Tercera parte de la entrega

2024-11-29

Objetivos de la Tercera Parte

En esta parte del proyecto, se nos pide implementar un programa llamado `train_analysis.R` que realice el análisis del corpus obtenido en la primera parte, almacenado en el archivo `spanish_train.qcorpus.rds`. El programa debe calcular, para cada documento, lo siguiente:

1. **Frecuencia de verbos del título**: Contar cuántos verbos, convertidos a su forma en infinitivo, presentes en el campo `title` del documento también aparecen en el texto completo correspondiente.
2. **Frecuencia de verbos del resumen**: Contar cuántos verbos, convertidos a su forma en infinitivo, presentes en el campo `summary` del documento también aparecen en el texto completo correspondiente.

Estos resultados se presentan de manera visual mediante dos histogramas: - Un histograma para los verbos del `title`. - Otro histograma para los verbos del `summary`.

En los histogramas obtenidos, se muestra la distribución de las frecuencias de coincidencia y el número de documentos que corresponden a cada frecuencia. Esto permite visualizar de forma clara cómo se relacionan los verbos presentes en los `title` y `summary` con el contenido completo de los textos.

Importación de Código para Tests

Para mantener el documento más organizado y legible, importamos el código de tests desde otro archivo. Esto nos permite ejecutar y mostrar los resultados de las pruebas específicas de las funciones utilizadas en el programa principal sin añadir bloques de código extensos.

La importación se realiza con la función `source()` de R, que permite cargar y ejecutar el contenido de un archivo externo. En este caso, importamos el archivo `test_train_analysis.R`, donde se encuentran definidas las pruebas.

Uso de Archivos RDS para Optimizar el Procesamiento

Dado que el parseo de textos con la función `spacy_parse()` puede ser un proceso lento y costoso, decidimos implementar una estructura `if-else` para gestionar el almacenamiento y la carga de los datos procesados. La idea principal es evitar realizar el parseo en cada ejecución, almacenando los resultados en archivos `.RDS`, que son compactos y rápidos de cargar.

Estructura General

La estructura que utilizamos funciona de la siguiente manera:

- **Si el archivo RDS ya existe**:
  - Cargamos directamente los datos procesados con la función `readRDS()`.
  - Esto reduce significativamente el tiempo de ejecución en futuras sesiones.
- **Si el archivo RDS no existe**:
  - Parseamos los datos correspondientes con `spacy_parse()` y los guardamos en un archivo `.RDS` utilizando `saveRDS()`.
  - Esto asegura que el procesamiento solo se realiza la primera vez.

Función `spacy_parse()`

1. **Tokenización**:
   - Divide el texto en **tokens** (en nuestro caso las palabras de los textos).
2. **Etiquetado gramatical (POS tagging)**:
   - Asocia cada token con su categoría gramatical (por ejemplo, `VERB`, `NOUN`, `ADJ`).
3. **Lematización**:
   - Convierte las palabras a su forma raíz (por ejemplo, "corriendo" → "correr").

El propósito principal de usar `spacy_parse()` en este proyecto es que nos permita trabajar específicamente con los **verbos** en infinitivo de los textos. Esto incluye:

- Identificar todos los verbos en un texto (`pos == "VERB"`).
- Obtener los verbos en su forma **lema** (`lemma`), para evitar inconsistencias debido a conjugaciones.

Resultado de `spacy_parse()`

Tras realizar el `spacy_parse()` sobre las palabras de los campos `title`, `summary` y los textos completos, obtenemos una tabla estructurada que contiene información clave sobre los tokens del texto. En nuestro caso, nos quedamos con las columnas `lemma` (forma en infinitivo de los verbos) y `doc_id` (identificador único de cada documento).

El uso del `doc_id` es bastante importante, ya que nos permite asociar cada verbo con su documento correspondiente.

Ejemplo de los Resultados

Verbos del campo `title`:

```
   doc_id     lemma
2   text1     decir
4   text1  peligrir
28  text4 reconocer
40  text5  inspirar
57  text6     parar
70  text7     tener
```

Verbos del campo `summary`:

```
   doc_id    lemma
13  text1 advertir
17  text1 invertir
26  text1    tener
28  text1  detener
48  text2   lanzar
52  text2 disparar
```

Verbos de los textos completos:

```
   doc_id    lemma
3   text1  recibir
15  text1 terminar
32  text1    decir
47  text1 recortar
50  text1   llegar
61  text1    hacer
```

Función `verbs_create_list`

La función `verbs_create_list` organiza los verbos extraídos del corpus en una estructura más manejable, agrupándolos por documento. Su objetivo principal es asociar cada verbo a su documento correspondiente, utilizando el identificador único de cada documento (`doc_id`).

