Recuperación y análisis de texto con R

Clase 5 - Educación Permanente FCS

Mag. Elina Gómez (UMAD)

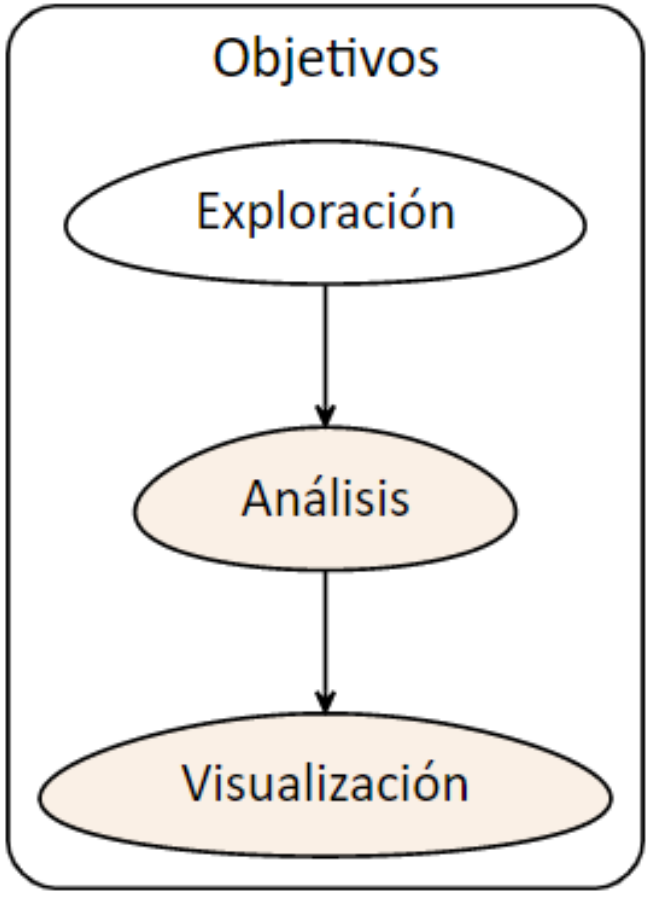
elina.gomez@cienciassociales.edu.uy

www.elinagomez.com

Mag. Gustavo Méndez Barbato

gustavo.mendez@cienciassociales.edu.uy

Objetivos de hoy



Objetivos de hoy

- Presentación del paquete **quanteda** para el análisis de textos.
- Nubes de palabras
- Asociaciones
- Redes
- Categorías gramaticales con paquetes **spacyr** y **udpipe**

Algunas nociones previas

- Pre procesamiento: antes del análisis necesitamos realizar un conjunto de actividades destinadas a preparar el texto
- Su complejidad dependerá de las características del texto que tengamos
- Siempre está en función de los objetivos de la tarea que nos propongamos
- No hay una receta única

Algunas nociones previas

Pre-procesamiento:

- Limipeza o eliminación del ruido (ej. números de página, encabezados, saltos de línea, etc.)
- Normalización:

1. Tokenización: dividir el texto en unidades más pequeñas (caracteres, palabras, oraciones)
2. Steeming: cortar las palabras para quedarnos con la raíz (gato, gata, gatitos -> gat*)
3. Lematización: quedarnos con la forma canónica de la palabra (tipo las entradas de un diccionario)
4. Homogeneización: eliminación de números, puntuación, símbolos, convertir a minúsculas, eliminación de stopwords (palabras no sustantivas para el análisis), etc.

