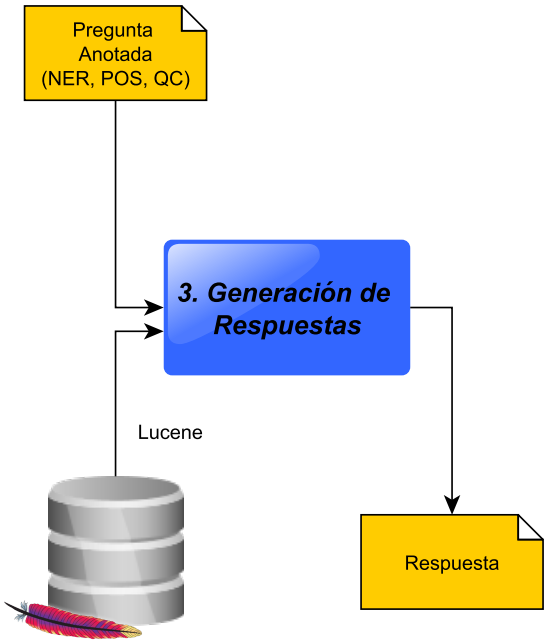
3. Generación de Respuestas



El algoritmo tiene los siguientes pasos:

1. Generación de Queries
2. Information Retrieval (documentos)
3. Extracción y rankeo de oraciones
4. POS tagging de las mejores oraciones
5. Heurísticas de AR basadas en el QC
   - Detectar respuestas candidatas
   - Re rankearlas según lógica específica
   - Extraer respuesta

3. Generación de Respuestas



El algoritmo tiene los siguientes pasos:

1. Generación de Queries
2. Information Retrieval (documentos)
3. Extracción y rankeo de oraciones
4. POS tagging de las mejores oraciones
5. Heurísticas de AR basadas en el QC
   - Detectar respuestas candidatas
   - Re rankearlas según lógica específica
   - Extraer respuesta

3. Generación de Respuestas (2)

Generación de Queries

- Caso baseline
  - Input: pregunta + 'target'(TREC) / tópico (CLEF)
    - Palabras no stop-words de *target*
    - Sustantivos, verbos y adjetivos de la pregunta.
- Evaluamos 3 métodos para generar queries
- Evaluamos 4 métodos para generar target

Extracción y re rankeo de oraciones post IR

- Documentos divididos en oraciones
- Re rankeo de oraciones con *scorers*:
- $rank = 0{,}9 \times covr + 0{,}05 \times freq + 0{,}05 \times prox$;
- Primeras 40 oraciones pos-taggeadas, el resto se desecha
- Evaluamos 6 métodos para rankear

Scorers: covr, freq, prox

cov & freq: Ocurrencia de la query en la respuesta candidata

- Cubrimiento: $\frac{\text{tokens de la query que ocurren en la oración}}{\text{tokens de la query}}$
- Frecuencia: $\frac{\text{tokens de la query que ocurren en la oración}}{\text{tokens de la oración}}$

Prox: distancia de dos strings en un tercero

- s_1 =“Argentina es un país americano” y s_2 =“independizado en 1810”
- t =“Argentina es un país americano, originalmente una colonia española, independizado en 1810”
- $prox(s_1, s_2, t) = \frac{dist(s_1,s_2)}{len(t)} = \frac{7}{12} = 0{,}58$,
  - $dist(s_1, s_2)$ = cantidad de strings entre ‘un’ y ‘en’ (tokens intermedios de $s_{1,2}$)
  - $len(t)$ = cantidad de strings del texto
  - 1 denota que los dos string están cercanos uno al otro en el tercer string.

3.6 Heurísticas de AR

3.6 Heurísticas de AR

- Basadas en Question Type
- Re ranking
- Extracción especifica

Casos

- ABBR:exp
- ABBR:abb $\rightarrow \emptyset$
- HUM:gr y ENTY:cremat
- HUM:ind
- HUM general
- LOC general
- NUM
  - date
  - period
  - count
  - general
- ENTY general.

Casos - ABBR:exp

Caso ABBR:exp - expansión de una abreviación

- Se extraen las entidades nombradas de tipo *Organización* de la pregunta.
- Se generan regex de expansión de esas abreviaciones.
  - UBA $\rightarrow [U][A-Za-z0-9][B][A-Za-z0-9][A][A-Za-z0-9]$.
- Busca este patrón en las 40 oraciones, devuelve el primer match.

