Guía rápida de análisis de corpus (con AntConc)

Borja Navarro Colorado
Universidad de Alicante

18 de diciembre de 2014

Índice general

1 Introducción

AntCon es un programa de ayuda para el análisis de corpus. Permite extraer datos de amplios corpus textuales como frecuencias de palabras, colocaciones, concordancias, búsquedas mediante expresiones regulares, y alguna cosa más.

En este pequeño tutorial se muestra cómo utilizar sus funciones básicas. Se sigue un orden de trabajo estándar: desde la apertura del corpus hasta la extracción de datos.

AntConc no analiza el corpus. Simplemente muestran los textos de manera diferente (por frecuencias, por patrones de búsqueda, por palabras, etc). En análisis debe hacerlo un especialista. Bien utilizado, te permitirá detectar aspectos de los textos y de la lengua del corpus que de otra manera no se podrían detectar ni analizar. Pero el análisis e interpretación de datos, al final, siempre es tarea del investigador.

2 Descargar AntConc

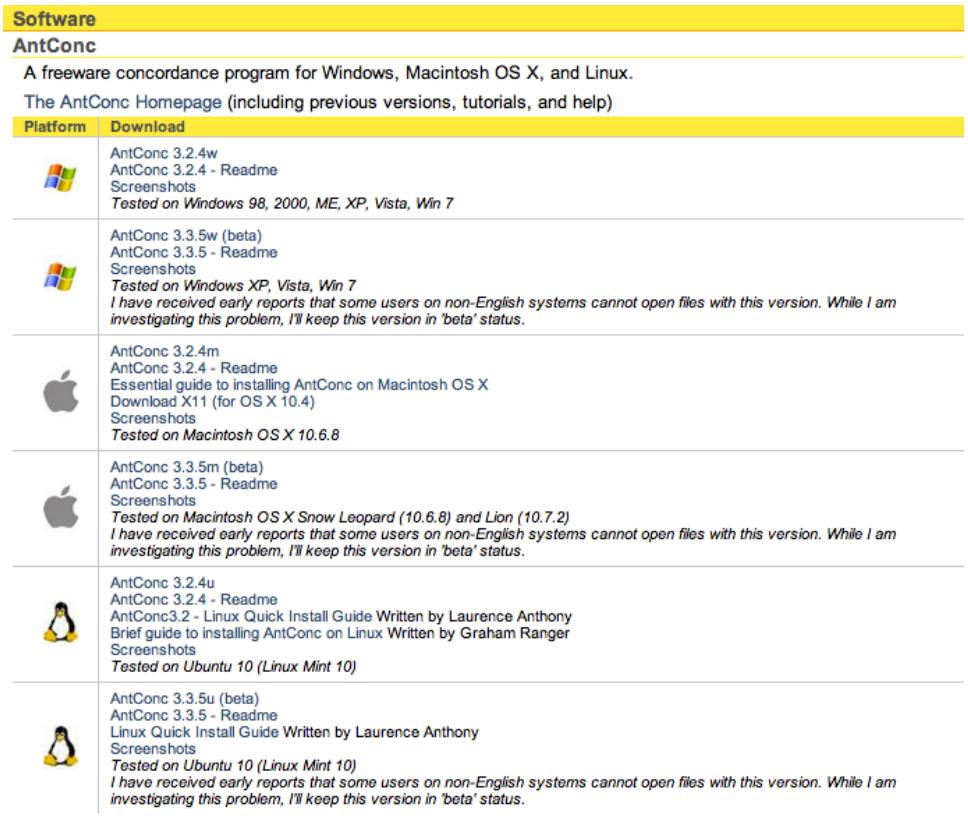
Para descargar AntConc, sigue estos pasos:

1. Ve a la página web de AntConc[1]:

   `http://www.antlab.sci.waseda.ac.jp/software.html`

2. Selecciona la versión de AntConc según el sistema operativo de tu ordenador: Windows, Mac o Linux.

[1] O busca "Antconc" en Google o cualquier otro buscador. Posiblemente la primera propuesta será la web de AntConc



Software

AntConc

A freeware concordance program for Windows, Macintosh OS X, and Linux.

The AntConc Homepage (including previous versions, tutorials, and help)

Platform	Download
	AntConc 3.2.4w AntConc 3.2.4 - Readme Screenshots *Tested on Windows 98, 2000, ME, XP, Vista, Win 7*
	AntConc 3.3.5w (beta) AntConc 3.3.5 - Readme Screenshots *Tested on Windows XP, Vista, Win 7* *I have received early reports that some users on non-English systems cannot open files with this version. While I am investigating this problem, I'll keep this version in 'beta' status.*
	AntConc 3.2.4m AntConc 3.2.4 - Readme Essential guide to installing AntConc on Macintosh OS X Download X11 (for OS X 10.4) Screenshots *Tested on Macintosh OS X 10.6.8*
	AntConc 3.3.5m (beta) AntConc 3.3.5 - Readme Screenshots *Tested on Macintosh OS X Snow Leopard (10.6.8) and Lion (10.7.2)* *I have received early reports that some users on non-English systems cannot open files with this version. While I am investigating this problem, I'll keep this version in 'beta' status.*
	AntConc 3.2.4u AntConc 3.2.4 - Readme AntConc3.2 - Linux Quick Install Guide Written by Laurence Anthony Brief guide to installing AntConc on Linux Written by Graham Ranger Screenshots *Tested on Ubuntu 10 (Linux Mint 10)*
	AntConc 3.3.5u (beta) AntConc 3.3.5 - Readme Linux Quick Install Guide Written by Laurence Anthony Screenshots *Tested on Ubuntu 10 (Linux Mint 10)* *I have received early reports that some users on non-English systems cannot open files with this version. While I am investigating this problem, I'll keep this version in 'beta' status.*

Figura 1: Página de descarga de AntConc

3. Pincha en AntConc xxx (donde xxx es la versión del programa). Con esto empezará la descarga...

2.1. ¿Versión beta o estable?

Si te has fijado, para cada sistema operativo hay dos versiones: una "beta" y otra que no pone nada. La primera es una versión de prueba. La segunda es la versión estable.

Salvo que seáis usuarios expertos, descargad siempre la versión estable.

Efectivamente, la que no pone nada es la versión estable. Es una versión cerrada, definitiva, lista para ser utilizada por cualquier usuario. Las versiones betas, por el contrario, son las versiones que aún no son definitivas, es decir, que aún pueden tener algún error. Están pensadas para que sean utilizadas por usuarios expertos, de tal manera que, si éstos detectan errores, se los comuniquen a los desarrolladores y conseguir, así, mejorar el

programa.

Por ello lo mejor es que descargues y te instales la versión estable, la que no tiene indicación beta.

3 Instalación

Una vez descargado el fichero del programa, no hace falta seguir ningún proceso de instalación. Simplemente:

1. Pon el fichero descargado (corta y pega) en una carpeta de referencia, que te sea fácil acceder a ella. Por ejemplo, puedes crear una carpeta en tu escritorio que se llame "AntConc" y dentro de ella poner el fichero. Así tendrás acceso rápido al programa.

Figura 2: Icono del programa

2. Pincha (doble click) en el fichero. El programa arranca y está listo para trabajar. La interfaz de usuario que vas a ver es ésta:

4 Cargar el corpus

El primer paso para analizar un corpus con AntConc es cargarlo o abrirlo en el programa.

Para ello necesitamos un corpus bien diseñado y compilado. Si el corpus está mal diseñado y mal compilado, los resultados del análisis serán erróneos o falsos. Por ello es importante dedicarle el tiempo suficiente al diseño y análisis del corpus.

Físicamente el corpus será uno o más ficheros de texto, a ser posible con formato de texto sencillo (extensión .txt). Si son varios ficheros de texto,

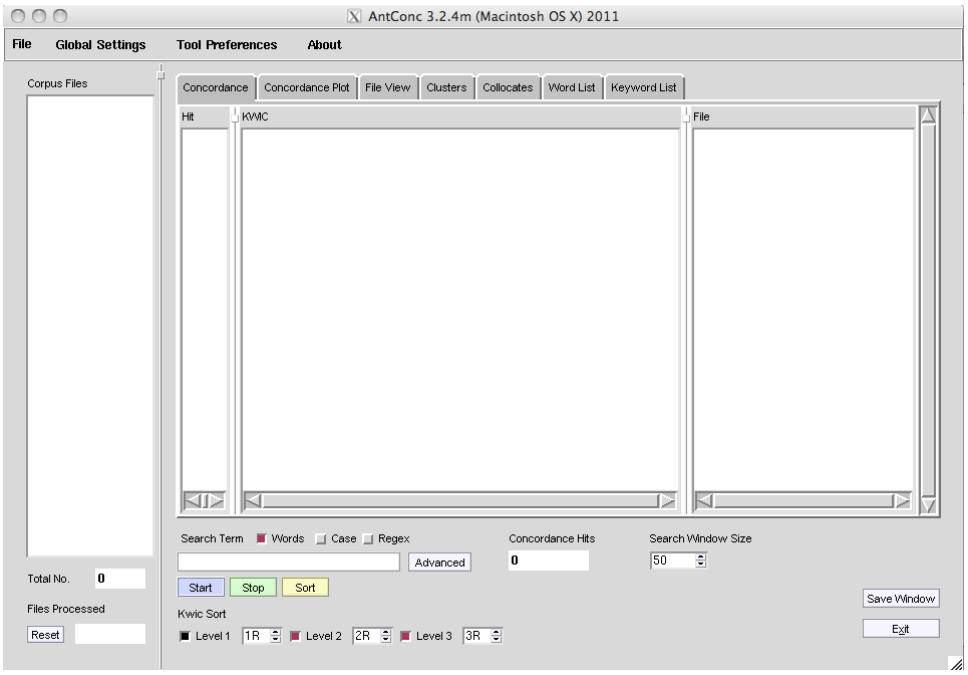


Figura 3: Interfaz de usuario de AntConc

ponlos todos en una única carpeta donde sólo estén los ficheros de texto del corpus.

