# Question Answering Over Linked Data en Español

**Robertino Nicolazzi**

UNC, FaMAF

Noviembre, 2017

# Tarea

Este proyecto se basa en la tarea propuesta en QALD-2017 Task 1 en el cuál se presentan preguntas con distintos niveles de dificultad y que precisan diferentes técnicas para procesar su respuesta.
Se analizan 4 tipos de preguntas:

- Date
- Resource
- Number
- Boolean

En las cuales puede ser requerido establecer ordenes en las respuestas o contabilizar las mismas

# Tarea

A su vez cada pregunta presenta las respuestas correctas y la query que genera dichos resultados escrita en SPARQL

**Tipo:** Resource
**Pregunta:** ¿Quién es la esposa de Barack Obama?
**Keywords:** Barack Obama, esposa
**Query:**

**SPARQL Query**

```
PREFIX dbo: http://dbpedia.org/ontology/
PREFIX dbr: http://dbpedia.org/resource/
SELECT DISTINCT ?result WHERE {
dbr:Barack_Obama dbo:spouse ?result.
}
```

# Arquitectura

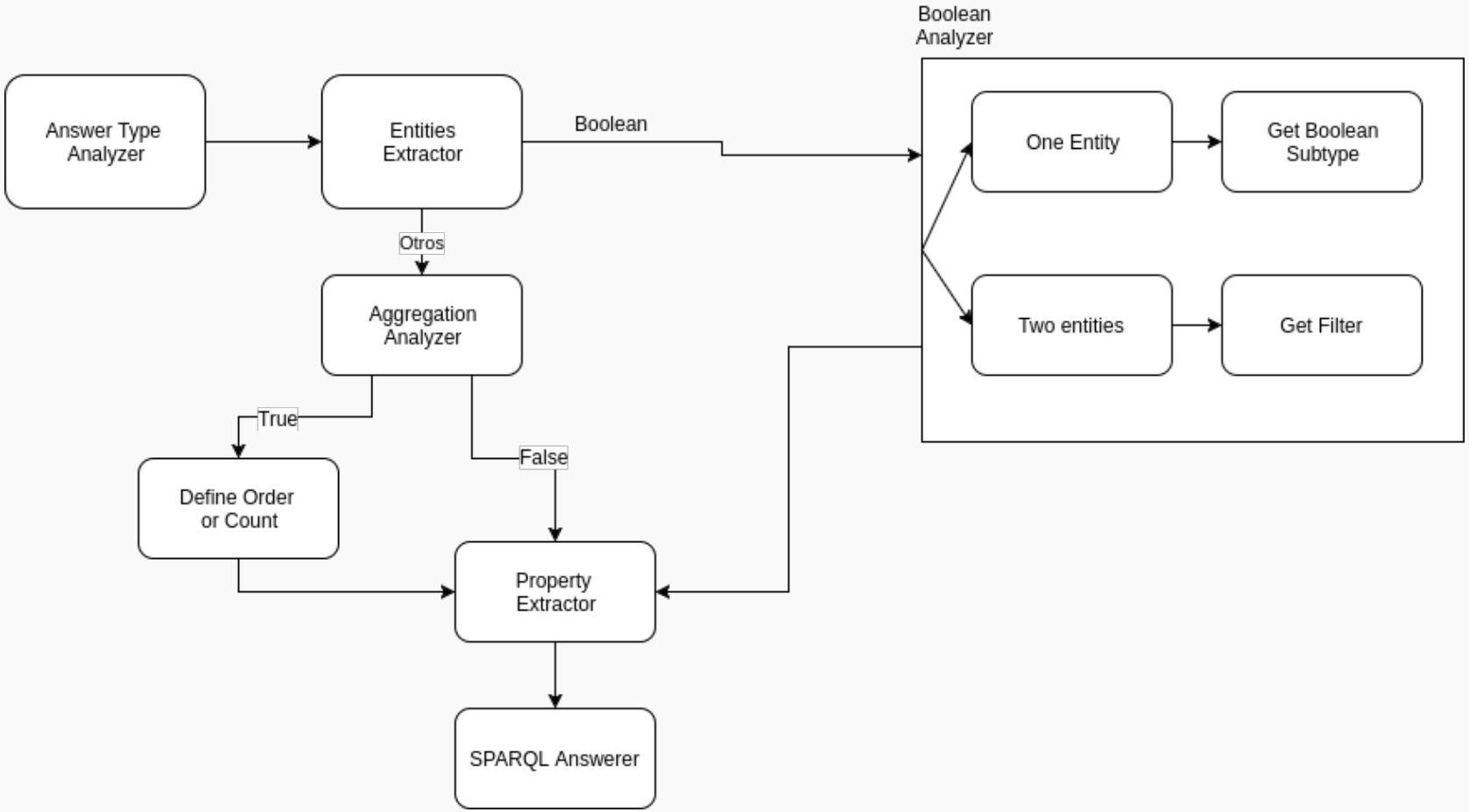


**Figura:** Arquitectura

# Simple Questions

Primero detectamos el tipo de pregunta mediante una serie de features para luego procesarla acorde a su tipo.

Luego mediante el Entity Extractor definimos la entidad (o entidades) principal de la pregunta utilizando los Nouns Groups de la misma estableciendo una prioridad.

Por último utilizando el Property Extractor y los keywords asociados a la pregunta, obtener la propiedad deseada analizando el mapeo keywords - dbo property

# Answer Type Extractor

Analizando las preguntas del Corpus de entrenamiento, se extrajeron diferentes keywords y patrones que ocurren en los distintos tipos de preguntas. A partir de los cuales se diseñaron features viendo la presencia en cierta ubicación o no de cada patrón para entrenar un NaiveBayes Classifier