La función toma los verbos extraídos de un campo del corpus, como `title` o `summary`, y los organiza en una lista, donde cada elemento corresponde a un documento específico. Si un documento no contiene verbos en un campo determinado, se incluye en la lista con un vector vacío, asegurando que todos los documentos estén representados de manera consistente.

El resultado de esta función es una lista con nombres que corresponden a los `doc_id` del corpus y cuyo contenido son los verbos en su forma lematizada. Esta estructura facilita el acceso y análisis de los datos, permitiéndonos comparar los verbos presentes en los títulos o resúmenes con los del texto completo de cada documento.

Resultados tras aplicar la función sobre los dataframes obtenidos anteriormente:

Aplicando para `verbs_titles`:

```
$text1
[1] "decir"    "peligrir"

$text2
character(0)

$text3
character(0)

$text4
[1] "reconocer"
```

```
$text5
[1] "inspirar"

$text6
[1] "parar"
```

Para `verbs_summary`:

```
$text1
[1] "advertir" "invertir" "tener"    "detener"

$text2
[1] "lanzar"    "disparar"  "dispersar"

$text3
character(0)
```

Lo mismo con `verbs_text`:

```
$text1
[1] "recibir"  "terminar" "decir"    "recortar" "llegar"   "hacer"    "juntar"
[8] "proteger"

$text2
 [1] "hindúes"   "gritar"    "tener"     "decir"     "producir"  "volver"
 [7] "empezar"   "lanzar él" "resultar"  "arrestar"  "evitar"    "liderarar"

$text3
[1] "ver"     "mostrar" "buscar"  "perder"  "ganar"   "crear"
```

Eliminación de Verbos Duplicados

Como indica el enunciado, necesitamos trabajar con los **verbos únicos** de los campos `title` y `summary`, ya que no queremos contar verbos repetidos dentro de un mismo documento. Para lograr esto, definimos la función `make_unique()`, que elimina duplicados en cada elemento de una lista. Su objetivo es garantizar que cada sublista contenga solo valores **únicos**.

La función recorre cada elemento de la lista (correspondiente a un documento) y, si este no está vacío, utiliza la función `unique()` para filtrar los valores duplicados. De esta forma, el resultado es una lista en la que cada sublista contiene únicamente los verbos únicos asociados a su documento.

Ejemplo de Uso

Si observamos el contenido original de `list_verbs_summary`, en el documento **text__5** aparece el verbo `tener` dos veces.

```
$text5
[1] "infectar" "tener"    "tener"    "ver"
```

Después de aplicar `make_unique()`, el resultado muestra este verbo solo una vez.

```
$text5
[1] "infectar" "tener"    "ver"
```

Cálculo de Frecuencias de Verbos

Llegamos al núcleo del proyecto: **contar cuántas veces los verbos presentes en los campos `title` y `summary` aparecen en el texto completo de cada documento**. Para ello, definimos la función `freq_verbs`, que calcula la frecuencia total de los verbos en el texto correspondiente.

Explicación de la Función `freq_verbs`

La función `freq_verbs` toma tres argumentos:

1. **`text_list`**: Lista de verbos en los textos completos.
2. **`list_comp`**: Lista de verbos a comparar (pueden ser de `title` o `summary`).
3. **`corpus_ids`**: Identificadores únicos de los documentos.

El objetivo de esta función es recorrer los textos y contar cuántas veces los verbos presentes en `list_comp` aparecen en el texto completo correspondiente. Para ello:

- Se inicializa un vector vacío llamado **`freq`**, cuya longitud corresponde al número de documentos, y se asignan los `corpus_ids` como nombres.
- Para cada documento, se recorren los verbos de `list_comp` y se cuenta cuántas veces aparecen en el texto completo (`text_list`) utilizando `sum()`.
- Finalmente, el vector `freq` almacena la frecuencia total de los verbos por documento y se devuelve como resultado.