AntConc permite abrir también ficheros HTML y ficheros XML. Pero por ahora vamos a trabajar con textos sencillo, sin ningún tipo de anotación ni metadato.

Como corpus de ejemplo, voy a utilizar la novela *Trafalgar* de Benito Pérez Galdós (1873), en la versión de la Biblioteca Virtual Miguel de Cervantes. En este caso el corpus lo compone un único fichero de texto en formato sencillo, en el que sólo está el texto de la novela, sin nigún tipo de metadato ni anotación.

Para cargar el corpus en AntConc, ve al menú de herramientas y pincha en "File". Si, como en este ejemplo, el corpus es un único fichero, pincha en "Open File", busca el fichero entre el árbol de carpetas, selecciónalo y pulsa "open".

Si el corpus está formado por dos o más ficheros y los tenemos todos en una carpeta exclusiva, pincha en .ºpen dir", busca y seleccionar la carpeta, y pulsa .ºpen". En esta carpeta o directorio sólo pueden estar los ficheros .txt que forman el corpus. Quita cualquier fichero que no sea del corpus, pues AntConc lo cargaría como parte del corpus.

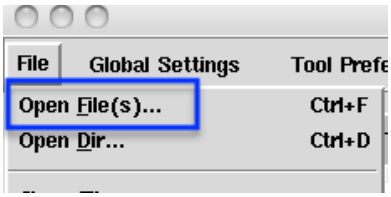


Figura 4: Opción para cargar corpus de un solo fichero

Figura 5: Opción para cargar corpus de dos o más ficheros

Con esto ya tenemos el corpus cargado. Si todo ha ido bien, debe aparecer en la ventana "Corpus Files" de AntConc (a la izquierda) el nombre de el o los ficheros que forman el corpus.

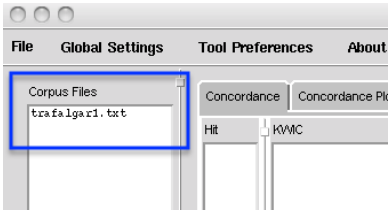


Figura 6: Ficheros del corpus cargados en AntConc

4.1. Problemas con la codificación.

Para que los análisis sean correctos, debes indicar a AntConc cuál es la codificación de caracteres de tu fichero.

La codificación de caracteres[2] no es más que la relación entre cada carácter y su representación por la máquina. Cada carácter textual es reconocido por el ordenador por un código numérico binario: lo que nosotros vemos como un carácter o letra en la pantalla, para el ordenador es un conjunto de ocho (o siete) ceros y unos.

[2]Para más información, consulta `http://es.wikipedia.org/wiki/Codificacion_de_caracteres`

El problema es que hay diferentes sistemas para codificar caracteres.

El sistema más común es ASCII[3]. Éste es el que está en la mayoría de los ordenadores. Pero ASCII es limitado: sólo puede representar 128 caracteres. Para el inglés es suficiente, pero para el resto de idiomas resulta escaso. Así, la norma ASCII original no puede representar caracteres españoles como la eñe, las vocales acentuadas o la ce con cedilla ("ç").

Para español se pueden utilizar dos sistemas de codificación: la norma ISO 8859-1, también conocida como latin-1[4], que define la codificación del alfabeto latino (y por tanto incluye eñes y vocales acentuadas); y la norma Unicode UTF-8 [5]. Ésta, al ser Unicode, puede representar en principio cualquier carácter.

Si tu corpus está escrito en español o en lengua europea, posiblemente tenga codificación Latin-1 o UTF-8. AntConc, por defecto, espera ficheros de texto con codificación Latin-1. Si tu corpus tienes esta codificación, no hay problema.

Ahora bien, si tu corpus está codificado con UTF-8, debes indicárselo a AntCon. Para indicar la codificación del corpus, sigue estos pasos:

1. Ve a la barra de menú y pincha en "Global Setting". Se abrirá una ventana donde se pueden cambiar la configuración general del sistema.

2. Pincha en "Language Encoding". Verás que la codificación que ahora está usando AntConc es Latin-1.

3. Para cambiar la codificación, pincha en "Edit" y selecciona la codificación de tu corpus. UTF-8 está en "Standar encoding ¿Unicode (UTF-8)".

4. Finalmente, pulsa en "Apply". Con esto AntConc ya sabe cuál es la codificación de tu corpus y mostrará los caracteres correctamente.

Si te fijas, la lista de codificaciones es muy amplia. Hay una o varias normas de codificación por cada familia lingüística.

Lo más recomendable es trabajar con Unicode: UTF-8 o su versión ampliada UTF-16. Así, en principio, evitarás problemas de codificación, sobre todo si tu corpus tiene textos en diferentes idiomas.

[3] `http://es.wikipedia.org/wiki/ASCII`
[4] `http://es.wikipedia.org/wiki/ISO_8859-1`
[5] `http://es.wikipedia.org/wiki/UTF-8`

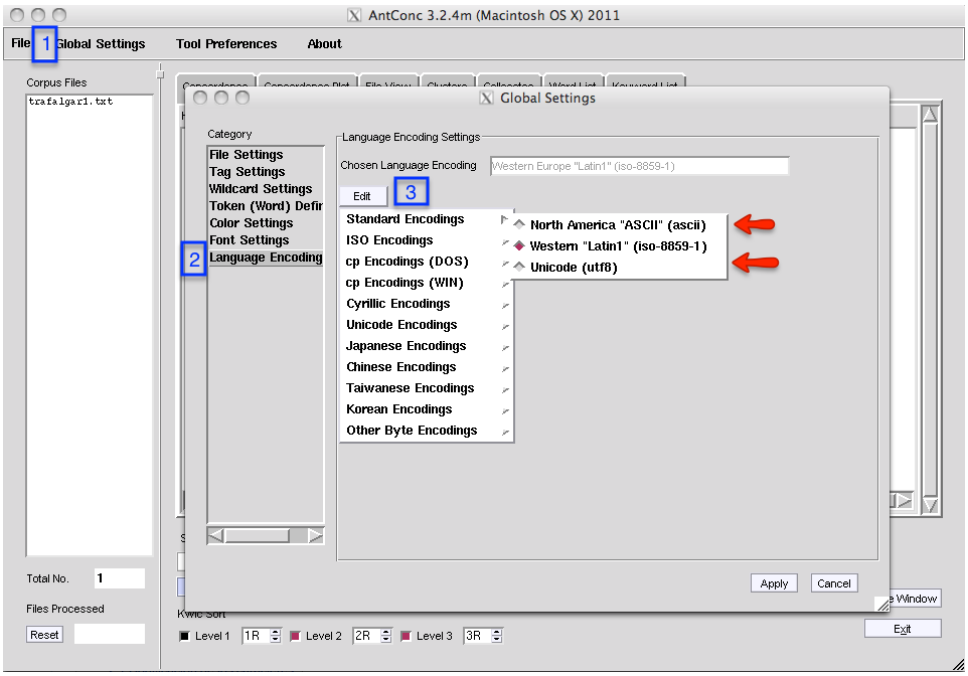


Figura 7: Opción para cambiar la codificación.

4.2. ¿Pero cuál es la codificación de mi corpus?

Es posible que no sepas de antemano la codificación de tu corpus. Depende del editor de texto sencillo que hayas utilizado para compilarlo y del sistema operativo en el que trabajes, el corpus puede tener una codificación u otra. Lo más probable es que sea Latin-1.

Hay diferentes formas de saber qué codificación tiene un texto, pero depende del editor que utilices.

Una forma sencilla es abrir el corpus con un editor de texto sencillo y pinchar en "Guardar como...", igual que si fueras a hacer una copia del texto. En la ventana "Guardar como..."suele aparecer, en alguna parte, la codificación que utilizará el editor para guardar el texto. Puedes cambiarla (y guardar el corpus con otro codificación), o puedes cancelar y adaptar la codificación de AntConc a tu corpus.

Vale, quizá no sea la mejor forma de saber la codificación de caracteres

de un texto, pero es sencilla y funciona.

5 Extracción de frecuencias

Una vez que hemos cargado el corpus en AntCon , ya podemos empezar a desarrollar análisis de corpus.

El primer tipo de análisis, el más sencillo, es el análisis de frecuencias.

Analizar frecuencias no es más que contar las palabras de un corpus y mostrarlas en una lista ordenada desde la palabra más frecuente hasta la menos frecuente (o a la inversa).

Para calcular las frecuencias de un corpus ya cargado en AntConc, sigue los siguientes pasos:

1. Pulsa en la pestaña "Word List"
2. Pulsa en el botón "Start"

Y eso es todo. Si el corpus está bien cargado, te habrá aparecido en la ventana de AntCon una lista de palabras. Estas palabras son las palabras que hay en el corpus, ordenadas desde la más frecuente a la menos frecuente.