Casos - HUM:gr y ENTY:cremat

Caso HUM:gr y ENTY:cremat - grupos humanos (compañías, organizaciones) y objetos humanos (libros, inventos, discos, etc)

- Respuestas candidatas → Nombres propios (NNPs) de todas las 40 oraciones (extracción).
  - Nombres propios consecutivos se agrupan (Tiger Woods)/NNP
- Ranking de las candidatas según scores:

$$TotalScore = \begin{cases} 0{,}55 & \text{Coverage: cubrimiento del target en la oración de soporte} \\ 0{,}20 & \text{Proximity: distancia entre el target y la respuesta candidata en la oración} \\ 0{,}10 & \text{Sentence: posición de la oración en las 40 seleccionadas} \\ 0{,}15 & \text{Repeated: penalización (-) target aparece en respuesta final.} \end{cases}$$

Más casos

Caso HUM:ind - individuo humano

- Respuestas candidatas = Extraer NER tipo *PERSON* de las oraciones.
- Rankear con *scorers* ponderados para el caso

HUM general y ENTY general: - otros temas humanos y otros objetos - Primer nombre propio de la oración mejor rankeada y se utiliza eso como respuesta.

LOC general: - preguntas que refieren a lugares - Se extraen todas las entidades nombradas de tipo 'LOCATION' de las oraciones rankeadas y se las evalúa según un score ponderado

Casos NUM - fechas, períodos, cuentas y números en general -

- La primer fecha encontrada según rank de pasajes (NUM:Date)
- El primer número encontrado según pasajes (NUM:Count)
- El primer numérico (número o fecha) para los otros casos NUM

Experimentos

Corridas

- Generación de Queries: baseline (1), improved baseline (2), lasso (3)
- Inferencia de Temas: Test (1), NERs (2), sustantivos (3), híbrido (4)
- Ranking de Pasajes: baseline (1), 2, 3 + nuevos *a posterior*
- Optimos español + portugués (2 opciones)

Evaluación de resultados

- Cantidad de respuestas dadas por el sistema: 1, 5, 10, 25 (4 opciones)
- Forma de evaluación automática: exacto, cubrimiento del 100 %, 75 % y 50 %, 15 % y 1 % (6 opciones)

Inferencia del tópico del grupo de preguntas

- Grupos de 1 a 4 preguntas con tema
- Ejemplos: Colegio de Harry Potter, Pez Espada, Revolución de Terciopelo
- Condiciones del tema
  - Nombrado en primer pregunta/respuesta
  - Las siguientes preguntas pueden contener correferencias a este tópico
- Incorporado al sistema como "Target"
- Probamos diferentes valores:
  1. Test (el del test set)
  2. NERs + Numeros + Fechas
  3. Sustantivos
  4. 2 y 3
- 2, 3 y 4 basados solo en la 1er pregunta del cluster.

Generación de queries

1. Baseline / Qanus:
   - Tokens no repetidos ni stopwords de target (tópico)
   - Sustantivos, verbos y adjetivos de la pregunta completa
   - + números + fechas
2. Baseline "mejorado": *(TITLE: target)*^{n} *OR* ***query baseline***, $n = 5$
3. Lasso: como describimos en Estado de Arte
   1. Palabras no stop words entre comillas
   2. NERs
   3. Construcciones nominales con sus adjetivos
   4. Demás construcciones nominales
   5. Sustantivos con sus adjetivos
   6. Demás sustantivos
   7. Verbos
   8. ~~El focus de la pregunta~~

Generación de respuestas

- Scorers nuevos para rankear los pasajes
- Scorers:
  - length : prioriza oraciones cortas pero no demasiado cortas.
  - qverb: presencia de verbos de la pregunta en la oración.
  - qnoun: presencia de sustantivos.
  - qner: presencia de NERs.
- Tres métodos de rankeo de pasajes:
  $pscore_{bl} = 0{,}9 \times cov + 0{,}05 \times freq + 0{,}05 \times prox$
  $pscore_2 = 0{,}4 \times pscore_{bl} + 0{,}2 \times qner + 0{,}15 \times qverb + 0{,}25 \times qnoun$
  $pscore_3 = 0{,}4 \times pscore_{bl} + 0{,}15 \times length + 0{,}1 \times qner + 0{,}1 \times qverb + 0{,}1 \times qnoun$

Corrida 2.1: Generación de Queries

- Idioma: Español
- Documentos retornados por Lucene: **50** (corrida 1 vs 70 y 100)
- Pasajes extraídos de los documentos: **40** (corrida 1 vs 20 y 60)
- **Generación de Queries: Baseline, Baseline mejorado, Lasso**
- Inferencia de Temas: NERs propio (2)
- Ranking de Pasajes: Baseline (1)