Aplicamos esta función y obtenemos las frecuencias de los verbos de cada `title` en su texto y de cada `summary` en su texto.

Veamos las frecuencias de los `titles` 50 - 100:

```
freq_verbs_titles[50:100]
```

```
 text50  text51  text52  text53  text54  text55  text56  text57  text58  text59
     14       0       0       0       4       0       1       6       2      13
 text60  text61  text62  text63  text64  text65  text66  text67  text68  text69
      4       0       0       1       0       0       0       0       0       0
 text70  text71  text72  text73  text74  text75  text76  text77  text78  text79
      0       0       6       0      11       0       6       1       0       0
 text80  text81  text82  text83  text84  text85  text86  text87  text88  text89
      7       0       4       0      14       0       0       0       0       0
 text90  text91  text92  text93  text94  text95  text96  text97  text98  text99
      0       0       0       0       1       2       0       0       8       1
text100
      1
```

Veamos las frecuencias de los `summary` 50 - 100:

```
freq_verbs_summary[50:100]
```

```
 text50  text51  text52  text53  text54  text55  text56  text57  text58  text59
      0       0       0       0       8       0       0       5       2      19
 text60  text61  text62  text63  text64  text65  text66  text67  text68  text69
```

```
    11       9       6       6       1      10       2       4       0      12
 text70  text71  text72  text73  text74  text75  text76  text77  text78  text79
     0       0       0       1       7       2       1       0       3       0
 text80  text81  text82  text83  text84  text85  text86  text87  text88  text89
     0       3       2       9       0       0       0       0       0       0
 text90  text91  text92  text93  text94  text95  text96  text97  text98  text99
     0       0       6       2       8       2       0       0       2       0
text100
     3
```

Ahora viene la parte de los **testeos** para confirmar que lo que hemos hecho es correcto. Para realizarlo, vamos a crear otra función `test_freq_verbs`.

La función `test_freq_verbs` prueba la función `freq_verbs` en un **subconjunto** de documentos, calculando la frecuencia de verbos en títulos y resúmenes. Muestra los verbos coincidentes entre texto, título y resumen, y sus frecuencias correspondientes para cada documento.

Veamos el testeo:

```
test_freq_verbs(list_verbs_text, list_verbs_title_unique, list_verbs_summary_unique, corpus_ids)
```

—— Testing freq_verbs Function ——

Document ID: text56 Text Verbs: saber desafeír responder animar realizar seguir

Title Verbs: saber Matched Title Verbs: saber Matched Count (Titles): 1

Matched Verbs:

Matched Count (Summaries): 0

Document ID: text57 Text Verbs: desconocer dar recetar mandar recibir enfermar deber hacer esparcir aumentar temer interesar admitir fracasar detectar confirmar necesitar requerir decir pedir hacer solicitar monitorear detectar apoyar desarrollar designar dirigir combatir anunciar adelantar intentar calmar estabilizar desconocer intentar lidiar creer estar decir causar pasar saber comentar reportar considerar creer presentar monitorear llevar dar contar dar reflejar señalar desconocer mostrar indicar distribuir saber agregar coincidir tener detectar permitir aislar demorar considerar recordar comenzar decidir fabricarir examinar comenzar enviar encontrar funcionar tener cambiar él demorar comenzar hacer hacer señalar colocar ayudar prevenir enviar realizar contar realizar limitar contar hacer ultramar contar ofrecer enviar confirmar confirmar tener enviar confirmar esperar recordar obtener realizar limitar acceder creer comiencir extender él conducir opinar comentar terminar tener existir cumplir someter detectar indicar hacer especificar adoptar abordar anunciar contagiar realizar él corregir acudir hacer él prescribir mostrar presentar encontrar aceptar tener añadir detectar practicar contagiar presentar contagiar tener crear hacer hacer él hacer creer creer estar él afirmar convertir él contener tener manifestar transmitir hacer existir servir detener considerar desconocer someter detectar hacer dejar publicar rondar transcurrir reportar él publicar comenzar hacer ascender indicar tomar tener reportar realizar realizar analizar reportar evaluar realizar intentar contactar conocer realizar dejar ofrecer obtener detectar separar requeír tener llevar presentar requeír acudar tener desembolsar cobrar hacer desestimir ir encontrar definir cubrir presentar tener indicar indicar posponer dejar comprar responder preocupar llevar él vivir entrar limitar utilizar creer tomar contagiar acceder autoaislar él evitar opinar registrar requerir hacer proteger trabajar elegir hacer él autoaislar él indicar coincidir sugerir alentar quedar él trabajar tomar presentar tener tener hacer tener tener escuchar autoaislar él dejar trabajar tener hacer él quedarir mantener él comentar continuar trabajar experimentar agravar contagiar manejar ocurririr opinar pedir tomar garantizar indicar anunciar garantizar quedar él tomar tener proporcionar dirigir afirmar decir requerir contar hacer hacer pasar durar dar señalar visitar recibir descargar activa él perdertir

