Jornadas CHARTA 2024. Herramientas y recursos digitales para el desarrollo de corpus históricos, 2024/2/29-3/1, Universidad de Alcalá.

Lematización de los datos de CODEA

Nuevos métodos

Hiroto Ueda (Universidad de Tokio)

1. Introducción

Agradezco, en primer lugar y de todo corazón, a la profesora Belén Almeida la invitación a impartir esta charla. En ella vamos a abordar tres temas de forma simultánea: el sistema Lyneal (*Letras y números en análisis lingüísticos*)[1] al servicio de investigaciones lingüísticas, los nuevos métodos de lematización de los datos de CODEA[2] y un ejemplo de la utilización del sistema y de los datos en un estudio sobre la posición del pronombre átono. El primero será un resumen del documento *Cómo usar Lyneal* situado en la sección de Información de la plataforma Lyneal, el segundo abordará los procesamientos de datos necesarios para construir nuestra base de datos y el tercero constituye un estudio reciente que hemos llevado a cabo con los nuevos materiales del corpus CODEA.

Para el análisis de los métodos de lematización, hemos utilizado los materiales propios de estudios anteriores: Sánchez-Prieto Borja, Pedro / Ueda, Hiroto. 2018, Sánchez-Prieto Borja, Pedro / Moreno Sandoval, Antonio / Ueda, Hiroto. 2020, Agujetas, María / Sánchez-Prieto Borja, Pedro / Ueda, Hiroto. 2022a, Agujetas, María / Sánchez-Prieto Borja, Pedro / Ueda, Hiroto. 2022b.

2. Procesamientos de datos

2.1. Datos anteriores

Para realizar la lematización del corpus, aprovechamos los datos

[1] https://h-ueda.sakura.ne.jp/lyneal/
[2] https://www.corpuscodea.es/corpus/consultas.php

anteriormente lematizados, puesto que la mayoría de las formas clave (C) y sus contextos inmediatos (A1, A2, P1, P2) aparecen de forma repetida. Para ver los contextos inmediatos, nos fijamos en las dos palabras inmediatamente anteriores (A1, A2) a la clave (C) y las dos palabras inmediatamente posteriores (P1, P2) a esta palabra clave. En total, obtenemos cinco palabras junto con la información gramatical correspondiente: Crit (forma crítica), Norm (forma normalizada), Lema y Cg (categoría gramatical), que trataremos en la sección 3, con el añadido de n (número de veces que ocurre la combinación de forma clave y formas de contexto). Estos son los datos de referencia (R: R1-R9):

A2	A1	C	P1	P2	Crit	Norm	Lema	Cg	n
por	la	gracia	de	D	gracia	gracia	gracia	n	499
,	por	la	gracia	de	la	la	el	ar	471
,	de	Toledo	,	de	Toledo	Toledo	Toledo	t	346
,	de	León	,	de	León	León	León	t	346
,	de	Córdova	,	de	Córdova	Córdoba	Córdoba	t	337

Tabla 1. Datos de referencia, R9, en orden descendente de n.

La mayoría de las veces, no se necesitan contextos para identificar los lemas y categorías de la forma clave. Pero también existen numerosos casos de homógrafos, por ejemplo, *vino* (nombre y verbo), *fuera* (adverbio, verbos 'ser', 'ir'), *la* (artículo, pronombre), etc. Para desambiguar estos casos polivalentes, necesitamos observar sus contextos.

El programa, sin conocer la gramática, utiliza solo la información de las cinco palabras (A2, A1, C, P1, P2). Cuando las cinco palabras coinciden en los datos de trabajo y los datos de referencia, el programa agrega a los datos de trabajo la información lingüística (Crit, Norm, Lema, Cg) que hay en los datos de referencia. Cuando no coinciden, el programa intenta buscar los contextos cada vez más reducidos (A2, A1, C, P1), (A1, C, P1, P2), (A1, C, P1), ..., y así sucesivamente. De esta manera, se realizan las siguientes búsquedas de coincidencias. Por ejemplo, R9 se refiere a los datos de referencia con 5 columnas coincidentes (A2, A1, C, P1, P2) y R8 a los datos de referencia con 4 columnas coincidentes (A2, A1, C, P1):

1. R9: (A2, A1, C, P1, P2)
2. R8: (A2, A1, C, P1)

3. R7: (A1, C, P1, P2)
4. R6: (A1, C, P1)
5. R5: (A2, A1, C)
6. R4: (C, P1, P2)
7. R3: (A1, C)
8. R2: (C, P1)
9. R1: (C)

Las primeras búsquedas devuelven soluciones acertadas y la últimas pueden equivocarse, especialmente la última 9. R1: (C), que es la búsqueda solo de la forma clave (C) sin contexto.