Corrida 2.1: Resultados

	Baseline		Baseline mejorado		Lasso	
Medida	Exacto	Covr 1	Exacto	Covr 1	Exacto	Covr 1
Exacto	**7.75**	**7.75**	**3.91**	**3.91**	**7.81**	**7.81**
MRR_5	**8.94**	**9.25**	**5.36**	**5.91**	**9.27**	**9.58**
MRR_{10}	9.31	9.38	5.96	6.43	9.64	9.71
MRR_{25}	9.54	9.61	5.96	6.43	9.94	10.01

Cuadro: Corrida 2.1: Generación de Queries

Observaciones y Conclusiones

- Método Lasso ligeramente superior a Baseline
- Baseline "mejorado" rezagado siempre

Corrida 2.2: Inferencia de Temas

- Idioma: Español
- Documentos retornados por Lucene: 50
- Pasajes extraídos de los documentos: 40
- Generación de Queries: **Baseline**
- **Inferencia de Temas: Test (1), NERs (2), sustantivos (3), híbrido (4)**
- Ranking de Pasajes: Baseline (1)

Corrida 2.2: Resultados

	1) Test		2) NERs	
Medida	Exacto	Covr 1	Exacto	Covr 1
Exacto	6.15	6.15	**7.75**	**7.75**
MRR_5	7.87	8.41	**8.94**	**9.25**
MRR_{10}	8.08	8.49	9.31	9.38
MRR_{25}	8.29	8.72	9.54	9.61

	3) Sustantivos		4) 2+3	
Medida	Exacto	Covr 1	Exacto	Covr 1
Exacto	3.85	3.85	5.47	5.47
MRR_5	4.19	4.19	7.12	7.43
MRR_{10}	4.19	4.30	7.36	7.43
MRR_{25}	4.47	4.58	7.64	7.71

Cuadro: Corrida 2.2: Métodos 1, 2, 3 y 4 respectivamente

Conclusiones:

- El método utilizado (NERs + números de la primer pregunta como target) es el más efectivo en todos los casos.

Corrida 2.3: Ranking de Pasajes

- Idioma: Español
- Documentos retornados por Lucene: 50
- Pasajes extraídos de los documentos: 40
- Generación de Queries: Baseline
- Inferencia de Temas: **NERs (2)**
- **Ranking de Pasajes: Baseline (1),** $pscore_2$**,** $pscore_3$

Corrida 2.3: Resultados

	$PassageScore_{bl}$		$PassageScore_2$		$PassageScore_3$	
Medida	Exacto	Covr 1	Exacto	Covr 1	Exacto	Covr 1
Exacto	7.75	7.75	3.10	3.10	5.38	5.38
MRR_5	8.94	9.25	5.06	5.22	6.69	6.85
MRR_{10}	9.31	9.38	5.35	5.50	7.15	7.31
MRR_{25}	9.54	9.61	5.39	5.55	7.19	7.35

Cuadro: Corrida 2.3: Fórmulas 1, 2 y 3 respectivamente

Observaciones y conclusiones

- Todo mal
- diff $pscore_2$ y $pscore_3 \rightarrow LengthScore$
  - Único agregado sustantivo a $pscore_3$
  - Aplicarlo al baseline
- Nuevas corridas con LenghtScore: 90/10 80/20 y 70/30
- La fórmula del score es la siguiente:

$$LengthScore = \begin{cases} 1{,}0 & 4 < \#tokens < 100 \\ 0{,}5 & 100 \leq \#tokens < 200 \\ 0{,}0 & \text{En cualquier otro caso} \end{cases}$$

Corrida 2.3: Resultados 2

	0.9 + 0.1		0.8 + 0.2		0.7 + 0.3	
Medida	Exacto	Covr 1	Exacto	Covr 1	Exacto	Covr 1
Exacto	7.81	7.81	**10.08**	**10.08**	10.00	10.00
MRR_5	9.47	9.78	11.68	11.99	11.94	12.50
MRR_{10}	9.71	9.78	11.92	11.99	12.35	12.67
MRR_{25}	10.01	10.08	12.27	12.34	12.58	12.91

Cuadro: Corrida 2.3: Combinación ponderada de métodos 1 y 3

Observaciones y conclusiones

- El score funciona como filtro de pasajes mal formados o bien formados pero largos. En ambos casos, NLP pierde eficacia.
- Mejor resultado cuando más ponderacion salvo MRR_1 que se redujo en la tercer corrida.
  - Dado que esta métrica es la preferida, decidimos quedarnos con esta fórmula (8/2 y no 7/3)
- Mejoras fáciles acá.

