Búsqueda y recuperación de información. Master Data Science

Curso 2016/2017

People Name Disambiguation

Alejandro Pérez Barreiro

1. Objetivo

El objetivo de la práctica será la desambiguación del nombre de Thomas Baker utilizando las herramientas aprendidas en la asignatura como son el procesamiento de texto con la librería NLTK de Python. Para ello se dispone de los 19 primeros textos que arrojo una búsqueda sobre este nombre, y, analizando y procesando estos textos se debe conseguir asignar a cada uno de ellos la persona sobre la que habla. Entre todos los textos hay cuatro personas diferentes con el nombre de Thomas Baker

La bondad del resultado de nuestras trasformaciones se mirará con el *rand_score* que arroja el script, estando este valor entre 1 y -1 siendo 1 el mejor resultado. El valor inicial con los textos sin procesar es de -0.15, veamos cómo cambia. Primero mostraré la combinación de transformaciones que me dieron mejor resultado y al final los cambios que no aportaron mejoras al *rand_score.*

Nota: Todas las transformaciones que se muestran se van a realizar sobre los tokens obtenido en el script de partida.

2. Signos de puntuación/Palabras vacías

En primer lugar voy ver cómo cambia el resultado eliminando las stopwords y los signos de puntuación.

```
stop = set(stopwords.words('english'))
filter_tokens = []
for token in tokens:
    if token not in string.punctuation and stop:
        filter_tokens.append(token)
```

El resultado obtenido es de -0.11 que supone una mejora pero no muy importante, esto es normal pues difícilmente un signo de puntuación puede ayudar a clasificar un texto, puesto que nunca va ligado a un contenido.
Se observa que sin filtrar las palabras vacías el resultado es el mismo por tanto no sería necesario hacerlo.

3. Palabras no alfabéticas

Tras eliminar los signos de puntuación el siguiente paso fue quitar los tokens que no tuviesen ninguna letra como son las fechas, espacios... En un principio pensaba que esta transformación empeoraría el resultado porque se puede pensar que cada persona distinta tendrá asociadas fechas como son su nacimiento y acontecimientos de su vida y que estas fechas pueden ayudar a distinguir entre una persona y otra pero el resultado fue el contrario, al eliminar los tokens no alfabéticos el resultado mejoró significativamente dejándolo en -0.08.

```
filtered_tokens = []
for token in filter_tokens:
    # Quito las palabras que no contengan letras
    if re.search('[a-zA-Z]', token):
        filtered_tokens.append(token)
```

4. Entidades nombradas

Una vez hecha la limpieza que considero necesaria procedo a reconocer las entidades nombradas con el NER de Stanford de 7 clases y me quedo solo con las palabras que son reconocidas como tal. Esta transformación la hago porque creo que al fin y al cabo el contenido de un texto se puede deducir viendo únicamente sus entidades nombradas.

```
ENs = []
EN_words = st.tag(filtered_tokens)
for word in EN_words:
    if word[1] != 'O':
        ENs.append(word[0])
```

El *rand_score* tras este procesamiento es 0.06

5. Agrupación en trigramas

Por ultimo lo que hice con las palabras que eran entidades nombradas fue agruparlas en trigramas para así intentar darle un sentido a cada token agrupándolo con la anterior EN y la siguiente en el texto. Además también eliminé los trigramas repetidos en un mismo documento para que a la hora de crear los vectores solo tuviese en cuenta "el vocabulario" del texto y que no desvirtuara la medida TF el hecho de que ese trigrama apareciese más de una vez en el mismo texto.

A la hora de agrupar probé en bigramas, trigramas, cuatrigramas… hasta agrupar los tokens de seis en seis, obteniéndose con los trigramas el mejor resultado.

```
ngramas = []
n = 3
grams = ngrams(ENs,n)
for gram in grams:
    ngramas.append(gram)

ngramas = list(set(ngramas))
```

Este último procesamiento sumado a los anteriores dio un valor en el *rand_score* de 0.8, valor que se puede considerar alto.

6. Otras transformaciones

Los pasos explicados anteriormente y en ese orden fueron los que arrojaron un mejor resultado pero también se probaron otras técnicas vistas en clase como:

- Lematización y steming: estos procesos no empeoraban el resultado obtenido pero tampoco lo mejoraban. Esto probablemente se deba a que como he elegido quedarme solo con las entidades nombradas no tiene mucho sentido hacer stemming ni lemmatizacion sobre ellas ya que la mayoría quedarian igual que estaban.