Algunas nociones previas

- El lenguaje (y sus usos) son complejos y eso genera muchas veces necesidad de desambiguar.
- Ejemplo: tokenizando por palabras ¿Nueva York o Graciela Bianchi son una palabra o dos? ¿Las tomamos como un token o como dos?
- En última instancia, la respuesta es una decisión que hay que justificar como cualquier otra decisión metodológica
- Además: es importante conocer cómo tokenizan las diferentes herramientas, especialmente si vamos a usar distintas

Algunas nociones previas

- **Corpus**: colección de textos escritos, orales o ambos. En linguística [1] conjunto cerrado de textos o de datos destinado a la investigación científica concreta. [2] Muestra representativa de una lengua (datos lingüísticos reales que reflejen el uso de la lengua)
- **Palabras**: en nuestro contexto de análisis son palabras distintas en un corpus
- **Tokens**: en nuestro contexto de análisis son el total de palabras (apariciones) en un corpus (siempre que tokenizemos por palabras)

Paquete quanteda

- **quanteda** es un paquete R para administrar y analizar datos textuales desarrollados por *Kenneth Benoit* y otros colaboradores. Su desarrollo inicial fue apoyado por la beca del Consejo Europeo de Investigación.
- El paquete está diseñado para usuarios de R que necesitan aplicar el procesamiento de lenguaje natural a los textos, desde los documentos originales hasta el análisis final.
- Sus capacidades coinciden o superan las que se ofrecen en muchas aplicaciones de software para usuarios finales, muchas de las cuales son caras y no de código abierto.

quanteda

- Documentación quanteda
- Tutorial
- Más información

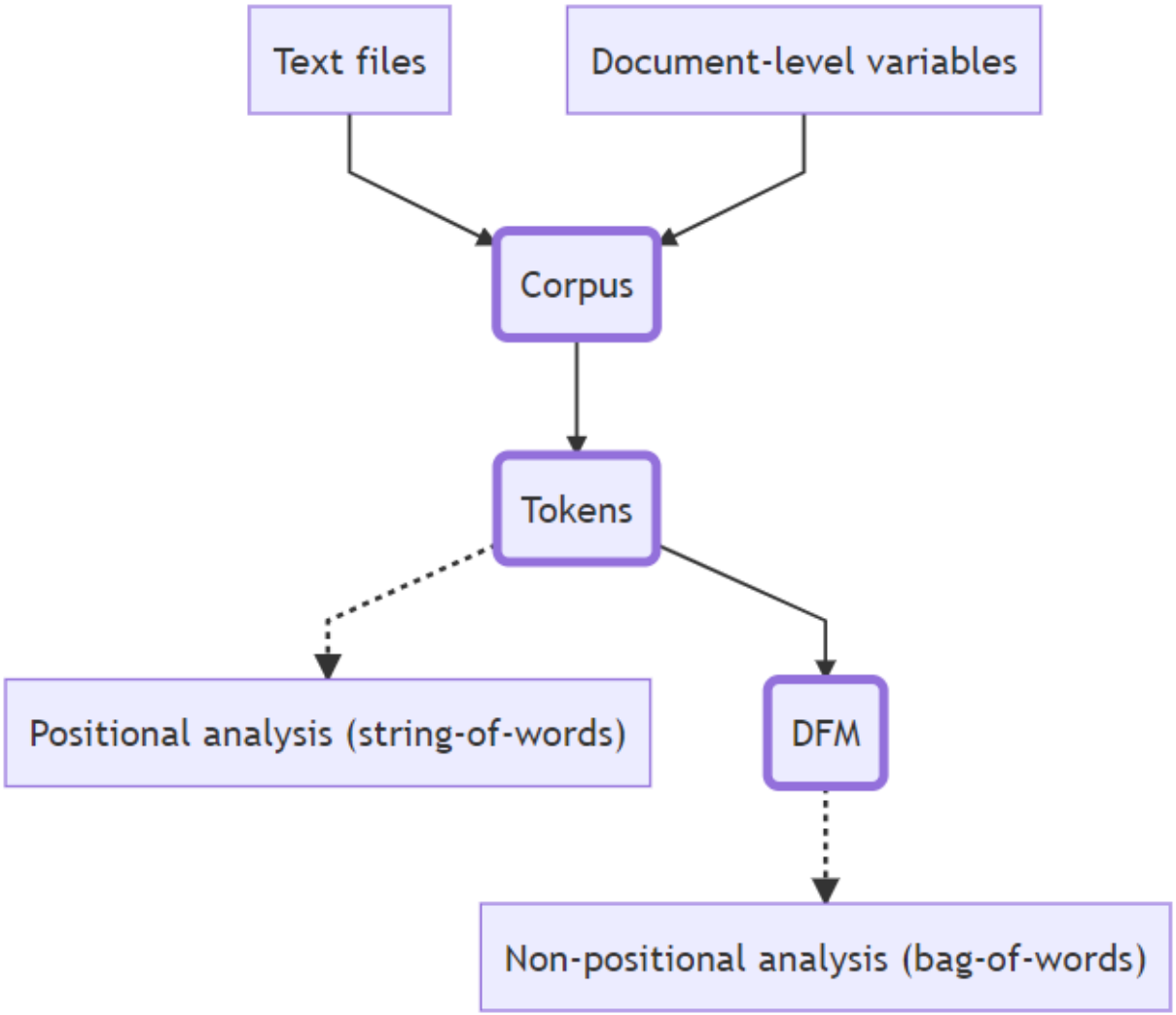
Algunos de los conjuntos de funciones de quanteda fueron independizándose en paquetes específicos: `quanteda.textplots`, `quanteda.textmodels` y `quanteda.textstats`