Corrida 3: Combinación de Óptimos

Medida	Exacto	Covr 1	Covr .75	Covr .5	Covr .15	Covr .01
Exacto	**10.65**	10.65	10.77	12.31	13.08	13.08
MRR_5	**12.00**	12.31	13.46	16.18	16.95	17.01
MRR_{10}	12.24	12.31	13.46	16.26	17.44	17.50
MRR_{25}	12.55	12.62	13.78	16.72	18.00	18.06

Cuadro: Corrida 3.1: Combinación de óptimos para español

Medida	Exacto	Covr 1	Covr .75	Covr .5	Covr .15	Covr .01
Exacto	**1.92**	2.88	2.88	3.85	3.85	3.85
MRR_5	**3.16**	5.03	5.03	6.91	7.15	7.15
MRR_{10}	3.45	5.33	5.33	7.33	7.71	7.86
MRR_{25}	4.08	6.00	6.00	8.21	8.84	8.94

Cuadro: Corrida 3.2: Combinación de óptimos sobre portugués

Resultados de competidores de CLEF '07

Run	% Overall Accuracy [200]
Priberam	44,5
Inaoe	34,5
Miracle	15
UPV	11,5
TALP	7

Figura: Competidores español

Run Name	Overall Accuracy (%)
diue071ptpt	40,9%
esfi071ptpt	7,4%
esfi072ptpt	4,0%
feup071ptpt	22,8%
ines071ptpt	11,4%
ines072ptpt	14,1%
prib071ptpt	61,7%

Figura: Competidores portugués

Corrida 3: Respuestas ES

Q	A
¿De dónde es típica la ensaimada?	Mallorca
¿De cuántos bits era este procesador?	8 bits
¿En qué fecha El Corte Inglés compró Galerías Preciados?	24 de noviembre de 1995
¿Qué empresa preside Steve Jobs?	Apple Computer
¿Dé qué organización es comisionado David Stern?	NBA
¿Qué animales tiran del carro de Cibeles?	leones
¿Cuántos Premios Goya ganó "Torrente: El brazo tonto de la ley"?	dos
¿Quién fue la primera mujer que viajó al espacio?	Valentina Vladimírovna Tereshkova
¿A cuántos kilómetros de Huesca está el Castillo de Loarre?	35 kilómetros
¿Cuál es el nombre del presidente de Burundi que murió en 1994?	Cyprien Ntaryamira
¿Qué día se promulgó la constitución española de 1812?	19 de marzo
¿Qué dibujó Leonardo Da Vinci en 1492?	Hombre de Vitruvio
¿Qué programa de televisión presentó Takeshi Kitano de 1986 a 1989?	Humor Amarillo

Cuadro: Respuestas español - 10,65 % = $\frac{13}{122}$

Corrida 3: Respuestas PT y observaciones

Q	A
De que estado foi governador Adhemar de Barros?	São Paulo
Quando foi fundado o Nacional da Madeira?	8 de Dezembro de 1910

Cuadro: Respuestas portugués - 1,92 % = $\frac{2}{104}$

Observaciones

- Español peor que en 2.3 para muchas métricas
  - Español: 10.65 % exacto con una sola respuesta!
  - No es el peor
- Portugués: baja performance
  - Apenas un 3.16 % considerando la métrica MRR_5.
  - Las herramientas para portugués no son tan maduras como las herramientas para español;
  - Todo el desarrollo y el análisis estuvo basado en español (POC soporte otro idioma)

Conclusiones: Dominio abierto

Sobre modelo de dominio abierto:

- Impresión: falta de una "doctrina" formal teórica accesible sobre generación de respuestas. ¿Oportunidad?
- Una mejora argumentada y con cierta presencia en la literatura del área fue el mecanismo de generación de queries que llamamos Lasso (por el sistema en el que fue implementado por primera vez) y nosotros lo incorporamos aquí,
  - mejoras mínimas en la performance, de 7.75 a 7.81 en MRR_1 exacto (+0.26), de 8.94 a 9.27 en MRR_5 exacto (+0.31).

Mejoras

- Módulo de *focus* open source en inglés. Problema acotado, bien definido e inexistente. Herramienta útil.
- Módulo multilingüe de Question Classification.
- Mejorar las heurísticas

Cierre

Dominio cerrado:

- Sistema baseline para QA como interfaz para una base de datos
- Modelo de tratabilidad semántica

Dominio abierto:

- Sistema baseline para dominio abierto multilingüe.
- QANUS + Freeling + Lasso + Scorers
- Preservamos la performance original en español, con 10.65 % de precisión exacta y 12,00 % para MRR_5.

Ambos acá: `github.com/julian3833`

¿Preguntas?