A continuación, por medio de la función MakeRef, obtenemos los datos de referencia: R9, R8, ..., R1[3]:

R = MakeRef(D1), donde D1 es datos anteriores

Los datos anteriores deben estar en la siguiente forma:

*	A2	A1	C	P1	P2	Crit	Norm	Lema	Cg
1	*	*	Coñocida	cosa	sea	coñocida	conocida	conocer	pa
2	*	Coñocida	cosa	sea	a	cosa	cosa	cosa	n
3	Coñocida	cosa	sea	a	todos	sea	sea	ser	v
4	cosa	sea	a	todos	los	a	a	a	p
5	sea	a	todos	los	que	todos	todos	todo	id

Tabla 2. Datos anteriores

La función MakeRef devuelve una lista de datos: R9, R8, ..., R1. Ya que hemos visto el primer fichero R9 (Tabla 1), veamos R8 y R1:

[3] Los programas utilizados están reunidos en la parte final de este estudio. Utilizamos el lenguaje R (R Core Team, 2023).

A2	A1	C	P1	Crit	Norm	Lema	Cg	n
V	.	S	.	*	*	*	ag	1248
días	del	mes	de	mes	mes	mes	n	942
de	la	dicha	villa	dicha	dicha	decir	pa	603
de	Santa	María	de	María	María	María	np	524
por	la	gracia	de	gracia	gracia	gracia	n	505

Tabla 3. Datos de referencia, R8, en orden descendente de n.

C	Crit	Norm	Lema	Cg	n
de	de	de	de	p	149734
e	e	y	y	c	74858
que	que	que	que	cp	73386
y	y	y	y	c	61876
en	en	en	en	p	50577

Tabla 4. Datos de referencia, R1, en orden descendente de n.

Naturalmente, cuanto menor es el número de los contextos, incluida la forma clave, el número de ocurrencia (n) aumenta. Aprovechamos esta información de frecuencia a la hora de analizar los datos de trabajo. Concretamente, averiguamos con más atención los resultados de combinaciones poco frecuentes.

2.2. Palabras clave

A modo de ejemplo, utilizamos un texto de CODEA-0001, que está constituido por el identificador, Id = 1, y el texto, Tx:

Id	Tx
1	Coñocida cosa sea a todos los que esta carta vieren cómo yo don Ferrando, ...

Tabla 5. Corpus CODEA-0001

Vamos a convertir este texto en una matriz de tres columnas por medio de la función Text2Word:

```
D = Text2Word(D)
```

N	Id	Tx
1	1	Coñocida
2	1	cosa
3	1	sea
4	1	a
5	1	todos

Tabla 6. Matriz de tres columnas, parte inicial[4]

donde N: número secuencial, identificador de la forma, Id: identificador del documento, Tx: texto.

El segundo paso es agregar a la Tx, convertido en forma clave (C), dos palabras anteriores (A1, A2) y dos palabras posteriores (P1, P2). Por ejemplo, A1 va a ser un vector de palabras que es un asterisco (*) más el vector de la palabra clave: "*, Coñocida, cosa, sea, a, todos, ...". A2, P1, P2 van a presentarse de manera parecida, con diferencia en la parte inicial:

A1: *, Coñocida, cosa, sea, a, todos, ...
A2: *, *, Coñocida, cosa, sea, a, todos, ...
P1: cosa, sea, a, todos, ...
P2: sea, a, todos, ...

Para conseguir estas columnas, hemos preparado el programa Adjacent

```
A1=Adjacent(D[,3],1) #Forma anterior-1
A2=Adjacent(D[,3],2) #Forma anterior-2
P1=Adjacent(D[,3],3) #Forma posterior+1
P2=Adjacent(D[,3],4) #Forma posterior+2
```

Formamos un marco de datos ('data.frame', D) de la siguiente manera:

```
D = data.frame(A2,A1,Clave=D[,3],P1,P2,D[,1:2])
```

[4] De aquí en adelante, nos limitamos a mostrar solo la parte inicial de los datos. en cuestión.