quanteda: objetos

`quanteda` tiene sus propios tipos de objetos

- `corpus`
  Guarda cadenas de caracteres y variables en un marco de datos
  Combina textos con variables a nivel de documento
- `tokens`
  Almacena tokens en una lista de vectores
  Conserva las posiciones de las palabras
- `dfm` (document-feature matrix)
  Representa frecuencias de características en documentos en una matriz
  No tiene información sobre las posiciones de las palabras

quanteda: flujo



Fuente: quanteda.io

quanteda: análisis básicos

- Frecuencia de palabras y frecuencias ponderadas de términos
- Asociación y correlación
- Identificación del contexto de aparición de palabras y co-ocurrencia
- Uso de diccionarios para identificar o clasificar textos
- Visualizaciones específicas: nubes de palabras, redes de co-ocurrencia

Caso práctico: LUC en el Senado

- Análisis de los discursos vinculados a la discusión de la *Ley de Urgente Consideración (LUC)* en la Cámara de Senadores del 5 de junio de 2020.

"Limpieza" del texto

- Creo un Document feature matrix (DFM), aplicando algunos argumentos que me permiten limpiar las palabras que no me interesan al efecto del análisis.
  - Homogeneizo las palabras en minúscula
  - Elimino números
  - Elimino puntuaciones
  - Elimino stopwords (por defecto y lista propia con palabras varias (ej. "Risas"))
  - Elimino palabras con pocos caracteres (1 y 2)

"Limpieza" del texto

```
dfm_intervenciones <- quanteda::dfm(quanteda::tokens(intervenciones$speech,
remove_punct = TRUE, ##saco puntuación
remove_numbers = TRUE), #saco números
tolower=TRUE, #paso a minúsculas
verbose = FALSE) %>%
  quanteda::dfm_remove(pattern = c(quanteda::stopwords("spanish"),tolower(intervenciones$legislator)),
  min_nchar=3)%>% ##saco palabras específicas
   quanteda::dfm_trim(min_termfreq = 6)%>%
  quanteda::dfm_group(groups = intervenciones$party) #defino grupos
```

Ponderación

- Como factor de ponderación del dfm puedo usar la métrica *tf-idf* que relativiza el peso de cada término, poniendo en relación la frecuencia de aparición por el inverso de la frecuencia en los documentos.
- Ayuda a identificar los términos más frecuentes en un documento pero que no lo son en todos.
- No es posible usarlo en funciones que impliquen agrupación.
- La función de quanteda es *dfm_tfidf()*

