

# Información no estructurada

## Práctica 4 – Relevance feedback

Javier Aróstegui Martín

### Contenido

Implementación: .....	2
Fórmulas utilizadas:.....	2
Primer Caso: .....	2
Generación del nuevo ranking: .....	2
Resultados y evaluación: .....	3
Segundo Caso: .....	3
Generación del nuevo ranking: .....	4
Resultados y evaluación: .....	4

## Implementación:

La implementación se ha realizado en Python con la ayuda de las librerías de pandas para el manejo de los datos y de numpy para operaciones matemáticas. El fichero que nos devuelve las palabras de la nueva consulta es RelevanceFeedback.py. Además se hace uso de los ficheros vectorial.py (para calcular el nuevo ranking) y metrics.py (para el cálculo de las métricas).

## Fórmulas utilizadas:

$$\vec{q}_m = \alpha \vec{q} + \beta \sum_{\vec{d}_j \in D_r} \vec{d}_j - \gamma \sum_{\vec{d}_j \in D_n} \vec{d}_j$$

Con parámetros  $\alpha = \beta = \gamma = 1$ .

$$BIR(w) = \log \frac{(|r_w| + 0.5)(|D| - |D_w| - |r| + |r_w| + 0.5)}{(|D_w| - |r_w| + 0.5)(|r| - |r_w| + 0.5)}$$

Con parámetro  $|D| = 1000$ .

$$rsv(w) = |r_w| \times BIR(w)$$

## Primer Caso:

El primer caso ha consistido en coger el ranking devuelto por el modelo vectorial en la P3 con la consulta original y compararlo con el ranking generado por el mismo modelo, pero con la consulta modificada.

## Generación del nuevo ranking:

Para generar la nueva consulta se ha calculado  $q'$  como se indica en la sección de [fórmulas utilizadas](#), luego rsv para elegir los 4 (valor arbitrario) términos con mayor relevancia. Para que los términos salgan con sentido he puesto que la colección total de documentos  $|D|$  sea igual a 1000. La nueva consulta escoge el valor  $q'$  de esos 4 términos y generamos una nueva consulta y con ella el nuevo ranking.

## Resultados y evaluación:

### Ranking original:

```
documento,score,relevancia,posicion
d15,1.277657,1,1
d20,1.122590,0,2
d3,1.030306,1,3
d16,0.854546,1,4
d14,0.836464,0,5
d11,0.822602,0,6
d7,0.758935,0,7
d8,0.702156,0,8
d19,0.573546,1,9
d13,0.513747,0,10
d2,0.459423,1,11
d6,0.439368,0,12
d4,0.433969,1,13
d12,0.429221,0,14
d5,0.379693,0,15
d1,0.326487,1,16
d18,0.298897,0,17
d9,0.275907,0,18
d17,0.241560,1,19
d10,0.238879,0,20
```

### Ranking con la nueva consulta:

```
documento,score,relevancia,posicion
d20,1.412630,0,1
d3,1.320634,1,2
d15,0.978663,1,3
d19,0.936838,1,4
d16,0.735506,1,5
d11,0.645033,0,6
d4,0.576161,1,7
d2,0.519193,1,8
d7,0.423367,0,9
d1,0.414368,1,10
d17,0.365711,1,11
d8,0.356299,0,12
d12,0.308290,0,13
d13,0.269528,0,14
d18,0.193274,0,15
d5,0.075630,0,16
d9,0.016829,0,17
d10,0.005735,0,18
d6,0.005720,0,19
d14,0.000000,0,20
```

Una vez tenemos los 2 rankings, podemos analizarlos y evaluarlos. Lo primero que llama la atención del nuevo ranking es que el primer documento no es relevante, mientras que en el ranking original si lo era. Sin embargo, todos los documentos relevantes se encuentran en las primeras 11 posiciones mientras que en el ranking original están más dispersos. Para evaluar el rendimiento se utilizan  $P@3$ ,  $P@5$ ,  $P@10$  y  $nDCG$ .

### Métricas originales:

```
Precision@3: 0.6666666666
Precision@5: 0.6
Precision@10: 0.4
nDCG: 0.821893753305707
```

### Métricas con la nueva consulta:

```
Precision@3: 0.6666666666
Precision@5: 0.8
Precision@10: 0.7
nDCG: 0.8006307096148536
```

En las métricas podemos ver como en términos de precisión la nueva consulta obtiene mejores valores de  $P@5$  y  $P@10$  ya que ha conseguido aglutinar casi todos los documentos relevantes al principio. Sin embargo, obtiene un  $nDCG$  ligeramente inferior. Esto puede ser debido a que el primer documento del ranking (el más importante para la métrica  $nDCG$ ) en el nuevo ranking no es relevante y por tanto pierde mucha puntuación. Modificando algunos parámetros del cálculo de  $q'$  o el *cutoff* de los términos podríamos llegar a mejorar la consulta en todos los aspectos.

## Segundo Caso:

Este segundo caso se ha realizado haciendo uso del ranking que nos proporciona Google. Para ello he evaluado el ranking original de la consulta respecto a una nueva consulta que contiene los 4 primeros términos más importantes 1 sola vez. He ignorado el valor de  $q'$  para cada uno de estos términos, simplemente he incluido los top 4 términos más importantes.

### Generación del nuevo ranking:

La nueva consulta se ha calculado de la misma manera que en el primer caso. Una vez se dispone de los 4 términos más importantes he ejecutado la consulta en Google y he revisado los 20 primeros documentos devueltos para ver su relevancia.

### Resultados y evaluación:

#### Métricas originales:

```
Precision@3: 0.6666666666666666  
Precision@5: 0.4  
Precision@10: 0.3  
nDCG: 0.8047773711885111
```

#### Métricas con la nueva consulta:

```
Precision@3: 0.3333333333333333  
Precision@5: 0.6  
Precision@10: 0.4  
nDCG: 0.6339730860414478
```

Los resultados en este caso son relativamente parecidos, aunque con un peor rendimiento. La precisión para tops bajos empeora ya que al principio no hay muchos documentos relevantes. Sin embargo, en total en los 20 documentos hay más relevantes que para la consulta original y por ello la P@10 por ejemplo ya proporciona un mejor resultado. También podemos observar que nDCG es claramente inferior, esto es debido a que las posiciones donde se encuentran los documentos relevantes son muy altas.