Title Verbs: fracasar combatir dejar Matched Title Verbs: fracasar, combatir, dejar Matched Count (Titles): 6

Matched Verbs: contagiar Matched Count (Summaries): 5

Document ID: text58 Text Verbs: estar participar participar esperar producir estar alcanzar él decir decidir viajar pasar tratar resolver interesar viajar asistir

Title Verbs: viajar Matched Title Verbs: viajar Matched Count (Titles): 2

Matched Verbs: participar Matched Count (Summaries): 2

Document ID: text59 Text Verbs: causar explorar encontrar decidir oponer él ordenar detener morir pasar gobernar dejar descubrir creer pertenecer interesar analizar descubrir asesinar tener conducir pertenecer desconocer desaparecer hacer extinguir surgir saltar contraer llevar ocurrir comenzar generar desaparecer tener representar hacer apresuran averiguar él desaparecer enterar recibir describir matar ocurrir servir despachar comer encontrar surgir infectar morir tener dominar significar evolucionar propagar respirar evitar preocupar causar infectar matar ocurrir desaparecer anunciar describir existir contraigar decir él aplicar creer escapar pasar tener llevar enfermar enfermar tener morir significar identificar tomar incubar él volver él dar encontrar transmitir actuar decir contraer identificar mostrar desarrollar infectar tratar trabajar aprender convivir viajar asistir registrar sufrir estar hacer empeorar dirigir pedir aislarar infectar querer llevar generar estimar rastrear él morir eliminar salir extinguir afectar decir tener ganar utilizar eliminar disminuir ver pasar afectar desaparecer convivir llegar extinguir él terminar descubrir causar intervenir existir soler ocurrir infectar estar descubrir saltar dar sugerir afectar declarar causar persistir comenzar limitar creer causar adquirir enfrentar tratar combatir dificultar hacer dejar circular seguir mostrar buscar preferir quedar él matar erradicar seguir circular significar llevar eliminar él creer llegar surgir infectar trasladar decir creer dar habitar considerar manjar señalar replegar él soler transmitir infectar contraer decir creer pertenecer pasar identificar decir convertir comenzar preguntar él propagar decir él dificultar llevar involucrar existir estar resultar desvanecir tomer estar infectar acompañar causar estar infectar causar preguntar propagar él pensar vivir evolucionar infectar yo mostrar resultar morir infectar existir deber existir hacer extinguir causar desaparecer provocar matar circular surgir tender seguir evolucionar extinguir evolucionar reemplazar él enfocas replicar decir desaparecer ver adaptar él parecer causar desaparecer acumular llegar extinguir él acelerar sugerir acelerar permitir yo utilizar existir sugerir combatir enfocar incluir incluir significar secuestrar copiar él incluir verificir considerar significar existir permitir evolucionar dirigir volver gustar pensar representar decir hacer erradicar reconocer infiltrar él detectar volver dar dificultar conducir jugar acelerar estimulir mutar traer debilitar reducir circular facilitar existir funcionar encontrar favipiravir demostrar parecer hacer acumular hacer él desaparecer suceder mutar sugerir dirigir él ir intenter mostrar decir ir decir resultar acechar acechar almacenar acechar decir extinguir señalar existir llevar él terminar preocupar tener desencadenar mencionar ensamblar saber extinguir recrear él utilizar solicitar significar pensar deberir centrar yo reducir esperar reflexionar querer erradicar decir saber saber inspirar contraer volver tener preocupar él leer recordar recibir descargar activa él perdertir

Title Verbs: desaparecer dejar combatir Matched Title Verbs: dejar, desaparecer, combatir Matched Count (Titles): 13