Nubes de palabras: general

Las nubes de palabras las hago con la función **textplot_wordcloud** del paquete **quanteda.textplot**

```
quanteda.textplots::textplot_wordcloud(dfm_tfidf(dfm_intervenciones), min.count = 2,max_words = 200,
   random.order = FALSE,colors = RColorBrewer::brewer.pal(8,"Dark2"),comparison = F)
```

Nubes de palabras: general

Nubes de palabras: grupos

Para hacer nubes de palabras comparando entre grupos de interés, agregamos el argumento **comparison = T**

```
quanteda.textplots::textplot_wordcloud(dfm_intervenciones, min.count = 2,max_words = 500,
      random.order = FALSE,colors = RColorBrewer::brewer.pal(8,"Dark2"),comparison = T)
```

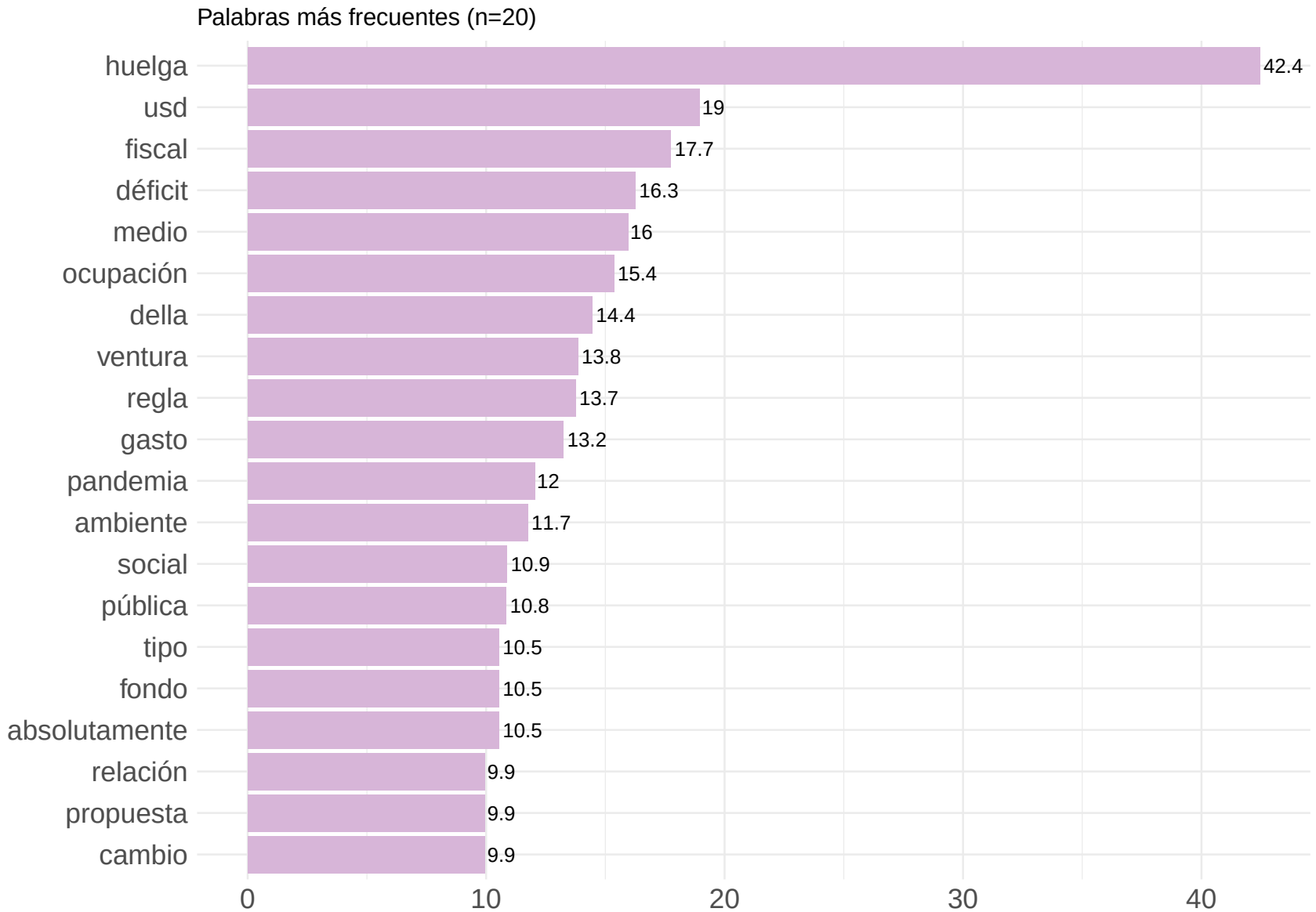
Nubes de palabras: partidos

Palabras más frecuentes

Para analizar las palabras más frecuentes utilizo la función *topfeatures()*

```
topfeatures(dfm_intervenciones,20)
```

Palabras más frecuentes:



Asociación de palabras

- Buscamos la asociación de palabras en los documentos
- Analizamos la asociación con las palabras: *sindicato*, *reforma*
Asociación de palabras
- Utilizamos la función *textstat_simil* del paquete **quanteda.textstats**, cuyos argumentos son el/los términos con los que quiere buscar una asociación en un *dfm* determinado.
