Trabajo de minería de texto: Detección de especies de pájaros

Jaime García Lozano

16 de mayo de 2022

1. Introducción

Disponemos de distintas publicaciones extraídas de tres blogs relacionados con la observación de pájaros ([1], [2] y [3]). Cada una de ellas ha sido guardada en un fichero txt junto a su fecha de publicación y el url de su correspondiente blog. En el siguiente enlace se puede acceder a un repositorio que contiene el código en el *notebook Trabajo_pajaros*, los archivos txt utilizados y los resultados del trabajo en forma de ontología (*birds_ontology*): [4].

Tenemos dos objetivos principales:

1. **Encontrar los pájaros**: Será necesario desarrollar un programa capaz de detectar la mayoría de especies mencionadas en cada publicación. A su vez, cada resultado ha de ser asociado con su enlace de la *Dbpedia*.
2. **Crear una ontología**: Una vez completado el paso anterior, se organizarán los resultados en una ontología donde cada especie de pájaro será un individual que tendrá asociado su enlace de la *Dbpedia* y la publicación (o publicaciones) donde ha sido mencionado.

2. Extracción de las especies mencionadas en la Dbpedia mediante Sparql

La idea del proceso que vamos a desarrollar es utilizar una función de detección de *strings* similares. Con ella vamos a comparar los tokens de cada publicación con el conjunto de especies de pájaro mencionadas en la base de datos *dbo* de la *Dbpedia*, la cual se puede conseguir mediante el lenguaje de consultas de *Sparql* (véase sección *Extraer las especies de Dbpedia* en el *notebook*).

Finalmente guardamos los resultados de la consulta en un diccionario (llamado *species*) con los nombres de las especies (*labels*) como llaves y los enlaces de la *Dbpedia* como valores.

```
{ [...],
 'trichastoma abbotti':["http://dbpedia.org/resource/Abbott's_babbler"],
 'turdinus abbotti':["http://dbpedia.org/resource/Abbott's_babbler"],
 "abbott's booby": ["http://dbpedia.org/resource/Abbott's_booby"],
 'sula abbotti': ["http://dbpedia.org/resource/Abbott's_booby"],
 'abd al-kuri sparrow': ['http://dbpedia.org/resource/Abd_al-Kuri_sparrow'],
 "abdim's stork": ["http://dbpedia.org/resource/Abdim's_stork"],
 'aberdare cisticola': ['http://dbpedia.org/resource/Aberdare_cisticola'],
 'aberrant bush warbler': ['http://dbpedia.org/resource/Aberrant_bush_warbler'],
 ' cettia flavolivacea': ['http://dbpedia.org/resource/Aberrant_bush_warbler'],
 [...]}
```

3. Encontrar los pájaros

La función *get_close_matches* tiene como *inputs* una palabra de interés y una lista de strings de donde queremos extraer la palabra más similar, lo cual nos obliga a tokenizar el texto. Esto supone un reto en el caso en el que el *string* de interés esté formado por dos o

más palabras. Por ejemplo, si tuviéramos una especie en la Dbpedia llamada *Red-winged Cormorant* y en una publicación estuviera escrito 'Red Cormorant', al tokenizar el texto nos quedaría como dos palabras separadas. Al aplicar la función no nos podría hacer el *match* que buscamos. Podríamos tokenizar también la especie y buscar por una parte el *string* más similar de *Red-winged* y por otra el de *Cormorant*, pero no estaríamos teniendo en cuenta el orden, lo que afectaría muy negativamente a los resultados.

3.1. Primera opción: Filtrar entidades con Spacy (finalmente descartado)

La librería *Spacy* permite detectar distintos tipos de entidades dentro de un texto (fechas, ciudades y países, organizaciones, etc.). Lamentablemente, no dispone de una etiqueta destinada a especies de animales. Aún así, en general, detecta correctamente los nombres propios. Como en la mayoría de publicaciones los pájaros están escritos en mayúsculas, era de esperar que entre las entidades extraídas estuvieran sus nombres. Al final lo que se tiene es una lista de entidades para cada publicación (véase función *entities*).