A2	A1	C	P1	P2	N	Id
*	*	Coñocida	cosa	sea	1	1
*	Coñocida	cosa	sea	a	2	1
Coñocida	cosa	sea	a	todos	3	1
cosa	sea	a	todos	los	4	1
sea	a	todos	los	que	5	1

Tabla 7. Datos de trabajo

2.3. Combinación

Ahora que tenemos dos datos: datos de trabajo (D) y datos de referencia (R), procedemos a producir los datos de correspondencia, donde se combinan las informaciones de ambos datos. Para ello, utilizamos la función AddInf:

D = AddInf(D, R)

A2	A1	Clave	P1	P2	N	Id	Crit	Norm	Lema	Cg	n
*	*	Coñocida	cosa	sea	1	1	coñocida	conocida	conocer	pa	30
*	Coñocida	cosa	sea	a	2	1	cosa	cosa	cosa	n	30
Coñocida	cosa	sea	a	todos	3	1	sea	sea	ser	v	14
cosa	sea	a	todos	los	4	1	a	a	a	p	43
sea	a	todos	los	que	5	1	todos	todos	todo	id	9

Tabla 8. Datos combinados

2.4. Asignación, desambiguación y revisión

Aproximadamente una décima parte de todo el corpus son nuevas formas y, por ello, están en blanco en las columnas de Crit, Norma, Lema y Cg, por no poseer información previa sobre ellas. De modo que tenemos que realizar un inmenso trabajo manual de asignación. No obstante, no efectuamos la asignación ni en el orden alfabético, ni en el orden de aparición en el texto, sino con los datos clasificados por la categoría gramatical de las palabras en contextos anteriores y posteriores.

A pesar de que no se ofrecen las categorías gramaticales de las nuevas formas, tenemos la información gramatical de la mayoría de las formas clave,

que podemos aprovechar para conocer las categorías gramaticales de las formas que hay en los distintos contextos. Especialmente el contexto de la palabra inmediatamente anterior (A1) es importante. Por esta razón, preparamos otro programa parecido a las formas de contextos inmediatos (A1, A2, P1, P2). Ahora se tratan las categorías de estas formas: A1c, A2c, P1c, P2c.

Para ello, vamos a preparar las categorías anteriores, A1c y A2c, y las posteriores, P1c y P2c, por medio de la función Adjacent:. Para aplicar Adjacent, los datos deben estar ordenados de manera ascendente. Por ello, trasladamos N e Id que están en las columnas 6 y 7 en el inicio[5], columnas 1 y 2 y reordenamos los datos por la columna 1, con la función SortM:

```
D=D[,c(6:7,1:5,8:12)]; D=SortM(D) # Trasladar y ordenar
```

Ahora preparamos las columnas de categorías gramaticales de formas adyacentes, utilizando la categoría gramatical de la clave, que está en la columna 11:

```
A1c=Adjacent(D[,11],1) # Cg anterior-1
A2c=Adjacent(D[,11],2) # Cg anterior-2
P1c=Adjacent(D[,11],3) # Cg posterior+1
P2c=Adjacent(D[,11],4) # Cg posterior+2
```

Para asignación, desambiguación y revisión, necesitamos no solamente contextos léxicos inmediatos, sino también contextos secuenciales largos, constituidos por 8 palabras anteriores a A2 y 8 palabras posteriores a P2. Vamos a construir los contextos secuenciales anteriores (Ant) y posteriores (Pos) por medio de la función Context, utilizando la columna de Clave (=5):

```
Ant=Context(D[,5],1) # Contexto anterior, 20 segundos
Pos=Context(D[,5],2) # Contexto posterior, 20 segundos
```

Por otra parte, para realizar la asignación, desambiguación y revisión de datos fijándonos en la parte final de formas (flexiones y sufijos), es

[5] Para procesoss posteriores, también es conveniente que las columnas de N e Id estén en la parte inicial del marco de datos.

conveniente utilizar la lista de las formas en orden inverso. Por ejemplo, para la forma 'coñocida', utilizamos 'adicoñoc', donde las letras constituyentes están alineadas en orden inverso. Por lo tanto, preparamos un programa Reverse y lo aplicamos a las columnas de formas críticas (Crit > Rc), formas normalizadas (Norm > Rn) y lemas (Lema > Rl):

```
Rc=Reverse(D[,8])
Rn=Reverse(D[,9])
Rl=Reverse(D[,10])
```