- Defino el método de similitud ("correlation", "cosine", "jaccard", "ejaccard", "dice", "edice", "hamman", "simple matching")

```
quanteda.textstats::textstat_simil(dfm_tfidf(dfm_intervenciones),selection = "sindicato",
                                   method = "correlation",margin = "features")%>%
  as.data.frame()%>%
  dplyr::arrange(-correlation)%>%
  dplyr::top_n(15)
```

Asociaciónde palabras: $sindicato

feature1	feature2	correlation
plantea	sindicato	1
brasil	sindicato	1
pbi	sindicato	1
período	sindicato	1
inflación	sindicato	1
pone	sindicato	1
mayores	sindicato	1
recién	sindicato	1
obras	sindicato	1
ayer	sindicato	1
estrategia	sindicato	1
plantear	sindicato	1
asimismo	sindicato	1
preopinante	sindicato	1
poblaciones	sindicato	1
aquellas	sindicato	1
ordenamiento	sindicato	1

Asociación de palabras: $reforma

feature1	feature2	correlation
emergencia	reforma	1.0000000
mercado	reforma	1.0000000
relación	reforma	0.9999932
conjunto	reforma	0.9999705
enfrentar	reforma	0.9999058
razonable	reforma	0.9999058
pregunta	reforma	0.9999058
interno	reforma	0.9999058
mujeres	reforma	0.9999058
integral	reforma	0.9999058
entiende	reforma	0.9999058
cálculo	reforma	0.9999058