Esta es la salida con el corpus "Trafalgar":

Los datos que muestra son los siguientes:

- La primera columna (*Rank*) muestra la posición de la palabra.
- La segunda columna (*Freq*) muestra el número de veces que se repite esa palabra en el corpus
- La tercer columna (*Word*) es la palabra en sí.
- De la cuarta columna no hablaremos ahora. Lo haremos cuando expliquemos la lematización.

Además, encima de estas columnas aparece el número total de *Tokens* (conjunto de caracteres separados por espacio en blanco) y el número de *Types* (tokens iguales).

Con esto el programa ya ha terminado su trabajo: ha calculado las frecuencias de las palabras del corpus. El corpus es el mismo, pero se nos muestra de manera diferente, de manera transversal: en vez del orden lineal, el texto aparece ordenado por frecuencias léxicas. Ahora empieza el trabajo del investigador para estudiar y analizar esos datos: la lista de frecuencia. En este caso, ¿qué rasgos del texto podemos observar que no podríamos ver si analizáramos el texto de manera lineal?

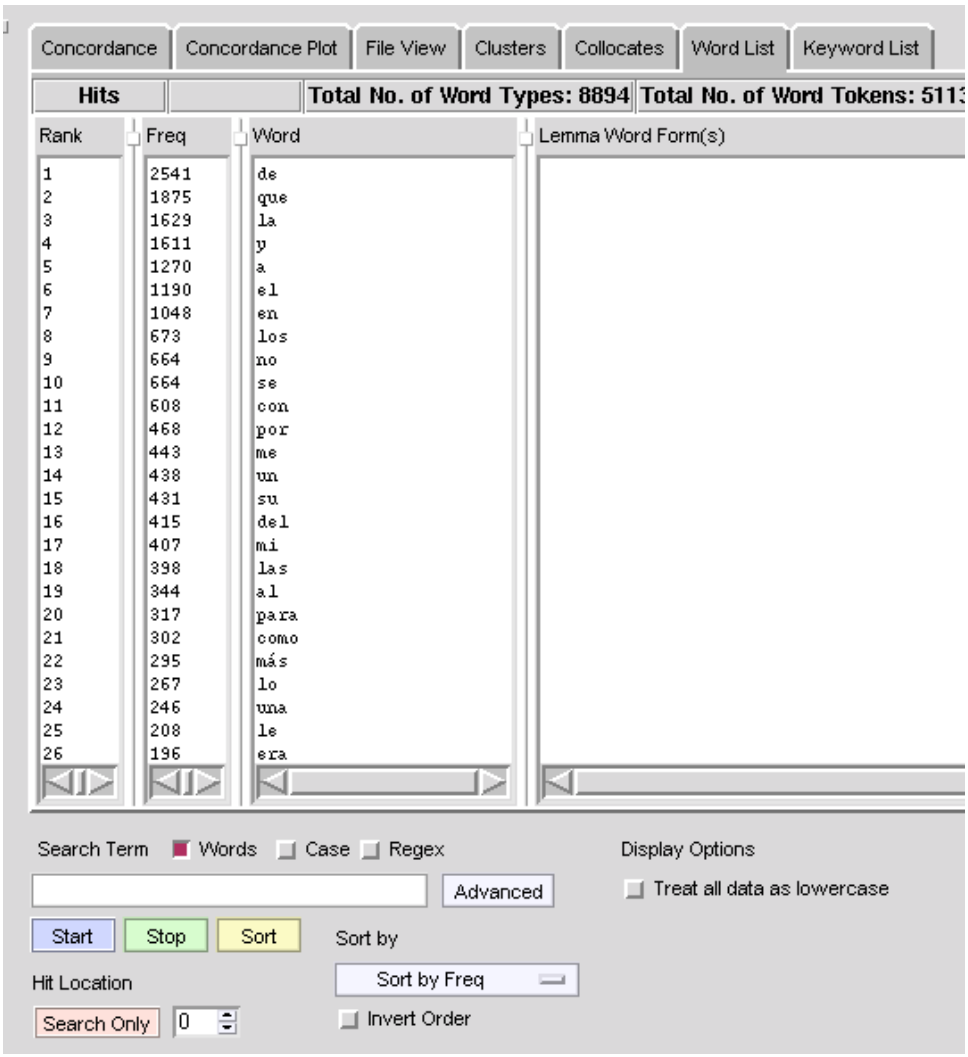


Figura 8: Frecuencias de palabras.

5.1. ¿Aparecen caracteres extraños?

Has cargado el corpus, has hecho un primer análisis de frecuencias y te aparecen caracteres extraños, ¿es así?

Si te ocurre esto, tienes un problema con la codificación. Tu corpus no tiene la codificación que espera AntConc.

Ve al apartado anterior sobre codificación y cambia la codificación que espera AntConc. Si trabajas con lenguas europeas, prueba primero a cambiar la codificación a UTF-8. Una vez cambiada la codificación, vuelve a la pestaña "WordList" y calcula las frecuencias de nuevo (pincha en "start").

Si aún así continúan apareciendo caracteres extraños, la codificación de tu corpus no es estándar. Intenta averiguar cuál es la codificación, y adapta AntConc a ésta.

5.2. Primer análisis, primeros problemas.

Una vez calculadas las frecuencias del corpus, comienza el proceso de analizar la lista resultante. Buscamos datos, rasgos, evidencias que ayuden a conocer mejor el corpus, la lengua, el estilo, el autor, la obra...

Sin embargo, antes de encontrar algo interesante, es recomendable refinar el cálculo de frecuencias. Veamos el resultado de este primer cálculo, qué problemas tiene y cómo se pueden refinar los datos.

Analiza los datos de frecuencia de *Trafalgar* de la Figural 8. Con este cálculo sabemos que el número total de *tokens* de *Trafalgar* es de 51135, y el número total de *types* es de 8894. Es decir, que hay 8894 formas diferentes en la novela.

Con estos dos datos se pueden ya deducir algunas cosas, como el tamaño objetivo del texto o la riqueza léxica de la novela. Luego se volverá sobre esto.

Sin embargo, analizando la lista nos encontramos con datos como estos:

Posición	Palabra
6	el
40	El

Cuadro 1: Posición diferente según mayúsculas y minúsculas.

Con lo que tenemos un primer problema: el cálculo distingue como palabras diferentes aquellas que se diferencian sólo por las mayúsculas. Efectivamente, tal y como está ahora hecho el cálculo, se consideran palabras (*tokens*) diferentes “la” y “La”, “a” y “A”, “al” y “Al”, “parecía” y “Parecía”, etc.

Este primera problema será tratado en la siguiente sección.

Hay un segundo problema. Si nos fijamos en las diez palabras más frecuentes, nos encontramos con estas palabras:

de que la y a el en los no se

¿Qué nos dicen las palabras más frecuentes del corpus? Exactamente nada, no dicen nada. Por un lado, ninguna de esas palabras tiene significado

léxico (salvo “no”) y, por otro, son las palabras más frecuentes en cualquier corpus en español.

El primer nombre con significado léxico está en la posición 39 y es “amo”. Esta palabra ya nos dice algo sobre la novela (si la has leído, sabrás por qué). Pero hemos tenido que ir hasta la posición 39. Este es el segundo problema: qué hacer con las palabras frecuentes sin significado léxica y comunes a cualquier texto. Como veremos después, este tipo de palabras se suele filtrar con un una lista de *stopwords*.

Existe un tercer problema: todas las palabras con variación morfológica se cuentan como palabras diferentes. Efectivamente, si te fijas en los verbos, cada forma flexionada aparece como una palabra distinta. Lo mismo ocurre con nombres en singular y plural, adjetivos, etc.

¿Y si queremos saber el número de veces que se utiliza un verbo, con independencia de su flexión?

La solución a esto requiere un procesamiento computacional del texto más complejo. Se puede utilizar las concordancias o las expresiones regulares para extraer todas las formas flexionadas de una palabra. O se puede lematizar el corpus. Estas soluciones serán tratadas en próximas secciones.

5.3. Mayúsculas y minúsculas.

AntConc, como todo programa de análisis de corpus, permite eliminar la distinción entre mayúsculas y minúsculas.

Para el ordenador, las mayúsculas y las minúsculas son letras totalmente diferentes, pues tienen códigos de codificación diferentes.

En AntConc, tenemos la opción *Treat all data as lowercase*. Si se activa esta opción, AntConc hará los cálculos de frecuencia considerando todas las letras como minúsculas.

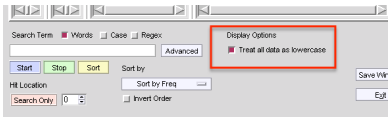


Figura 9: Opción para no diferencias mayúsculas de minúsculas.

Si vuelves a realizar el cálculo de frecuencias con esta casilla activada (pulsa ”Start”), verás que los resultados cambian:

Efectivamente, ahora hay menos *types*: 8458, frente a los 8894 *types* que había calculado considerando mayúsculas y minúsculas como letras diferentes. Palabras como “Al” y “al”, “Del” y “del”, “Parecía” y “parecía”, etc.