Para cada publicación se recorre la lista de entidades buscando el *string* más similar de cada una de ellas dentro de la lista de llaves del diccionario *species*. Si se encuentra algún *match*, lo guardamos en otro diccionario.

Se trata de un proceso realmente eficiente, pero los resultados son bastante pobres. Para algunas publicaciones no se encuentra ninguna especie y para otras son directamente erróneos.

En algunos casos (publicación 6) sí que se consigue información interesante:

```
----------------------------------------------------------------------
{'COLLARED INCA': ['collared inca',
                   'http://dbpedia.org/resource/Collared_inca'],
'HUMBOLDT PENGUIN': ['humboldt penguin',
                     'http://dbpedia.org/resource/Humboldt_penguin'],
'LADDERWINGED NIGHTJAR': ['ladder-tailed nightjar',
                          'http://dbpedia.org/resource/Ladder-tailed_nightjar'],
'Puna Plover': ['puna plover', 'http://dbpedia.org/resource/Puna_plover'],
'REDLEGGED CORMORANT': ['reed cormorant',
                        'http://dbpedia.org/resource/Reed_cormorant'],
'hummingbird': ['hummingbird', 'http://dbpedia.org/resource/Hummingbird'],
'the BLUEHEADED PARROTS': ['blue-headed parrot',
                           'http://dbpedia.org/resource/Blue-headed_parrot']}
----------------------------------------------------------------------
```

(Nota: las llaves del diccionario son las entidades y los valores una lista con la etiqueta de la especie en la *Dbpedia* y su enlace)

El problema de este proceso está en el punto de partida: filtrar las entidades nos hace perder demasiada información relevante. Se ha de pensar un método en el que se pueda utilizar el texto en crudo.

3.2. Segunda opción: Utilizar el texto en crudo

Desarrollamos una función (*birds*) que va a tener el siguiente proceso (véase el *notebook* para más detalles):

Para cada publicación:

1. Se tokeniza el texto.
2. Iniciamos un bucle que recorrerá todas las especies mencionadas en la *Dbpedia*
3. Se mira el número de palabras que tiene la etiqueta de la especie en cuestión.
4. Si está formada por una única palabra, se busca el *match* entre la etiqueta y la lista de tokens de la publicación. Si se detecta alguno, se guarda en un diccionario junto al enlace de la *Dbpedia*.
5. Si está formada por dos o más palabras:
   a) Iniciamos un bucle infinito
   b) Agrupamos los *strings* de la lista de tokens de n en n, siendo n el número de palabras de la etiqueta (Nota: se ha supuesto que, como máximo, la etiqueta estará compuesta por cuatro palabras).
   c) Buscamos el *match* entre la etiqueta y la nueva lista de tokens. Si se encuentra, se guarda y se detiene el bucle.
   d) Si no se encuentra ningún *match*, volvemos al paso *(b)* y desplazamos la agrupación una posición. Es decir, si *n=2*, y antes se concatenaban los tokens de la siguiente manera: 1-2, 3-4, 5-6,...; siendo 1,2,... la posición de cada *string* en la lista de tokens original, ahora lo hacen como 2-3, 4-5, 6-7,... Si ya se han hecho todas las agrupaciones posibles (que coinciden con el número de palabras de la etiqueta) se detiene el bucle.

El lado negativo que tiene este programa es que es computacionalmente costoso, ya que para cada publicación se han de recorrer todas las etiquetas de la Dbpedia. Sin embargo mejora sustancialmente los resultados del anterior apartado. Por ejemplo, para la primera publicación, en la que antes no detectábamos nada:

```
----------------------------------------------------------------------
{'greenwinged teal': ['green-winged teal',
                      'http://dbpedia.org/resource/Green-winged_teal']}

----------------------------------------------------------------------
```