Finalmente, vamos a reunir todas estas columnas en un marco de datos y lo reordenamos en orden conveniente:

```
D=data.frame(D,A2c,A1c,P1c,P2c,Ant,Pos,Rc,Rn,Rl)
D=D[,c(1,2,18,14,3,15,4,5,8:11,16,6,17,7,19:22,12:13)]
```

De esta manera, los datos D poseen 22 columnas, que mostraremos dividido en tres partes:

1.N	2.Id	3.Ant	4.A2c	5.A2	6.A1c	7.A1	8.Clave
1	1	* * * * * * * *	*	*	*	*	Coñocida
2	1	* * * * * * * *	*	*	pa	Coñocida	cosa
3	1	* * * * * * * *	pa	Coñocida	n	cosa	sea
4	1	* * * * * * * Coñocida	n	cosa	v	sea	a
5	1	* * * * * * Coñocida cosa	v	sea	p	a	todos

Tabla 9. Datos completos. Columnas 1:8.

*	9.crit	10.norm	11.lema	12.cg	13.P1c	14.P1	15.P2c	16.P2
1	coñocida	conocida	conocer	pa	n	cosa	v	sea
2	cosa	cosa	cosa	n	v	sea	p	a
3	sea	sea	ser	v	p	a	id	todos
4	a	a	a	p	id	todos	ar	los
5	todos	todos	todo	id	ar	los	cp	que

Tabla 10. Datos completos. Columnas 9:16.

*	17.Pos	18.Rc	19.Rn	20.Rl	21.n	22.g
1	a todos los que esta carta vieren cómo	adicoñoc	adiconoc	reconoc	1	9
2	todos los que esta carta vieren cómo yo	asoc	asoc	asoc	1	9
3	los que esta carta vieren cómo yo don	aes	aes	res	14	9
4	que esta carta vieren cómo yo don Ferrando	a	a	a	43	9
5	esta carta vieren cómo yo don Ferrando ,	sodot	sodot	odot	9	9

Tabla 11. Datos completos. Columnas 17:22

Las últimas dos columnas, 21.n (número de ocurrencia, frecuencia) y 22.g (grado de coincidencia: coincidencia mínima [1] – coincidencia máxima [9]) son útiles para evaluar la fiabilidad de asignaciones automáticas.

2.5. Inventario léxico

Una vez elaborados los datos completos, procedemos a preparar el inventario léxico, objeto de estudio. Para ello, sobran contextos secuenciales, formas inversas e informaciones de número de ocurrencia (n) y grado de asignación (g) y faltan variables extralingüísticas: tiempo (A.100, centuria), espacio (Rg. región) y ámbito documental (Ámbito). Lo podemos realizar por medio de la función VLookUp[6]:

*	1.ID	2.A.100	3.Rg	4.Ámbito
1	1	1200	AN	Cancilleresco
2	2	1200	AN	Cancilleresco
3	3	1200	AN	Cancilleresco
4	4	1200	CV	Cancilleresco
5	5	1200	CV	Cancilleresco

Tabla 12. Variables extralingüísticas, A.

[6] VLookUp es parecido a la funci/n Excel VLOOKUP, con diferencia de que permite asignaciones con múltiples columnas.

D = VLookUp(D[,c(1:2,4:16)], A, 2, 1)

*	1.N	2.Id	3.A2c	4.A2	5.A1c	6.A1	7.Clave	8.Crit	9.Norm
1	1	1	*	*	*	*	Coñocida	coñocida	conocida
2	2	1	*	*	pa	Coñocida	cosa	cosa	cosa
3	3	1	pa	Coñocida	n	cosa	sea	sea	sea
4	4	1	n	cosa	v	sea	a	a	a
5	5	1	v	sea	p	a	todos	todos	todos

Tabla 13. Inventario léxico (primera parte, columna: 1-9)

*	10.Lema	11.Cg	12.P1c	13.P1	14.P2c	15.P2	16.A.100	17.Rg	18.Ámbito
1	conocer	pa	n	cosa	v	sea	1200	AN	Cancilleresco
2	cosa	n	v	sea	p	a	1200	AN	Cancilleresco
3	ser	v	p	a	id	todos	1200	AN	Cancilleresco
4	a	p	id	todos	ar	los	1200	AN	Cancilleresco
5	todo	id	ar	los	cp	que	1200	AN	Cancilleresco

Tabla 14. Inventario léxico (segunda parte, columna: 10-18)

3. Criterios lingüísticos

En esta sección, explicamos los criterios lingüísticos que adoptamos en el proceso de lematización.