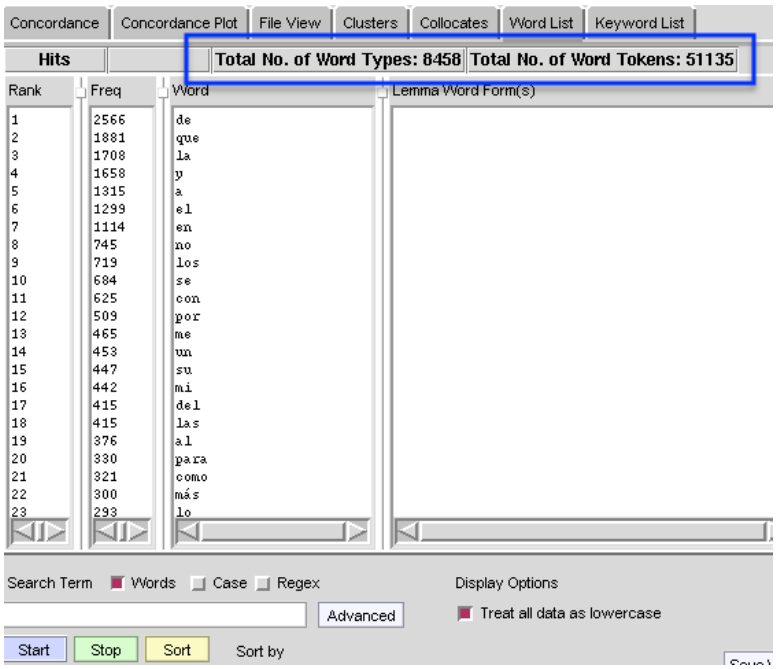


Figura 10: Frecuencias de palabras contando todas como minúsculas.

son ahora consideradas como el mismo *type*, por lo que hay menos. Además, para el caso de "el", ha pasado de tener 1190 apariciones a tener 1299. Se han sumado las apariciones de "el" y de "El".

El número de tokens no varía.

Activar esta casilla es una opción para la extracción de datos. Piensa qué objetivos tienes en tu análisis y si te es útil o no diferencias mayúsculas y minúsculas. Puede haber casos en que esta diferencia sea útil y pertinente. Por ejemplo, si quieres extraer datos de los nombres propios que utiliza Galdós en esta novela para sus personajes, la diferencia entre mayúsculas de minúsculas puede ser de gran utilidad.

5.4. Filtrar las *stopwords*

Si tomamos la lista de frecuencias de cualquier texto y las representamos en un gráfico, sale siempre la siguiente figura:

Esta figura indica que en todo texto hay unas pocas palabras que se repiten mucho (parte alta del gráfico a la izquierda), muchas palabras que se repiten pocas veces, y la gran mayoría de las palabras del corpus aparecen una vez (parte baja del gráfico a la derecha). Todo este conjunto de palabras

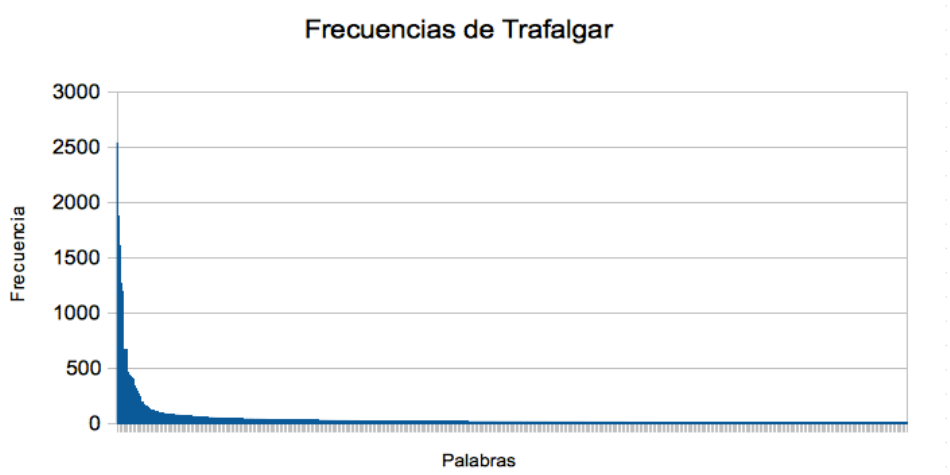


Figura 11: Gráfico de frecuencias léxicas.

que aparece sólo una vez en el corpus se denomina “Hapax Legomenon” (ἅπαξ λεγόμενον): dicho sólo una vez.

Para el análisis de frecuencias esta situación supone un problema serio. Queremos analizar las frecuencias de palabras, pero resulta que, por un lado, tenemos unas pocas palabras que se repiten mucho pero que no significan nada (*stopwords*), y por otro tenemos muchas palabras (la mayoría) que se repiten una sola vez.

Una forma de solucionar esto es mediante un filtro de *stopword*.

Una lista de *stopwords*, a veces traducidas por ”palabras de parada”, está formada por el conjunto de palabras con significado gramatical (sin significado léxico) que suelen aparecer con frecuencias muy altas en el corpus. En este grupo se consideran todas las palabras de categorías gramaticales cerradas (razón por la cual suelen tener mucha frecuencia en un corpus) como preposiciones, artículos, pronombres, determinantes, etc. Además se suelen incluir palabras de categorías abiertas con significado gramatical que también suelen tener mucha frecuencia y no significativas del corpus, como las formas del verbo “ser” o las formas del verbo “haber” en tanto que verbos auxiliares.

Físicamente, estas palabras se almacena en un fichero de texto sencillo (.txt, por ejemplo). Se puede crear a mano editando el fichero, o descargar alguno de los ficheros de *stopwords* de los muchos que que hay por la web.

En todo caso, como no es más que un fichero de texto, se puede modificar para dejar las palabras que NO queremos que aparezcan en la lista de frecuencias, sea cual sea su categoría gramatical. Esto ya es una decisión a

tomar por el investigador según los objetivos del análisis.

Calcular frecuencias con un filtro de *stopwords*

Para calcular las frecuencias del corpus ignorando estas palabras, debemos primero tener en nuestro disco local el fichero de texto con las *stopwords*. En segundo lugar se carga el fichero en AntConc y, por último, se repite el proceso de cálculo de frecuencias.

Para cargar el fichero de *stopwords* en AntConc:

1. En el menú de herramientas de AntConc, entra en "Tool preferences". Con esto se abre la ventana de preferencias de cada una de las herramientas de AntConc.

2. En esta ventana, seleccionar "Word List" para acceder a las preferencias del cálculo de frecuencias.

3. En el apartado "Word List Range Option", activa la casilla "Use a stoplist listed below".

4. Pulsa en el botón "Open" de "Ad Words From File" y seleccionamos el fichero con la lista de *stopwords*. Selecciona "Open".

   Si todo ha ido bien, aparecerá en la ventana la lista de palabras del fichero. Finalmente pulsamos en "Apply".

5. Repetimos el cálculo de frecuencias (botón "Start" desde "WordList").

En la Figura 13 tienes el resultado del mismo cálculo de frecuencias, pero con el filtro de stopwords.

Fíjate, primero cómo ha cambiado el número de *types* y de *tokens*.

Fíjate también en las palabras más frecuentes. Éstas sí son palabras con contenidos, son palabras que nos dan datos interesantes sobre el texto: las frecuencias léxicas empiezan a mostrarnos datos relevante.

Por ejemplo, la segunda palabra más frecuente es el pronombre personal de primera persona. Este dato es lógico, pues el narrador es el protagonista-testigo de la historia. En la octava posición aparece la palabra .^{es}cuadra", que tiene relación directa con la temática marítimo-militar de la novela. ¿Qué más conclusiones podrías extraer analizando las primeras 25 palabras más frecuentes de *Trafalgar*?

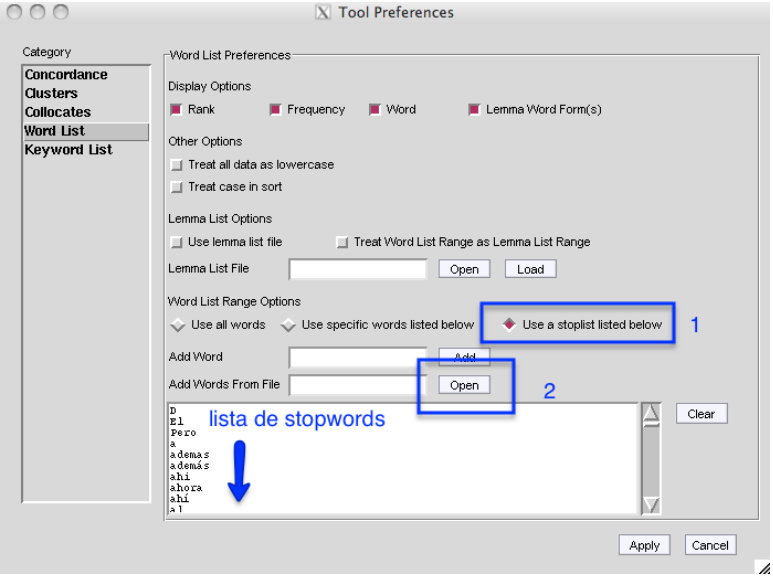


Figura 12: Método para cargar la lista de *stopwords*

5.5. Opciones de búsqueda en la lista de frecuencias

La interfaz de WordList tiene algunas funciones que ayudan a buscar en la lista de frecuencias y, así, facilitar el análisis.

