Guide 6 Introduction to text mining in R

September 30, 2019

1 Guía 6: Introducción a la minería de datos textuales en R

Computación 2, IES. Profesor: Eduardo Jorquera, eduardo.jorquera@postgrado.uv.cl

2 Datos ordenados

Recuerdas los principios de los datos ordenados en R? Te refrescaré la memoria, un conjunto de datos se dice que está ordenado si: * Las observaciones están en filas * Las variables están en columnas * Está contenido en un solo dataset

Esto último, dice que cada tipo de unidad observacional es una tabla.

Dado esto, podemos definir un formato de texto ordenado como una tabla con "un *token* por fila".

2.1 Pregunta: £Qué es un token? £Qué es "tokenizar"?

```
In [ ]:
```

La estructura de un token por fila va es opuesto con la forma en que el texto es almacenado en los análisis actualmente.

El paquete dplyr (que ya hemos visto con anterioridad), entre otros, permite la manipulación de datos ordenados.

3 Contrastando el texto ordenado con otras estructuras de datos

Como vimos anteriormente, nos avocaremos al formato de texto de un token por fila. Estructurar los datos textuales de esta manera, quiere decir que se conformará a los principios de los datos ordenados, y que podrán ser manipulados con un conjunto de herramientas consistentes. Vale la pena contrastar las diferentes formas que hay de almacenar texto en minería de datos textuales: * String: El texto puede (obviamente) ser guardado como strings, cadenas de caracters. En programas como R, el texto siempre se lee primero en este formato. Aquí entran los vectores con cadenas de caracteres, strings n-dimensionales, etc. * Corpus: Este tipo de objeto típicamente contiene las filas de texto (strings) anotadas con metadata adicional y más detalles. * Matriz documento-término (term-document matrix): Ésta es una matriz sparse que describe una colección (es decir, un corpus) de documentos con una fila por cada documento y una columna por cada término. El valor del elemento i-j-ésimo de la matriz corresponde a la frecuencia de la palabra j-ésimo en el i-ésimo documento, u otra medida de frecuencia.

4 La función `unnest_tokens`

Aquí un párrafo de Antonio Machado:

```
In [3]: texto <- c("Este donquijotesco",
                   "don Miguel de Unamuno, fuerte vasco,",
                   "lleva el arnés grotesco",
                   "y el irrisorio casco",
                   "del buen manchego. Don Miguel camina,",
                   "jinete de quimérica montura,",
                   "metiendo espuela de oro a su locura,",
                   "sin miedo de la lengua que malsina.")
        texto
```

1. 'Este donquijotesco' 2. 'don Miguel de Unamuno, fuerte vasco,' 3. 'lleva el arnés grotesco' 4. 'y el irrisorio casco' 5. 'del buen manchego. Don Miguel camina,' 6. 'jinete de quimérica montura,' 7. 'metiendo espuela de oro a su locura,' 8. 'sin miedo de la lengua que malsina.'

Éste es un tícipo vector de caracteres que podemos analizar. Con el objeto de ponerlo en forma de datos ordenados, primero debemos ponerlo en un dataframe.

```
In [6]: library(dplyr)
        text_df <- tibble(line = 1:length(texto), text = texto)

        text_df
```

A tibble: 8 Œ 2

line <int>	text <chr>
1	Este donquijotesco
2	don Miguel de Unamuno, fuerte vasco,
3	lleva el arnés grotesco
4	y el irrisorio casco
5	del buen manchego. Don Miguel camina,
6	jinete de quimérica montura,
7	metiendo espuela de oro a su locura,
8	sin miedo de la lengua que malsina.

4.1 Pregunta: £Qué quiere decir que el dataframe sea "tibble"?

```
In [ ]:
```

Nota que el dataframe que contiene el texto aún no es compatible con el concepto de *datos ordenados*. Podemos filtrar palabras o contar las que ocurren con mayor frecuencia, ya que cada fila está hecha con múltiples palabras combinadas. Necesitamos convertirlas para que queden como *un token por documento por palabra*.

Dentro del trabajo de texto ordenado, necesitamos *tokenizar* el texto y transformarlo en datos ordenados. Para esto, usamos la función `unnest_tokens()`:

```
In [9]: library(tidytext)

        text_df %>%
          unnest_tokens(word, text)
```

line <int>	word <chr>
1	este
1	donquijotesco
2	don
2	miguel
2	de
2	unamuno
2	fuerte
2	vasco
3	lleva
3	el
3	arnés
3	grotesco
4	y
4	el
4	irrisorio
4	casco
5	del
5	buen
5	manchego
5	don
5	miguel
5	camina
6	jinete
6	de
6	quimérica
6	montura
7	metiendo
7	espuela
7	de
7	oro
7	a
7	su
7	locura
8	sin
8	miedo
8	de
8	la
8	lengua
8	que
8	malsina

A tibble: 40 Œ 2

Aquí los argumentos básicos para usar esta función, son nombres de columnas. Primero nombramos a la columna que tenrá el output de la función, y luego a la columna que contiene el texto en el dataframe en donde están los datos.

Luego de usar `unnest_tokens`, separaremos cada fila de tal forma que sea un token (término) cada una; por defecto, la tokenización se hace para cada palabra. También nota lo siguiente: * Otras columnas, tales como el número de la línea de dónde viene cada palabra, es retenido. * La