Por ejemplo

¿Cuándo tuvo lugar la batalla de Gettysburg? $\rightarrow$ *DATE*
{ 'init_Cuando': True, 'init_Cuanto':False,'init_Verb':False ... }

¿Cuánto costo Pulp Fiction? $\rightarrow$ *NUMBER*
{ 'init_Cuando': False, 'init_Cuanto':True,'init_Verb':False ... }

# Entity Extractor

El proceso en la selección de entidades utilizados es el siguiente:

- Se obtienen todos los Noun Group
- Se verifica que el Noun Group sea una entidad en DBPEDIA Español
  - Si se encuentra, utilizando sameAs, se obtiene su referencia en DBPEDIA Ingles
  - Sino se intenta encontrar la entidad utilizando la DBPEDIA Ingles directamente
- Entre las entidades encontradas se elige 1 ( o 2 en Booleanas) mediante cierto orden de prioridad
  - Presencia de Sustantivos Propios
  - Tamaño del Grupo

# Entity Extractor

**Pregunta:** ¿Quién es la esposa de Barack Obama?
**Keywords:** Barack Obama, esposa
**Noun Groups:** esposa, Barack Obama, Obama, Barack
**Selecciones:** Barack Obama, Obama, Barack
**Entidad:** Barack Obama

**Pregunta:** ¿Tienen el príncipe Harry y el prínciple William los mismos progenitores?
**Keywords:** príncipe Harry, progenitores, principe William
**Noun Groups:** principe, principe Harry, Harry, progenitores, principe, principe William, William
**Selecciones:** principe Harry, principe William, Harry, William
**Entidades:** principe Harry, principe William

# Property Extractor

Utilizando DictVectorizer y el clasificador LogisticRegresion se arma un pipeline. Por cada pregunta de entrenamiento, utilizando sus keys asociados

- Remover nombres propios
- Remover StopWords
- Mantener verbos, sustantivos y adjetivos

Features
Por cada palabra en español se obtienen los sinónimos para generar mayor cantidad de datos y features por pregunta.