Cambiar en el orden de la lista (*sort*)

La lista de frecuencias se presenta por defecto de la palabra más frecuente a la menos frecuente. Ese orden se puede cambiar. Para ello se utiliza la opción "Sort by". En el menú desplegable verás que la lista se puede ordenar por frecuencia (orden por defecto), alfabético ("by Word") o alfabético desde el final de palabra ("by Word End").

Selecciona el tipo de orden que quieras, y pincha en el botón "Sort". La lista se reordenará según lo indicado.

Ordenar la lista desde el final de la palabra puede ser de utilidad para ciertos análisis. Por ejemplo, en un corpus de poesía, puede ser útil para analizar las rimas frecuentes. O en cualquier corpus, para analizar formas verbales determinadas.

Con la opción "Invert order" se puede mostrar la lista en orden inverso: desde las palabras menos frecuentes hasta la más frecuente.

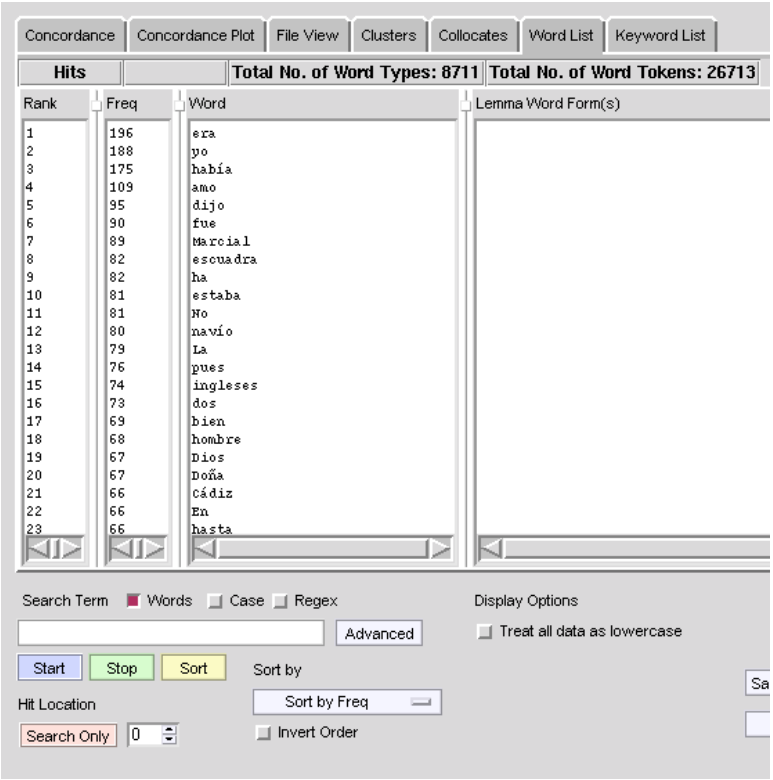


Figura 13: Frecuencias de palabras sin las *stopwords*

Figura 14: Cambiar el orden de la lista de frecuencias

Buscar palabras concretas

La lista de palabras resultante es larga, y en muchas ocasiones querremos analizar sólo las frecuencias de algunas de ellas.

Para no tener que ir repasando la lista entera hasta localizar las palabras que nos interesa, podemos utilizar el buscador. Introduce la palabra que quieras analizar, pincha en "Start" de nuevo, y el programa te mostrará en qué posición está la palabra que buscas y qué frecuencia tiene.

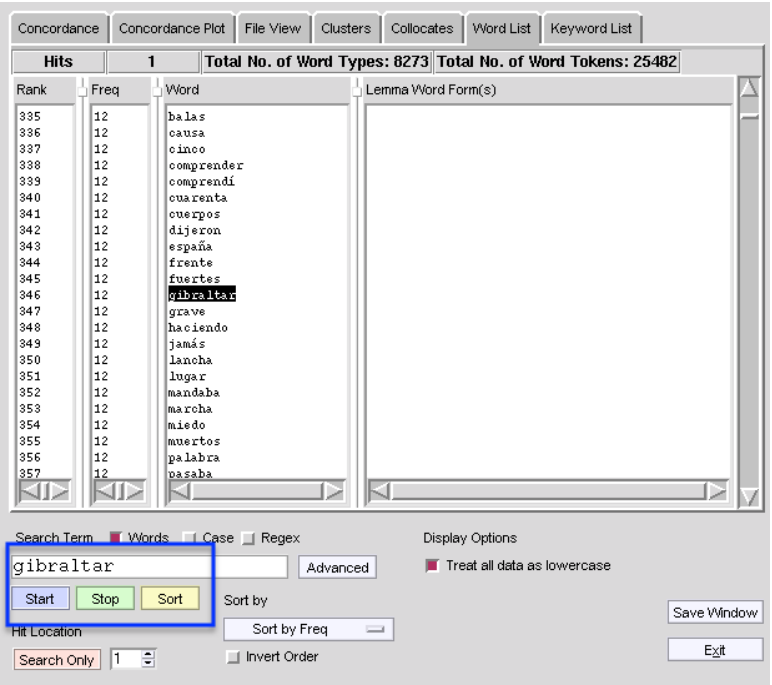


Figura 15: Posición de la palabra “Gibraltar”

Acceso a concordancias

Si pinchamos en una palabra de la lista de frecuencias, no llevará directamente a sus concordancias, donde podremos analizar todos los contextos de aparición de la palabra. En secciones siguientes se hablará más del análisis de concordancias.

5.6. Lematización

En sección anterior se planteó el problema de la variación morfológica: un análisis de frencuencias simple considera como palabras diferentes aquellas que tienen diferente variación morfológica. Así, este análisis de frecuencia no cuenta las veces que aparece el verbo çantar”, sino las veces que aparece çantar”(como infinitivo) por un lado, çantaría”por otro, çantaba”por otro, etc.

Para que el cálculo de frecuencias cuente todas las variantes morfológicas de una palabra como una misma palabra, es necesario lematizar el corpus.

La lematización es un proceso automático mediante el cual cada palabra del corpus (*token*) se relaciona con su forma no marcada, es decir, con su

forma canónica o lema (la forma que aparece en el diccionario). El lema de los verbos es la forma de infinitivo, de los nombres la forma en singular, y de los adjetivos la forma masculino singular.

Con un corpus lematizado, el cálculo de frecuencias contará todas las apariciones de un verbo como una única palabra, independientemente de su flexión. Y lo mismo con nombres y demás formas lingüísticas que tengan algún tipo de variación morfológica.

Lematización con AntConc

Para hacer cálculo de frecuencias sobre los lemas, es necesario lematizar el corpus. Para ellos se puede lematizar el corpus previamente con un lematizador[6].

Otra opción, más sencilla, es dejar que AntCon lematice el corpus. Para ello, necesitamos un fichero de texto donde cada palabra esté relacionada con su lema correspondiente. Para AntConc, este fichero debe tener el siguiente formato:

```
lema ->palabraFlexionada, palabraFlexionada, etc.
```

Por ejemplo:

```
casa ->casa,casas
listo->listo,lista,listos,listas
cantar->cantar,canto,cantas,canta,cantamos, cantaría, ...
etc.
```

Una vez dispongamos de este fichero, se debe cargar en AntConc de la siguiente manera:

1. Desde la barra de menú, seleccionamos “Tool Preferences”.
2. En la ventana de preferencias, pulsamos en ”WordList”
3. En “Lemma List Options”, seleccionamos la opción “Use lemma list file”.
4. Con el botón “open” seleccionamos el fichero con la lista de lemas.
5. Finalmente, se carga con el botón “Load”.

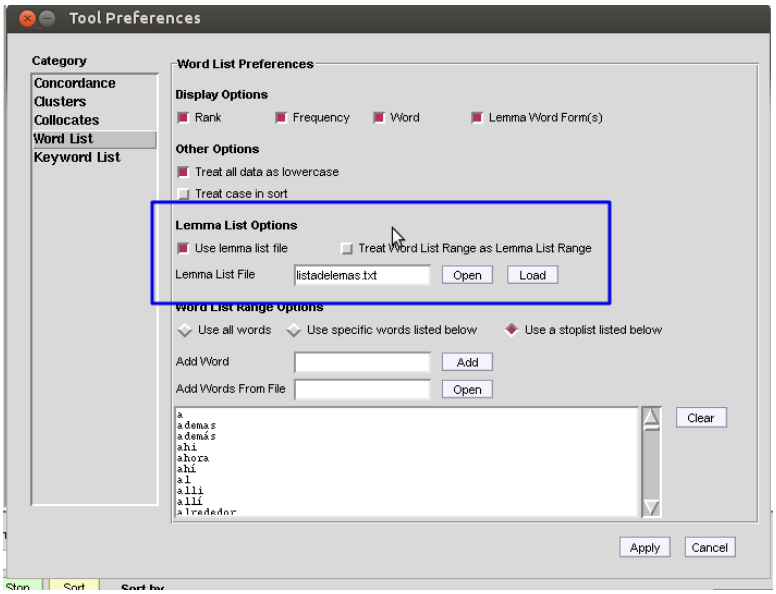


Figura 16: Proceso para cargar diccionario de lemas

Cargado el fichero con la lista de lemas, sólo queda volver a calcular frecuencias, tal y com se ha hecho anteriormente. La lista resultante ahora es diferentes:

Como se puede observar en la figura 17, aparece la lista de *Rank* y *Freq* como en casos anteriores. La lista de *lemmas* muestra las frecuencias, pero ya no por palabra , sino por lema. Obsérvese que ahora los verbos tienen muchas más frecuencias que antes. La razón es evidente: dado que se han contado todas la formas flexionadas de un mismo verbo como pertenecientes a la misma palabra, este verbo aparece ahora con mucha más frecuencia.