Contexto de la palabra: kwic

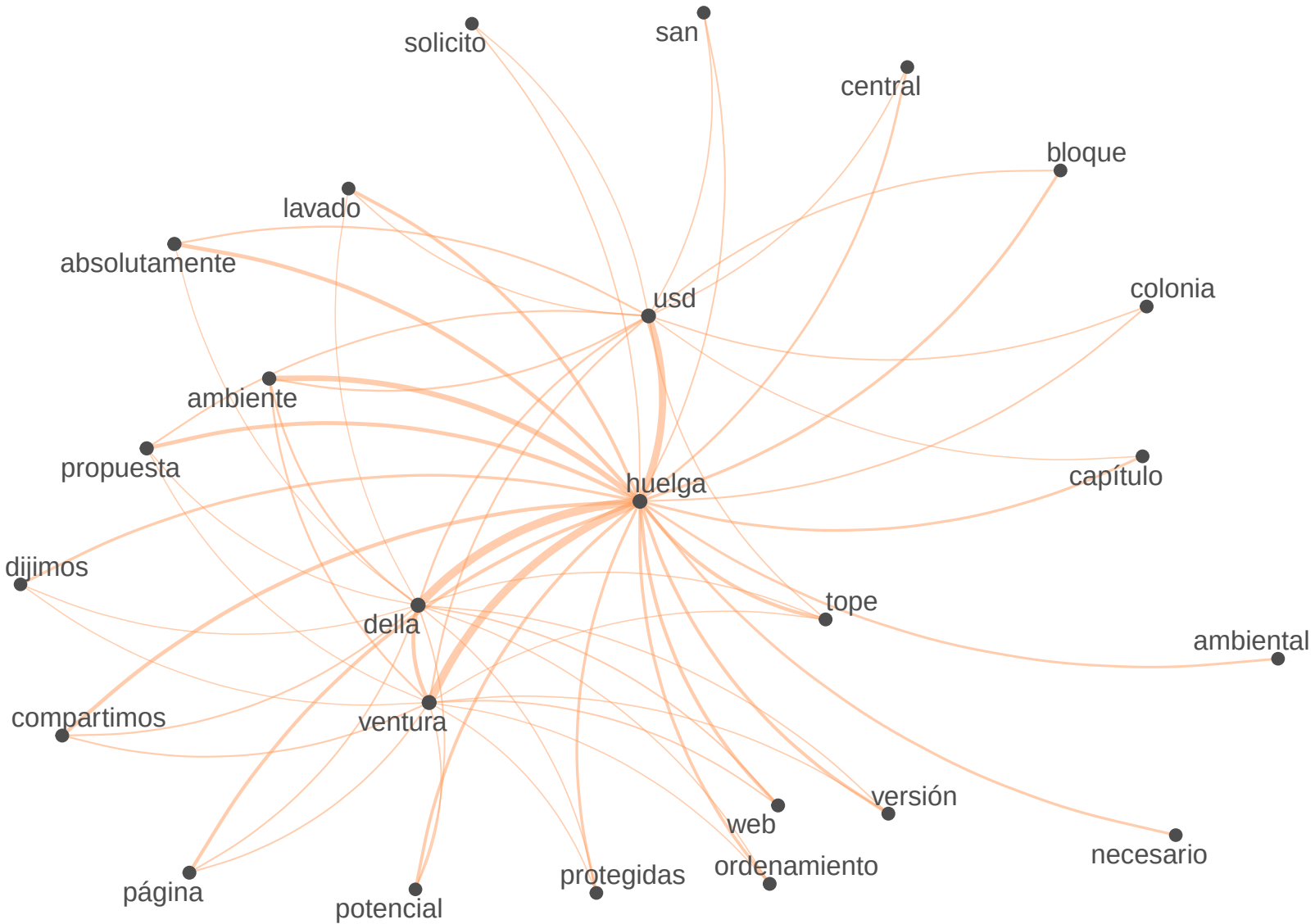
- Con la función `quanteda::kwic()` podemos ver el contexto de aparición de una palabra, término o frase, según una ventana (cantidad de palabras previas y posteriores) determinada.
- Extraer el contexto de ciertos términos puede ser de utilidad para construir un nuevo corpus y realizar un análisis focalizado y/o comparativo.

```
quanteda::kwic(quanteda::tokens(intervenciones$speech,
remove_punct = TRUE,
remove_numbers = TRUE),
pattern = quanteda::phrase(c("ley de urgente consideración")),
window = 5)
```

Contexto de la palabra: redes de co-ocurrencia

- Con la función `quanteda.textplots::textplot_network` podemos hacer redes de co-ocurrencia entre términos.

Contexto de la palabra: redes de co-ocurrencia



spacyr y udpipe

- Categorías gramaticales (ampliar)