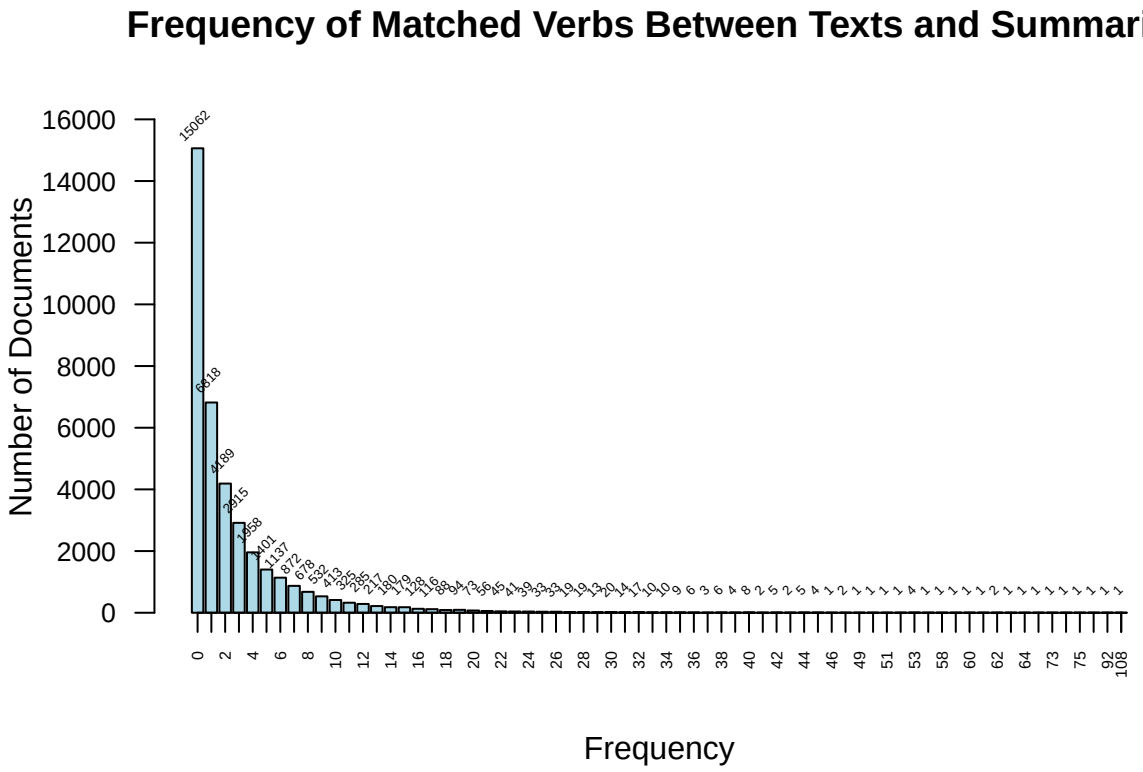
Matched Verbs: causar, desaparecer, persistir Matched Count (Summaries): 19

Document ID: text60 Text Verbs: jurar mandar insistir marcar tener definir él querer decir sorprender saber cultivar dejar él centrar interesar tocar construir encontrar reconocer detener proteger librar llegar establecer abrir tuitear temer interferir tornir llevar lograr opinar preferir pronunciar él -alegar acabar ceder detener mostrar tener saber hacer deshacer agregar creer unir decir rascar tener hacer querer tener meter coincidir esperar felicitar resolver mostrar ver venir mandar marcar activar venir prever firmar encontrar afectar exigir dejar poner solucionar construir querer afectar decir obligar asumir frenar imponer acabar convertir querer llegar aceptar pagar decir acabar firmar recibir contribuir ver acabar convertir esperar resolver convertir anunciar suspender admitir restablecer llevar aclarar permanecer estar esperar querer recibir alertar expertar consultorar exponer imponer decir querer hacer insistir esperar cumplir llevar tener acompañar él plantee ver emigrar declarar ver entrar ver tener devolver juzgar acabar exonerar considerar tener hacer probar él acusar fabricar destacar compartir tener recoger tener empezar enmendar afectar estallar aprobar limitar afectar generar generar opinar creer seguir dar causar llegar tener ir afirmar impulsar ver respetar valorar creer poner él empezar construir prestar complicar enmendar echar andar apagar quedar concluir recibir descargar activa él perdertir