La cuarta columna (*lemma word form*) muestra la cantidad de apariciones de cada una de las formas flexionadas del lema correspondiente. Así, en la posición número 13 apare el lema "navío" con 138 apariciones, de las cuales 80 son singular y 58 plural.

Errores en la lematización

La lematización es un proceso hasta cierto punto complejo, pero el sistema de lematización de AntConc es muy simple: simplemente mira para cada token su lema en el diccionario. Sin embargo, muchas palabras son ambi-

[6]Para español, por ejemplo, se podría utilizar el analizador morfológico de Freeling (`http://nlp.lsi.upc.edu/freeling/`). Estos programas, sin embargo, no son sencillos de utilizar ni de instalar.

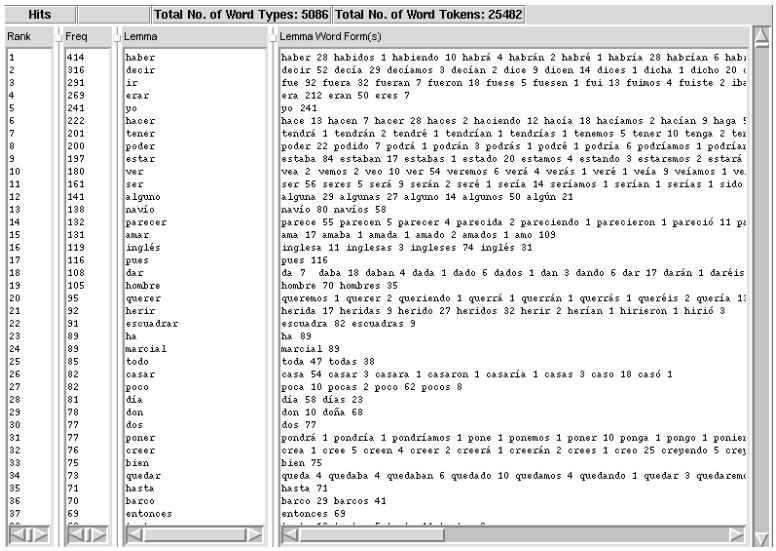


Figura 17: Frecuencias de palabras con el corpus lematizado

guas. Por ejemplo, ¿cuál es el lema de la palabra "amo", que tan frecuente es en *Trafalgar*?: ¿la forma verbal "amar" o la forma nominal "amo"?

AntConc no puede resolver esta ambigüedad. Sistemas más avanzados y complejos como Freeling sí, pero no vamos a entrar a estudiar el uso de estos programas.

Por todo ello, la lematización en AntConc se debe utilizar con mucho cuidado. Se debe asumir que habrá muchas palabras mal legalizadas y por tanto sus frecuencias serán erróneas. La solución a estos errores es analizar a mano la lematización y las frecuencias mediante la herramienta de concordancias, de la que se hablará en otra sección.

Si bien con los lemas el análisis de frecuencias mejora, no olvides nunca que **la lematización en AntCon siempre comete errores**.

5.7. Cálculo de la riqueza léxica

La riqueza léxica de un texto se refiere a la cantidad de palabras diferentes de las que está compuesto. Un texto con mucha riqueza léxica estará formado por muchas palabras diferentes con baja frecuencia; frente a un texto con poca riqueza léxica, que estará formado por un grupo reducido de palabras que se repiten mucho.

Calcula la riqueza léxica de un texto es relativamente sencillo con la fórmula llamada *type-token ration*. Basta con dividir el número de *types* del

texto por el número de *tokens*:

$$riquezaLexica = \frac{No.Types}{N.Tokens}$$

Dado que el número de *types* siempre será inferior al número de *tokens*, el valor resultante estará siempre entre 0 y 1. Cuanto más se acerque este valor a 1, más riqueza léxica tendrá el texto (es decir, el número de *tokens* se acerca al número de *types*). Cuanto más se acerque el valor a 0, más pobreza léxica tendrá el texto (el número de *tokens* se aleja del número de *types*).

Teóricamente, un valor 1 indica que en el texto hay el mismo número de *tokens* y de *types*: no se repite ninguna palabra.

Esta fórmula, sin embargo, tiene un problema: depende del tamaño del texto. Si el texto tiene una sola palabra, su riqueza léxica será 1. Cuanto más amplio sea el texto, más fácil será que la riqueza léxica baje[7].

Por eso, si se utiliza este valor para comparar la riqueza léxica de dos o más textos, hay que tener siempre presente que los textos deben tener el mismo tamaño. Si no lo tienen, los valore de riqueza léxica no son comparables.

6 Agrupamientos y *n-gramas*

Hasta ahora hemos analizado el corpus por las frecuencias de las palabras aisladas. El análisis de frecuencia se puede completar extrayendo datos sobre cómo se juntas las palabras, es decir, con qué frecuencia dos o más palabras aparecen juntas en el corpus.

Estos conjuntos de palabras que pueden aparecer juntas en un texto, con un orden consecutivo determinado, se llaman *n-gramas*, donde n es el número de palabras. Así tenemos:

bigramas: dos palabras consecutivas
trigramas: tres palabras consecutivas
tetragramas: cuatro palabras consecutivas
pentagramas: cinco palabras consecutivas
etc.

[7]Sobre éste y otros problemas del cálculo de la riqueza léxica, así como otras fórmulas alterativas, véase Baayen (2001) *Word Frequency Distribution* Dordrecht, etc. Kluwer Academic Publisher

Para calcular y analizar *bigramas* de un corpus con AntConc, sigue los siguientes pasos:

1. Pulsa en la pestaña Cluster. Con esto vamos a la herramienta de cluster o agrupamiento, dentro de la cual está el cálculo de *n-gramas*.

2. Activa la opción "N-Grams". Verás que el nombre de la pestaña ha cambiado.

3. Pulsa "Start".

Con esto AntConc nos muestra los bigramas más frecuentes del corpus.

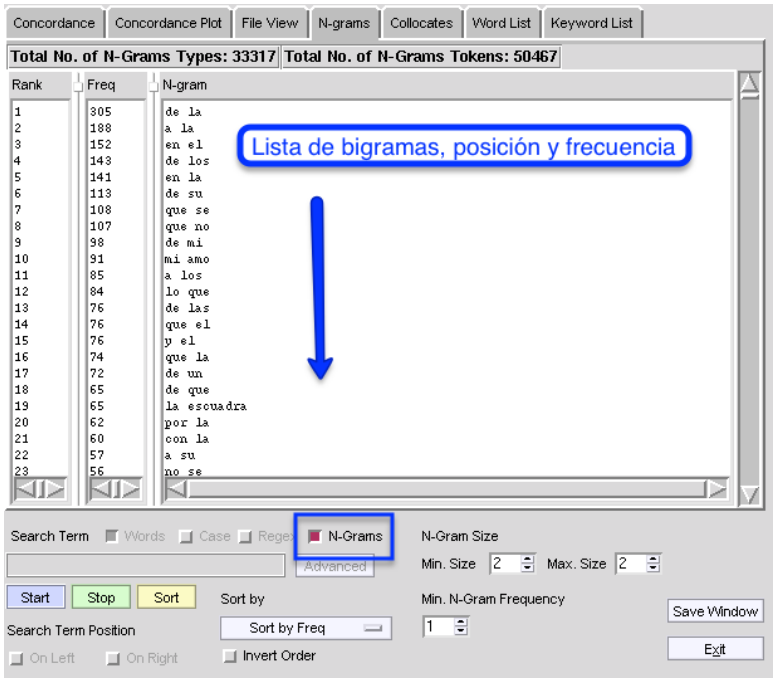


Figura 18: Cálculo de bigramas

Como se puede observar, para calcular n-gramas no se eliminan las *stop-words*. Dado que el concepto de *n-grama* es posicional (dos o más palabras que aparecen de manera consecutiva en el texto), no se pueden eliminar estas palabras.

Por ello, los bigramas más frecuentes están formados por palabras con significado gramatical: "de la", "a la", "en el". Tenemos que ir a la posición 10 para encontrar un bigrama interesante "mi amo". Y en la posición 19: "la escuadra".

Ambos bigramas son significativos de la novela. ¿Qué evidencias literarias se podría deducir de ellos?

Para calcular *n-gramas* de orden superior a dos (trigramas, tetragramas, pentagramas, etc.), indica en "N-Gram Size" el tamaño mínimo y máximo del *n-grama* que deseas conocer. Puedes ignorar los n-gramas de frecuencias bajas con "Min. N-Gram Frequency", donde se indica la frecuencia mínima de los *n-gramas* que deseas extraer. Todos los *n-gramas* que tengan una frecuencia inferior a ese valor serán ignorados.

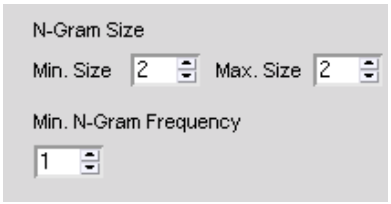


Figura 19: Cambio en el tamaño de *n-gramas*

7 Concordancias

Las concordancias son otra forma diferente de ver y analizar un texto: muestran todos los contextos en que aparece una palabra. Es una visión transversal del corpus: permite analizar cada uno de los contextos en los que aparecen las palabras del corpus. Es, por ello, una herramienta muy útil para analizar aspectos léxico y de vocabulario.