Y en la publicación 6, donde antes habíamos extraido menos de diez especies:

```
----------------------------------------------------------------------
{'a peruvian divingpetrel': ['peruvian diving petrel',
                             'http://dbpedia.org/resource/Peruvian_diving_petrel'],
 'andean avocets': ['andean avocet',
                    'http://dbpedia.org/resource/Andean_avocet'],
 'andean cockoftherock': ['guianan cock-of-the-rock',
                          'http://dbpedia.org/resource/Guianan_cock-of-the-rock'],
 'avocets': ['avocet', 'http://dbpedia.org/resource/Avocet'],
 'bluefooted boobies': ['blue-footed booby',
                        'http://dbpedia.org/resource/Blue-footed_booby'],
 'blueheaded parrots': ['blue-headed parrot',
```

```
                      'http://dbpedia.org/resource/Blue-headed_parrot'],
'chestnutcrested cotinga': ['chestnut-crested cotinga',
                            'http://dbpedia.org/resource/Chestnut-crested_cotinga'],
'cinclodes': ['cinclodes', 'http://dbpedia.org/resource/Cinclodes'],
'cinclodes a': ['cinclodes', 'http://dbpedia.org/resource/Cinclodes'],
'cockoftherock': ['cock-of-the-rock',
                  'http://dbpedia.org/resource/Cock-of-the-rock'],
'collared inca': ['collared inca',
                  'http://dbpedia.org/resource/Collared_inca'],
'dacnis': ['dacnis', 'http://dbpedia.org/resource/Dacnis'],
'eagle': ['eagle', 'http://dbpedia.org/resource/Eagle'],
'elaenia': ['elaenia', 'http://dbpedia.org/resource/Elaenia'],
'green honeycreeper': ['green honeycreeper',
                       'http://dbpedia.org/resource/Green_honeycreeper'],
'gulls': ['gull', 'http://dbpedia.org/resource/Gull'],
'hoatzin': ['hoatzin', 'http://dbpedia.org/resource/Hoatzin'],
'honeycreeper': ['honeycreeper', 'http://dbpedia.org/resource/Honeycreeper'],
'hooded mountain tanagers': ['hooded mountain tanager',
                             'http://dbpedia.org/resource/Hooded_mountain_tanager'],
'humboldt penguin': ['humboldt penguin',
                     'http://dbpedia.org/resource/Humboldt_penguin'],
'hummingbird': ['hummingbird', 'http://dbpedia.org/resource/Hummingbird'],
'inca terns': ['inca tern', 'http://dbpedia.org/resource/Inca_tern'],
'lyretailed nightjar': ['silky-tailed nightjar',
                        'http://dbpedia.org/resource/Silky-tailed_nightjar'],
'macaws': ['macaw', 'http://dbpedia.org/resource/Macaw'],
'nighthawk': ['nighthawk', 'http://dbpedia.org/resource/Nighthawk'],
'owl': ['owl', 'http://dbpedia.org/resource/Owl'],
'parrot': ['parrot', 'http://dbpedia.org/resource/Parrot'],
'pelicans': ['pelican', 'http://dbpedia.org/resource/Pelican'],
'peruvian thickknees': ['peruvian thick-knee',
                        'http://dbpedia.org/resource/Peruvian_thick-knee'],
'puna plover': ['puna plover', 'http://dbpedia.org/resource/Puna_plover'],
'purple sandpiper': ['purple sandpiper',
                     'http://dbpedia.org/resource/Purple_sandpiper'],
'rednecked phalaropes': ['red-necked phalarope',
                         'http://dbpedia.org/resource/Red-necked_phalarope'],
'sandcolored nighthawks': ['sand-coloured nighthawk',
                           'http://dbpedia.org/resource/Sand-coloured_nighthawk'],
'seabird': ['seabird', 'http://dbpedia.org/resource/Seabird'],
'shearwaters': ['shearwater', 'http://dbpedia.org/resource/Shearwater'],
'shining sunbeam': ['shining sunbeam',
                    'http://dbpedia.org/resource/Shining_sunbeam'],
'solitary eagle': ['solitary eagle',
                   'http://dbpedia.org/resource/Solitary_eagle'],
'sooty shearwaters': ['sooty shearwater',
                      'http://dbpedia.org/resource/Sooty_shearwater'],
'species': ['species:', 'http://dbpedia.org/resource/Ludiortyx'],
'terns': ['tern', 'http://dbpedia.org/resource/Tern'],
'trogons': ['trogon', 'http://dbpedia.org/resource/Trogon'],
'whitethroated toucan': ['white-throated toucan',
```

```
                          'http://dbpedia.org/resource/White-throated_toucan'],
 'wirecrested thorntail': ['wire-crested thorntail',
                           'http://dbpedia.org/resource/Wire-crested_thorntail'],
 'yellowbellied dacnis': ['yellow-bellied dacnis',
                          'http://dbpedia.org/resource/Yellow-bellied_dacnis'],
 'yellowbrowed sparrows': ['yellow-browed sparrow',
                           'http://dbpedia.org/resource/Yellow-browed_sparrow']}

-----------------------------------------------------------------------
```