- Se utiliza una Bag of Words de las palabras de los keys
- Bag of Words de los sinónimos.
- Se chequea igualdad entre traducciones ingles-español y español-ingles
- Igualdad traducción-sinónimo

# Property Extractor

**Caso Exitoso**

'es': 'esposa',
'en': 'spouse',
**'es=entoes': True,**
'en=estoen': False,
'entoes': 'esposa',
'estoen': 'wife',
'mujer=entoes': False,
...
'esposa=entoes': True,
'conyugue=entoes': False,
'costilla=entoes': False,
'senora=entoes': False,
'senora': True,
'pareja': True,
'consorte': True,
'mujer': True,
'esposa': True,
'spouse': True,
'conyugue': True

**Caso de Falla**

'en': 'birth place',
'es': 'esposa',
'entoes': 'lugar de nacimiento',
'estoen': 'wife'
'es=entoes': False,
'en=estoen': False,
'costilla=entoes': False,
...
'esposa=entoes': False,
'companera=entoes': False,
'mujer=entoes': False,
'birth': True,
'place': True,
'esposa': True,
'conyugue': True,
'pareja': True,
'mujer': True,
'consorte': True,
'senora': True,

# Boolean Questions

- 1 Entidad
  Se detectan 3 tipos de preguntas. Las cuales se identifican utilizando reglas y/o patrones
  - Si Entidad posee tal Propiedad
  - Si existe una Entidad
  - Si Tal Entidad es de cierto Tipo
- 2 Entidades
  En este caso se analiza para detectar el operador de comparación a utilizar entre la propiedad que involucra las dos Entidades.
  Utilizando reglas que verifican la existencia de tales palabras como "menor", "mayor", se define el Filtro.

# Ejemplos Booleanos

Ejemplos:
- ¿Tienen el príncipe Harry y el prínciple William los mismos padres?
- ¿Fué la Crisis de Cuba antes de la Invasión de Bahía de Cochinos?
- ¿Tiene Breaking Bad más episodios/episodios que Game of Thrones?

# Aggregation Questions

- Preguntas de cantidad.
  Se busca la presencia de palabras como ¸cuantoscuantas"
- Preguntas de Orden
  Se busca presencia de palabras que indican orden "mas grande", último", "mayor"

Posibles preguntas

- Subpropiedad de entidad.
- Orden de entidades
- Orden de entidades en determinada localización

# Ejemplos Aggregation

Ejemplos:
- ¿Cuántos hijos tuvo Benjamin Franklin?
- ¿Cuál es el hijo mas mayor de Meryl Streep?
- ¿Que libro tiene el mayor numero de paginas?

# Resultados

Para analizar el desempeño del sistema se utliza la Accuracy del mismo. Es decir, la cantidad de preguntas que responde correctamente.
El corpus de evaluación contiene preguntas simple, preguntas booleanas y preguntas que necesitan orden o contabilizar resultados.
Se utilizaron sinónimos de palabras conocidas para el sistema y propiedades de DBPEDIA conocidas también.
El sistema funciona relativamente bien en preguntas simples adivinando la propiedad requerida con tan solo una pregunta de entrenamiento por cada propiedad

# Resultados

**Logistic Regression**
**HITS:** 50
**TOTAL:** 63
**Accurancy:** 0.79

**Random Forest**
**HITS:** 43
**TOTAL:** 63
**Accurancy:** 0.68

**DecisionTree**
**HITS:** 29
**TOTAL:** 63
**Accurancy:** 0.46

**Logistic Regression CV**
**HITS:** 37
**TOTAL:** 63
**Accurancy:** 0.58

# Conclusión y Trabajo Futuro

Mejoras a realizar

- Utilización de Machine Learning para determinar la necesidad de utilizar funciones de Aggregation en un pregunta
- Generar las reglas booleanas mediante algún clasificador

Este proyecto es un primer paso para entender el funcionamiento de un sistema para responder preguntas en Español. Mediante análisis se podría concluir que la dificultad principal de un idioma como el español radica en que, a diferencia del Ingles, las preguntas no tienen una estructura definida, lo cual dificulta realizar un sistema que sea general.
Por ende, un desarrollo futuro seria necesario para poder resolver preguntas mas complejas que involucren mas propiedades o relaciones entre entidades.