Los programas que muestran las concordancias de un texto también se denominan KWIC, que son las iniciales de *Key Word In Context*.

En AntConc, la herramienta de concordancias se encuentra en la pestaña "Concordance".

Para ver las concordancias de una palabra, sigue los siguientes pasos:

1. Pulsa en la pestaña Çoncordance"
2. En "Search Term" escribe la palabra de la que deseas conocer sus concordancias.
3. Pulsa el botón "Start".

En ese momento te aparecerá la típica pantalla de concordancias:

En este ejemplo hemos buscado la palabra "amo", que según la lista de frecuencias es uno de los nombres más frecuentes y presenta una clara ambigüedad entre la forma nominal y la verbal.

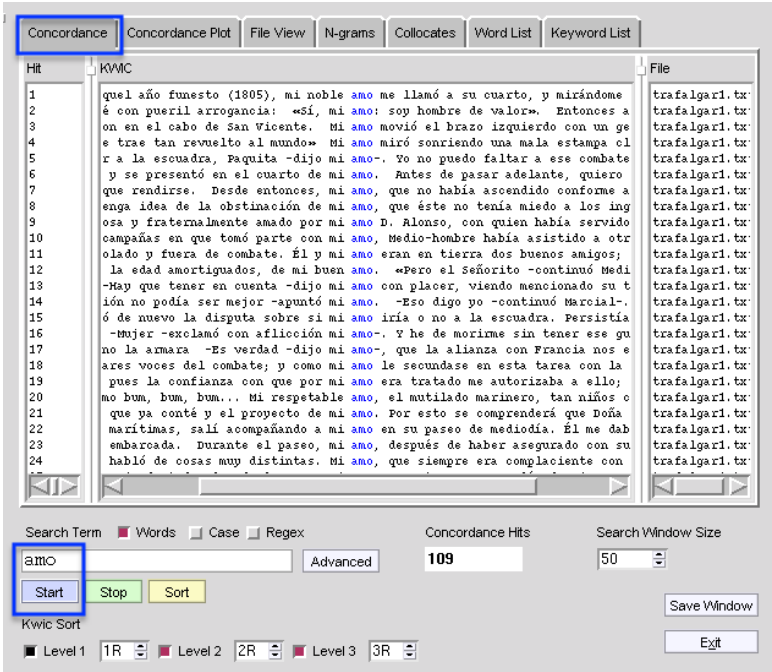


Figura 20: concordancias en AntConc

Como puedes ver en la imagen, la concordancias muestran la palabra objeto en el centro (en color azul) y a ambos lados el contexto en cada una los contextos donde aparece. De esta manera se puede analizar el uso concreto de esa palabra en el corpus.

Mediante el análisis de concordancias podemos concluir que, efectivamente, la palabra "amo" se utiliza en *Trafalgar* como nombre, no como verbo: de las 109 apariciones de "amo", se usa como nombre en 108 ocasiones, y sólo en una como verbo ("Yo amo a Dios y estoy tranquilo. Gabrielillo...").

El análisis ya es mucho más refinado. De esta manera, frente a los datos más o menos en bruto de las frecuencias y los *n-gramas*, las colocaciones permiten ir al análisis en detalle.

8 Keywords list: extracción de términos relevantes.

La opción *Keywords list* es muy útil para extraer los términos específicos de un corpus concreto.

Para ello, esta opción compara el corpus cargado en AntConc con otro corpus de referencia. Al compararlos, extrae aquellos términos que sean relativamente frecuentes en el corpus objeto de análisis y relativamente infrecuentes en el corpus de referencia.

Antes de utilizar esta opción debemos tener un corpus de referencia. En principio, debe ser un corpus genérico, representativo del idioma en general. Aquí puedes conseguir una muestra del corpus LesEsp:

`http://www.psico.uniovi.es/Dpto_Psicologia/metodos/soft/corpus/`

Una vez conseguido el corpus, el siguiente paso es cargarlo en AntConc. Como este corpus es sólo el corpus de referencia, debe cargarse de manera diferente al corpus objeto de estudio. Para cargar el corpus de referencia, ve a la opción *Tool Preferences* y de ahí a la opción *Keyword List*. Ahí verás la opción *Reference Corpus Options*. Con la opción *Add Directory* busca y selecciona la carpeta donde esté el corpus de referencia. Pulsa *aceptar* hasta volver a la página principal.

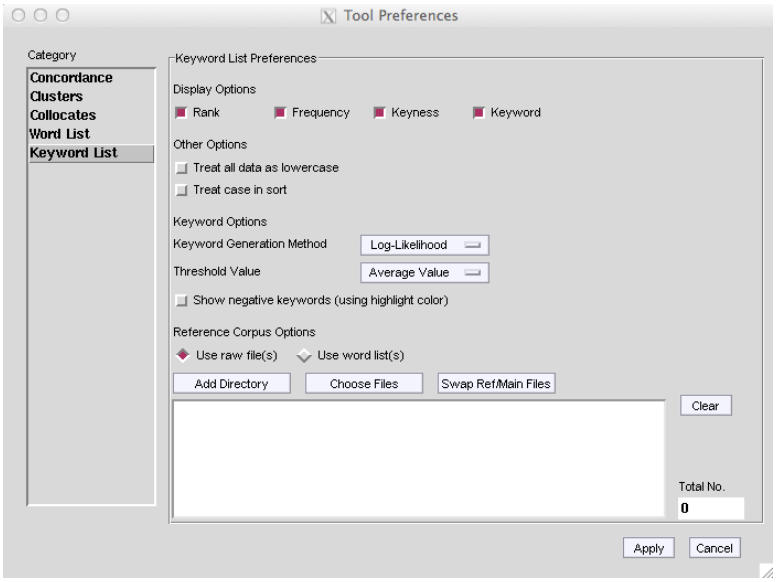


Figura 21: Opciones Keyword List

Ahora, desde *Keyword List* pulsa *start*. AntConc mostrará de nuevo una lista de palabras por frecuencias. En este caso aparecen primero las palabras más frecuentes del corpus que son infrecuentes en el corpus de referencia. De esta manera, las palabra específicas del corpus, aquellas que no están (o

lo están con poca presencia) en un corpus genérico, quedan patentes.

9 Expresiones regulares

Hasta ahora hemos visto dos formas de acceder a los datos de un corpus diferentes al acceso lineal: mediante listas de frecuencias (de palabras o *n-gramas*) y mediante concordancias (palabra clave en su contexto).

En ambos casos, nos situamos en un nivel superficial del corpus: se trabaja con "tokens", cadenas de caracteres separadas por espacio en blanco. A los más profundo que hemos llegado es a lematizar el corpus y extraer así frecuencias (no sin errores) de las formas canónicas no marcadas de las palabras.

Con las expresiones regulares vamos a dar un paso en profundidad de análisis, pero también en complejidad.

9.1. Definición

Para el ordenador, un texto es una cadena de caracteres: un conjunto consecutivo de caracteres. Todos los caracteres que forman un texto tienen una codificación determinada, incluyendo los espacios en blanco, los tabuladores o los cambios de línea.

Una expresión regular no es más que una forma de describir un conjunto de caracteres. Así, si queremos localizar en el corpus todas las apariciones de una determinada cadena, una determinada secuencia de caracteres, podemos hacerlo mediante expresiones regulares.

Una expresión regular es una fórmula textual: un lenguaje formal para expresar conjuntos de caracteres con diferentes niveles de complejidad. Así, por ejemplo, una expresión regular puede ser desde un simple:

```
cura
```

hasta una fórmula mucho más compleja como:

```
(?<!\bel\b)\s\bcura\b
```

La primera expresión regular detecta y muestra todas las apariciones de la cadena "`cura`" en un corpus. La segunda expresión regular detecta todas las apariciones de la cadena "`cura`" que no vayan precedidas de la cadena "`el`", es decir, detecta todas las formas verbales de la palabra "`cura`" (del verbo *curar*) e ignora las formas nominales (el nombre *cura*), asumiendo que

un artículo (en este caso, el artículo "`el`") nunca puede aparecer precediendo a un verbo. Siempre que aparezca "`cura`" precedido de "`el`", ese cura será nombre. Esta regla sintáctica es la que queda formalizada con la segunda expresión regular.

El funcionamiento de una expresión regular es relativamente sencillo: el ordenador detectará y mostrará todas las cadenas de caracteres de un corpus que emparejen exactamente con la expresión regular. Como todo lenguaje formal, las expresiones regulares deben ser muy precisas para que detecten exactamente lo que se busca, ni más cadenas ni menos.

Como se puede observar, las expresiones regulares pueden ser muy sencillas o muy complejas: depende de lo que busquemos. Dominar las expresiones regulares no es sencillo. Pero al mismo tiempo es una herramienta muy potente para extraer datos de grandes cantidades de corpus. Vale la pena aprender sus rudimentos.

9.2. Expresiones regulares en AntConc

En varias de las funciones de AntConc se pueden usar expresiones regulares: en aquellas en que aparezca la casilla "Regex". Si se activa la casilla, en vez de tomar la entrada como una palabra simple, la interpretará como expresión regular.

En esta sección utilizaremos sólo la función de "File View", que muestra los ficheros de texto tal cual.