(Nota: las llaves del diccionario son las entidades y los valores una lista con la etiqueta de la especie en la *Dbpedia* y su enlace)

4. Ontología

En este apartado se han organizado los resultados en una ontología que posteriormente podrá ser abierta en *Protegé*.

La estructura será muy básica:

- Cada pájaro será una individual asociado por la propiedad *IS_MENTIONED_BY* con su correspondiente publicación y por *DBPEDIA_URL* con su enlace de la Dbpedia.
- Cada publicación tendrá asociados todos los pájaros que menciona mediante la propiedad *MENTIONS*, el enlace del blog de donde se extrajo mediante *PUBLICATION_URL* y su fecha a partir de *PUBLICATION_DATE*.

Veamos cómo queda en *Protegé*:

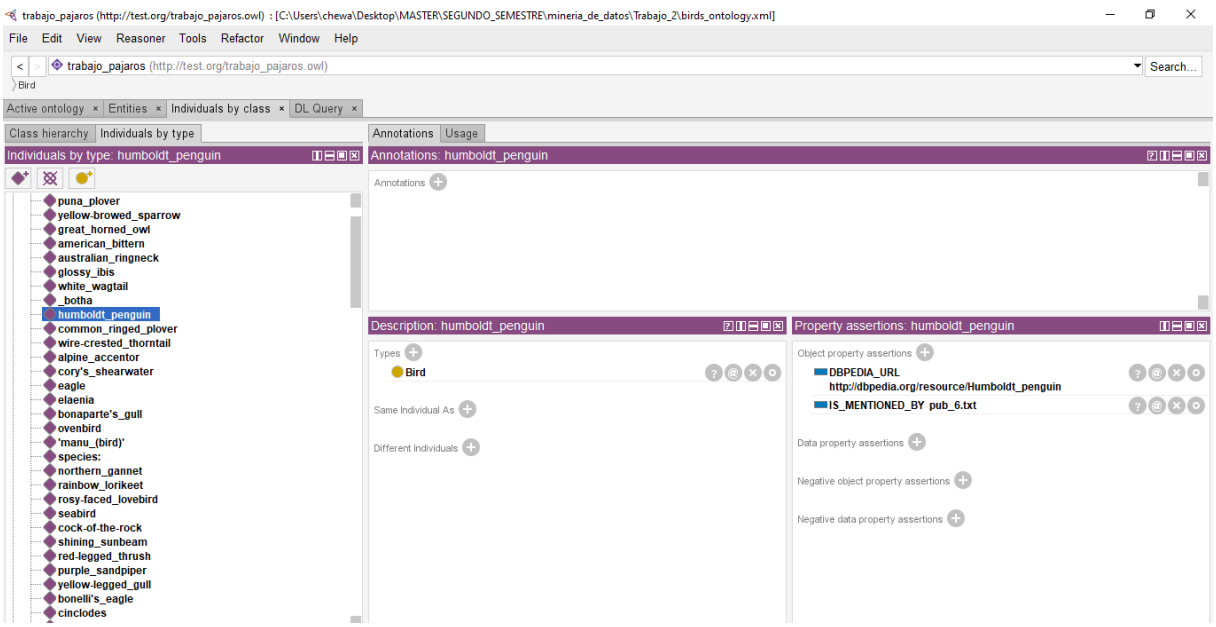

Figura 1: Ontología en *Protegé*. Véase que el pájaro seleccionado tiene como propiedades su enlace de la *Dbpedia* y las publicaciones donde se menciona (*pub_6* en este caso).