Nos corresponde la asignación de formas críticas, formas normalizadas, lemas y categorías gramaticales. Las formas críticas respetan básicamente las claves, que aparecen en la presentación crítica, con la distinción de mayúsculas y minúsculas, acentuación gráfica, algunas representaciones gráficas críticas, puntuaciones, etc. (Sánchez-Prieto Borja, 2011). En nuestra forma crítica, mantenemos la misma presentación crítica original, menos la distinción de mayúscula y minúscula. Mantenemos la mayúscula original en los nombres propios (nombre de persona, apellido y topónimos) y números romanos, mientras que las mayúsculas que aparecen en el inicio de oración, títulos, apodos, instituciones las convertimos en minúsculas.

3.1. Totalidad

Tenemos 46 categorías gramaticales: a, ab, ag, ap, ar, av, c, cl, cl.cl,

cp, d, e, g, g.cl, g.cl.cl, id, if, if.cl, if.cl.cl, ij, ip, ip.cl, ip.cl.cl, ir, l, n, na, nm, np, nr, p, p.ar, p.pn, pa, pn, ps, pt, pv, pv.cl, pv.cl.cl, q, r, t, v, v.cl, v.cl.cl. Son abreviaturas, que se entienden de la siguiente manera; a: adjetivo, ab: abreviatura, av: adverbio, ag: agregar, ap: apellido de persona, ar: artículo, cl: clítico (pronombre átono), cp: complementante, c: conjunción, d: demostrativo, e: eliminar, g: gerundio, id: indefinido, if: infinitivo, ij: interjección, ir: interrogativo, l: latín, np: nombre de persona (nombre de pila), n: nombre (sustantivo), nm: numeral, na: número arábigo, nr: número romano, pa: participio adjetival, pv: participio verbal, ps: posesivo, p: preposición, pn: pronombre tónico, pt: puntuación, q: quitar (forma no identificada), r: relativo, t: topónimo, v: verbo finito.

3.2. Adjetivo

Hacemos la distinción entre adjetivo y nombre de acuerdo con la función de sustantivación por el artículo. De modo que los adjetivos se convierten en nombre detrás de artículo. Los adjetivos exentos del artículo los consideramos como adjetivos, menos los casos evidentes de nombres, como el nombre de la lengua (*portugués*), personas (*ciego, difunto, pobres: amparo de pobres*), construcción (*catedral*), etc. *Persona inocente* es adjetivo, mientras que *los inocentes* es nombre.

3.3. Diminutivo, aumentativo, superlativo

Los diminutivos (-ito, -illo, -ico), aumentativos (-ón) y superlativos (-ísimo) los integramos en el lema base, por ejemplo, *chiquillo* en lema <chico>. Las formas lexicalizadas las tratamos como lemas independientes del lema base, por ejemplo, <bolsillo>.

3.4. Un, uno, una

Hacemos la distinción de *un, uno, una, unos, unas* entre artículo, indefinido y numeral, según el caso. Los sustantivados son indefinidos, mientras que los determinantes antepuestos al nombre son artículos. Cuando se cuentan el número en el repertorio de bienes, por ejemplo, en *una perla y dos pendientes de oro, una* es numeral. También es numeral en: *un alcalde e dos regidores*. También lo es delante de unidad: cada *un año, un doblón*.

3.5. Latín

Excluimos las formas que aparecen en el contexto en latín por no mezclar las formas castellanas y las latinas. Por ejemplo, consideramos *nullus ricohome*, como forma latina, por situarse detrás de *nullus*. La forma latina o latinizante dentro del contexto castellano la consideramos como forma latina, por ejemplo, *con mi mano escripsi*. Dentro de una misma oración pueden aparecer las expresiones latinas y las castellanas, por ejemplo: *In Dei nomine notum sit que yo don Domingo* ..., donde hacemos la distinción de lengua en cada parte correspondiente.

3.6. Nombre propio

En los nombres propios, distinguimos nombre de persona (np), apellido (ap) y topónimo (tp). La misma forma puede pertenecer tanto al nombre de persona como al apellido. Los clasificamos considerando su posición relativa. Los topónimos utilizados como apellidos los consideramos como apellido, aunque tras *de: Francisco Aguilar, Marco de Aguilar*.