9.3. Metacaracteres

Una expresión regular está formada en principio por la secuencia de caracteres que se quiere localizar en el corpus. En este sentido funcionan igual que un buscador simple.

Lo que hace de las expresiones regulares un recursos muy potente es la posibilidad de utilizar metacarateres. Los metacarateres son caracteres que se emparejan con varios caracteres. El Cuadro 2 muestra los principales metacaracteres:

Así, por ejemplo, si una palabra forma el plural añadiendo "s" a la forma singular, podemos detectar todas sus apariciones de esa palabra con independencia de su morfema de número con expresión regular como "`palabra?`":

casas? = casa casas.

.	cualquier carácter menos retorno de carro
*	el carácter anterior cero o más veces
+	el carácter anterior una o más veces
?	el carácter anterior cero o una vez

Cuadro 2: Principales metacaracteres

"`casa`" se empareja con los cuatro primeros caracteres por ser iguales. El último carácter "`s`" junto con el metacarácter "`?`" indica que este carácter "`s`" puede aparecer una vez o ninguna. Por eso, "`casas?`" se empareja tanto con "`casa`" como con "`casas`".

Si queremos detectar todas las formas flexionadas simples de un verbo regular, podemos utilizar una expresión regular tipo "`raízverbal.*`". Así:

cant.* = canto cantas canta cantamos etc.

La primera parte de esta expresión regular es "`cant`", que se empareja con esos mismos caracteres. La segunda parte "`.*`" significa cualquier carácter "`.`" repetido una o más veces "`*`". Con ello se está expresando que "`cant`" puede ir seguido de cualquier carácter repetido cero o más veces: es decir, de lo que sea.

Cuidado: esta expresión regular es muy genérica: se empareja con cualquier palabra que contenga `cant`. Junto a las formas flexionales del verbo "cantar" localizaría otras palabras como, por ejemplo, "cantamanañanas". En las siguientes secciones se explicará cómo limitar una expresión regular así.

De la misma manera, para detectar todos los adverbios que acaban en `-mente`, se podría utiliza una expresión regular como ésta:

.*mente = tranquilamente

9.4. Caracteres invisibles

Hay otro grupo de metacaracteres para representar caracteres invisibles. El Cadro 3 muestra los principales:

Para que las expresiones anteriores sean más precisas, sería mejor escribirla así:

\b.*mente\b
\bcant.*\b

\b	inicio o fin de palabra
\s	espacio en blanco
\t	tabulador
\n	salto de línea
\r	retorno de carro
^	principio de línea
$	fin de línea

Cuadro 3: Principales caracteres invisibles

De esta manera, limitamos el emparejamiento sólo a palabras.

Para representar dos o más palabras seguidas, habría que incluir en la expresión regular el carácter de espacio en blanco:

```
\sun\scura\s
```

9.5. Agrupaciones y alternancias

Las expresiones regulares también nos permiten trabajar con agrupaciones de caracteres. Para agrupar caracteres se utilizan los paréntesis. Utilizados junto al metacarácter disyunción "|", podemos crear expresiones regulares que detecten a la vez diferentes palabras o diferentes variantes de una misma palabra.

Por ejemplo, para representar que una palabra puede aparecer con la primera letra en mayúscula o minúscula, se puede crear una expresión regular así:

```
(C|c)asa
```

Si además queremos detectar esa palabra tanto en singular como en plural, podemos completar la expresión regular así:

```
(C|c)asa(|s)
```

El primer paréntesis expresa la posibilidad de que la primera letra sea mayúscula o minúscula, y el segundo la posibilidad de que acabe en "s" (plural) o no (singular). Si recuerdas lo comentado anteriormente, tenemos ya dos formas de expresar la presencia o ausencia de morfemas:

(C|c)asa(|s) = (C|c)asas?

La agrupación y disyunción de caracteres es útil también para detectar palabras de morfología irregular. Dado que en estos casos no hay un patrón de flexión ni derivación, sólo podemos acceder a todas las formas de una palabra irregular agrupando todas sus variantes. Por ejemplo, para detectar todas las formas de presente de indicativo del verbo "*caber*", podemos utilizar una expresión regular así:

(\b quepo \b | \b cab (é|e) i? (s|n|mos)? \b)

Cuidado: esta expresión regular detectaría el error ortográfico `cabeis`. Además no detectaría las apariciones del verbo con mayúscula.

Otra forma de agrupar caracteres es con los corchetes `[]`. Éstos permiten hacer rangos de caracteres. Así, una expresión como "`[A-Z]`" seleccionaría cualquier carácter alfabético ASCII (desde la "`a`" hasta la "`z`"). Para que seleccionaran cualquier carácter alfabético del español, habría que ampliar el rango así: "`[a-záéíóúñ]`".

Estos rangos permiten variaciones. Si queremos todos los caracteres alfabéticos incluyendo tanto mayúsculas como minúsculas, se puede establecer el rango "`[A-z]`"; y para español "`[A-záéíóúÁÉÍÓÚñN]`". Para seleccionar cualquier número de un dígito: "`[1-9]`".

Hay también metacaracteres que representan agrupamientos de este tipo. Los principales son:

\w	cualquier carácter alfanumérico (ASCII)
\d	cualquier número
\s	cualquier carácter invisible (espacio, tabulador, retorno carro)

9.6. Negación

Hay unidades lingüísticas que es más fácil detectarlas no por lo que son, sino por lo que no son, es decir, por negación.

Por ejemplo, para detectar las palabras de un corpus podemos utilizar una expresión positiva como "`\w`": todos los caracteres alfanuméricos. Esta expresión presenta el problema de los caracteres que no están en ASCII (eñes, tildes, etc.) Se puede hacer una expresión que detecte todas las

palabras del corpus con independencia del idioma: mediante un fórmula negativa. Definimos en este caso palabra como todo aquello que no sea espacio en blanco. No se indica lo que es, sino lo que seguro no es: espacio en blanco.

Para introducir negación en las expresiones regulares se utilizan los corchetes y el acento circunflejo ^. Todas las palabras de un corpus, según la definición negativa anterior, se podrían detectar simplemente con la expresión regular:

[^\s]

Muchos de los metacaracteres anteriores tienen su correlato negativo en mayúscula:

\W	todo lo que no sea carácter alfanumérico (ASCII)
\D	todo lo que no sea número
\S	todo lo que no sea carácter invisible

Estas dos expresiones son por tanto equivalentes: dos formas de expresar lo mismo:

[^\s] = \S

9.7. Búsquedas condicionales

Por último, vamos a ver cómo introducir condicionalidad en la búsqueda. Esta opción es muy útil porque permite representar el contexto de las palabras. Con esta opción podemos detectar determinadas palabras o cadenas de caracteres siempre y cuando aparezca algún elemento (condición) en su contexto anterior o posterior, pero sin que esa condición forme parte del resultado. El término técnico para la condicionalidad es *lookahead* y *lookbehind*.

En el primer caso, la condición contextual se sitúa después de la cadena de caracteres que se desea extraer. Para indicar la condición, se utiliza la expresión "`?=`".

Por ejemplo, si quisiéramos detectar todas las apariciones de la letra "`a`" que vayan seguida de la letra "`b`" (pero sin extraer la "`b`", sólo a "`a`"), se podría utilizar una expresión regular de este estilo:

a(? =b)

Se pueden introducir también condiciones negativas: detectar determinada cadena siempre y cuando en su contexto posterior no haya determinado elemento. Par expresar la negatividad se utiliza el símbolo de exclamación "`!`". Por ejemplo, para extraer todas las apariciones de la letra "`a`" que NO vayan seguidas de la letra "`b`", la expresión regular sería:

a(?!b)

Para situar la condición en el contexto anterior (*lookbehind*) se utiliza el carácter "menor que": "`<`". Así, "`?<=`" sería condición positiva y "`?<!`" condición negativa. Por ejemplo, para extraer todas las apariciones de la letra "`a`" que DELANTE tengan la letra "`b`", la expresión regular sería:

(? <=b)a

Veamos un ejemplo de cómo aplicar la condición a un problema lingüístico. Un fenómeno muy común en las lenguas es la ambigüedad categorial: palabras que son iguales en su forma pero diferentes en su categoría gramatical (y por tanto cambién en su significado). Así ocurre, por ejemplo, con el *token* "`cura`", que según el contexto puede ser nombre o verbo. Para saber la categoría gramatical de esas palabras se debe consular el contexto. Así, hay una regla en español que dice que después de artículo nunca aparecerá un verbo. Es una regla, como se ve, negativa. Para extraer de un corpus todos los casos en los que "`cura`" es verbo y no nombre, podemos crear una regla con condición negativa de este tipo:

(?<!\bel\b)\s\bcura\b

Esta expresión establece una condición negativa: delante de "`cura`" no puede aparecer el artículo "`el`". De esta manera, el resultado serían las apariciones de "`cura`" como verbo.

Esta expresión es incompleta, pues casos de "`cura`" antepuesto por otro tipo de artículo ("`un cura`", "`la cura`"), también sería detectados. Habría que completar la expresión con una agrupación de todos los tipos de artículo en español.

9.8. Práctica y más información

La mejor forma de dominar las expresiones regulares es practicar. En poco tiempo se asimila su lógica y se pueden crear expresiones muy potentes. Hay varias páginas para entrenarse con las expresiones regulares. Una buena página es ésta:

`https://regex101.com/`

Para más información sobre las expresiones regulares:

`http://www.regular-expressions.info/`