Title Verbs: esperar Matched Title Verbs: esperar Matched Count (Titles): 4

Matched Verbs: tener, meter Matched Count (Summaries): 11

Gráficamente podemos **visualizar** esto. Primero, vamos a crear una **tabla** con las frecuencias de los resumenes y hacer un `bar_plot`. Quedando lo siguiente:

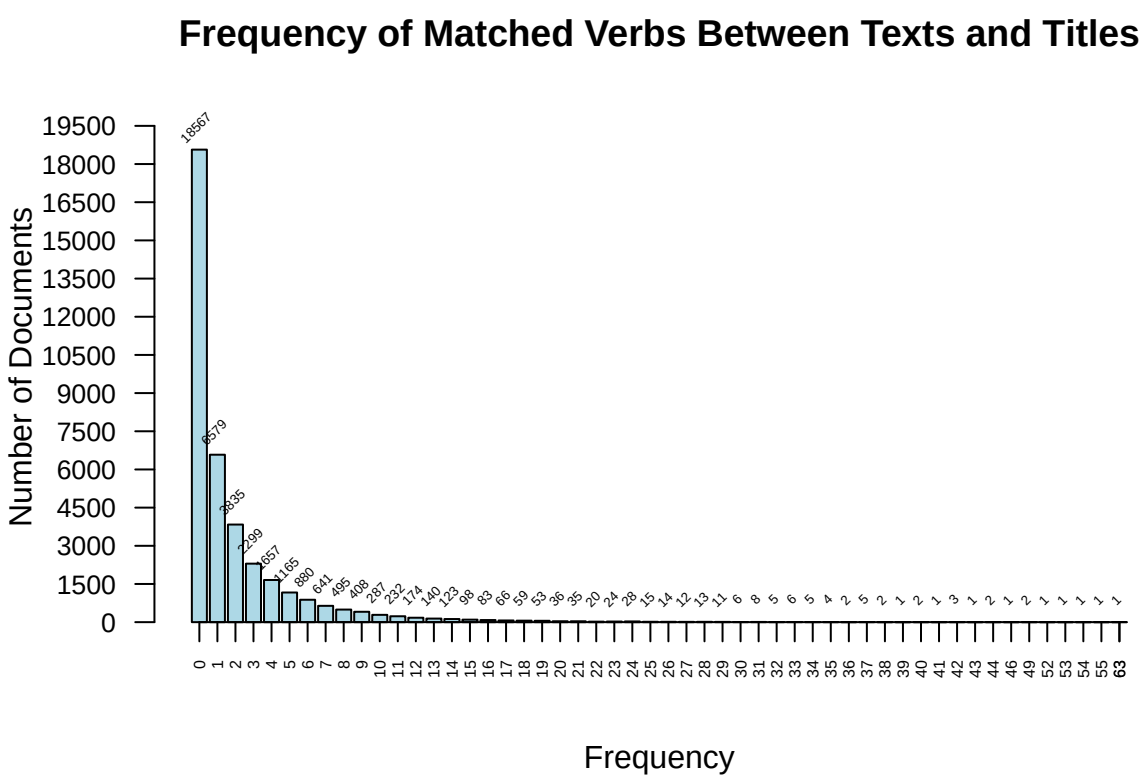


Y la **tabla de frecuencias**:

```
freq_table1
```

```
freq_verbs_summary
    0     1     2     3     4     5     6     7     8     9    10    11    12
15062  6818  4189  2915  1958  1401  1137   872   678   532   413   325   285
   13    14    15    16    17    18    19    20    21    22    23    24    25
  217   180   179   128   116    88    94    73    56    45    41    39    33
   26    27    28    29    30    31    32    33    34    35    36    37    38
   33    19    19    13    20    14    17    10    10     9     6     3     6
   39    40    41    42    43    44    45    46    47    49    50    51    52
    4     8     2     5     2     5     4     1     2     1     1     1     1
   53    56    58    59    60    61    62    63    64    67    73    74    75
    4     1     1     1     1     1     2     1     1     1     1     1     1
   89    92   108
    1     1     1
```

También podemos hacer esto con los **títulos** y quedaría lo siguiente:



Con su **tabla de frecuencias**:

```
freq_table2
```

```
freq_verbs_titles
    0     1     2     3     4     5     6     7     8     9    10    11    12
18567  6579  3835  2299  1657  1165   880   641   495   408   287   232   174
   13    14    15    16    17    18    19    20    21    22    23    24    25
  140   123    98    83    66    59    53    36    35    20    24    28    15
   26    27    28    29    30    31    32    33    34    35    36    37    38
   14    12    13    11     6     8     5     6     5     4     2     5     2
   39    40    41    42    43    44    46    49    52    53    54    55    63
    1     2     1     3     1     2     1     2     1     1     1     1     1
```

Ahora hagamos un testeo para una determinada **frecuencia de summary**. Para ello, creamos la función `test_specific_frequency_summary`.