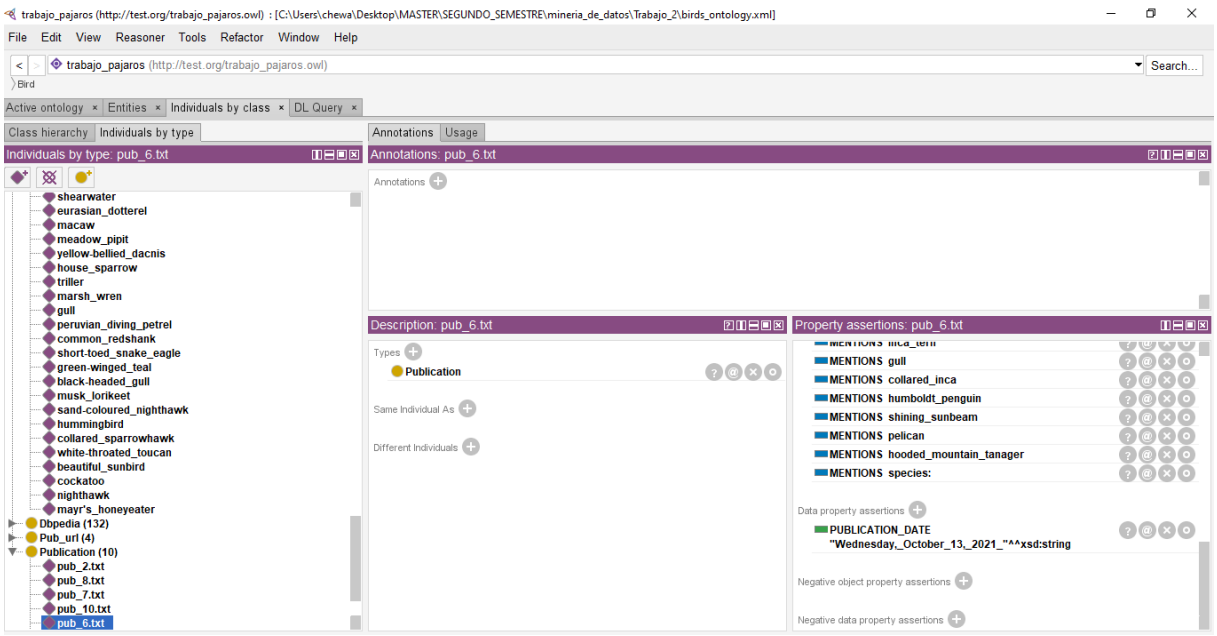

Figura 2: Ontología en *Protegé*. Véase que la publicación seleccionada tiene asociados los pájaros que menciona, el enlace del blog (no se ve en la imagen) y la fecha de publicación.

5. Conclusión

Se ha conseguido desarrollar un programa capaz de detectar especies de pájaro mencionadas en un texto. Su mayor debilidad es el alto coste computacional que podría ser reducida mediante algunas mejoras. Se plantean dos opciones:

- Recorrer los tokens del texto en vez de las etiquetas de la Dbpedia, contemplando las distintas agrupaciones que se han explicado anteriormente. Con esto se conseguiría reducir sustancialmente el número de iteraciones del bucle
- Utilizar expresiones regulares en vez de la función *get_close_matches*. Recorrer todas las etiquetas de la Dbpedia aplicando una expresión regular que busque cada una de ellas dentro de la publicación en cuestión. Con esto conseguimos ahorrarnos el proceso de tokenizar el texto y aumentar la eficiencia enormemente.

Por otra parte, se ha visto el potencial de organizar la información en una ontología. Aunque la que se ha creado sea muy simple, nos abre las puertas a añadir información relevante:

- Pájaros en peligro de extinción
- Lugar o lugares donde vive cada especie.
- Organizar en especies y subespecies

Toda esta información serviría para hacer inferencias o extraer información de interés del tipo "Se ha detectado un ejemplar de la especie X (en peligro de extinción), en un lugar Y , en torno a la fecha Z".

Referencias

[1] *Shorebirder*, https://www.shorebirder.com/.

[2] *Trevorsbirding*, https://www.trevorsbirding.com/.

[3] *Dantallmansbirdblog*, https://dantallmansbirdblog.blogspot.com/.

[4] *Repositorio*, https://github.com/JGL98/Detector_de_pajaros.