La función `test_specific_frequency_summary` busca documentos con una frecuencia **específica** de verbos en los **resúmenes**, muestra detalles de los verbos coincidentes entre texto y resumen, y presenta el **número de coincidencias por documento**. Si no hay documentos con esa frecuencia, informa que no se encontraron coincidencias.

Hagamos un test, por ejemplo, con los de frecuencia **43**:

```
test_specific_frequency_summary(43, freq_verbs_summary, list_verbs_text, list_verbs_summary_unique, cor
```

```
=== Testing for Frequency: 43 ===

Document ID: text5524
Text Verbs:
 pasar, regresar, preguntar, interesar, vivir, recordar, recordar, contar, presentar, casar, unir, acer

Summary Verbs:
 dar, decir, atender
Matched Verbs (Summaries) with Counts:
     decir ( 26 ),   dar ( 15 ),     atender ( 2 )
Matched Count (Summaries): 3

Document ID: text10317
Text Verbs:
 vivir, luchar, lograr, resultar, acostar, transmitir, renunciar, convencer, volver, sugerir, usar, onl

Summary Verbs:
 esperar, encontrar, venir, tener, suceder
Matched Verbs (Summaries) with Counts:
     tener ( 30 ),   venir ( 1 ),    encontrar ( 10 ),   esperar ( 1 ),      suceder ( 1 )
Matched Count (Summaries): 5
```

Podemos hacer lo mismo con los **títulos**, por ejemplo, con los que tengan **30** frecuencias y queda lo siguiente:

```
test_specific_frequency_titles(30, freq_verbs_titles, list_verbs_text, list_verbs_title_unique, corpus_
```

```
=== Testing for Frequency: 30 ===

Document ID: text801
Text Verbs:
 significar, rendir, ver, mirar, pasar, detener él, saludar, resultar, olvidir, llenar, sentar, comer, s

Title Verbs:
 lograr, hacer, ayudar
Matched Verbs (Titles) with Counts:
     hacer ( 25 ),   lograr ( 1 ),   ayudar ( 4 )
Matched Count (Titles): 3

Document ID: text1750
Text Verbs:
 especializar, tener, motivar, publicar, oir, escuchar, trabajar, tener, pasar, hablar, tener, poner, i

Title Verbs:
 tener, pedir, trabajar
Matched Verbs (Titles) with Counts:
     tener ( 27 ),   trabajar ( 3 )
Matched Count (Titles): 2

Document ID: text8043
Text Verbs:
 escribir, significar, regresar, estar, evitar, parecer, suceder, tener, adaptar él, presentar, ver, co
```

```
Title Verbs:
 tener, entrevistar
Matched Verbs (Titles) with Counts:
     tener ( 30 )
Matched Count (Titles): 1

Document ID: text9447
Text Verbs:
 nacer, llevar, tener, saber, nacer, adoptar, nacer, hacer, sospechar, decir, adoptar, decir, llevar, t

Title Verbs:
 descubrir, tener
Matched Verbs (Titles) with Counts:
     tener ( 25 ),   descubrir ( 5 )
Matched Count (Titles): 2

Document ID: text12070
Text Verbs:
 encerrar, incluir, generar, ordenar, hacer, partir, empezar, erigir, asegurar, sufrir, frenar, trabaja

Title Verbs:
 bastar, construir, hacer
Matched Verbs (Titles) with Counts:
     hacer ( 11 ),   construir ( 17 ),   bastar ( 2 )
Matched Count (Titles): 3

Document ID: text16974
Text Verbs:
 sobrevivir, formar, sentir, pensar, sentir, tener, vivir, ver, costar, salir, decir, sentir él, decir,

Title Verbs:
 sentir
Matched Verbs (Titles) with Counts:
     sentir ( 30 )
Matched Count (Titles): 1
```