3.7. Participio

Hacemos la distinción entre participio verbal (pv), participio adjetival (pa) y participio nominal (n), que pertenecen a verbo, adjetivo y nombre, respectivamente. Los reconocemos como participios verbales únicamente cuando se utilizan con *haber*: *á dicho, dado avedes*, etc., que no flexionan en género y número, excepto algunos casos iniciales no gramaticalizados plenamente, que pertenecen al participio adjetival. Los participios con flexión de género y número son participios adjetivales, que pertenecen a la categoría de adjetivo, incluidos los usos pasivos. Los participios nominales, *abogado, delegado, bordado, cuidado*, son nombres.

3.8. Puntuación

Las puntuaciones (pt) las respetamos de la misma manera que las palabras, puesto que son importantes condiciones que determinan algunos usos de formas lingüísticas, por ejemplo, la enclisis de pronombres átonos en la posición inicial de oración.

3.9. Agrupación máxima

A la hora de abordar los estudios comparativos, conviene preparar el lemario con el principio de máxima agrupación, es decir, reunir las formas en el mayor grado posible dentro del mismo lema. Por ejemplo, hemos reunido los diminutivos, aumentativos y superlativos dentro de lemas de base correspondientes: *cucharón* → <cuchara>; *santísima* → <santo>. También lo hacemos con los verbos + clíticos; *detúvose* → <detener>, prefijos y sufijos en formas no lexicalizadas: *exinfante* → <infante>. Las formas femeninas de nombres también las incluimos en el lema de base: *abad, abadesa* → <abad>, *hijo, hija* → <*hijo*>.

3.10. Formas antiguas

Marcamos con un asterisco (*) las formas antiguas desaparecidas en el español actual, tanto en la forma normalizada como en el lema, por ejemplo *maguer**, *otrosí**, *crebantar** en <quebrantar>.

4. Final

En la lingüística de corpus, la herramienta habitual son las líneas de concordancia con una palabra clave en el centro rodeada a ambos lados de contexto (ing. K*ey Word In Context*, *KWIC*, Etxeberría, García, Gil y Rodríguez, 1995: 154-156, Barnbrook, 1996: 68-85, Halliday, Teubert, Yallop and Čermáková, 2005: 133-139, McEnery and Hardie, 2012: 35-48). En su lugar, creemos que también es posible pensar en palabras clave en su contexto (*Key Words in Context*). Nuestro sistema Lyneal ofrece la posibilidad de tres palabras clave[7]:

Contexto anterior (A).	Ctx inmediato anterior	Forma	Ctx. pos. inm.	Contexto posterior (P)
nasiados problemas tenemos como para	para	fabricarnos	otros.	otros. Ellos que van a la escuela y al coleg
que se conoce enseguida porque tiene la	la	fachada	de	de ladrillo. Y de ahí al "Campeón", nada, d
. Llegaban al arco contrario con bastante	bastante	facilidad	.	. Hacían los pases con mucha precisión y
'distinta... $ – ¡Ah, la tortilla de patatas! ¡Es	¡Es	facilísima	!	! ¡Y muy rápida de hacer! Pues mira, sólo t
souer? Cuéntame, cómo se hace. $ – Es	Es	facilísimo	.	. Le echas una medida de jugo de limón, c
agar) $ – Buenas tardes señora, ¿boleta o	o	factura	?	? $ – Boleta, nomás.

En este estudio, hemos ampliado la zona central de concordandancia

[7] https://h-ueda.sakura.ne.jp/lyneal/varitex.htm

hasta un límite de cinco palabras clave, considerando su utilidad, tanto en la preparación de los datos como en el análisis de estos.

Como hemos visto, las categorías gramaticales de las formas adyacentes se preparan a partir de las de la única palabra clave en la línea de concordancia. Por lo tanto, siempre trabajamos con la categoría gramatical y hacemos caso omiso de las de formas adyacentes. No obstante, también nos fijamos en las categorías gramaticales de formas adyacentes, puesto que son ellas las que suelen condicionar la asignación gramatical de la palabra clave.

Lo mismo puede decirse de las formas adyacentes que dan pistas a la hora de asignar formas normalizadas, lemas y categoría gramatical de la palabra clave. Para ello, utilizamos las columnas del orden inverso de cada forma (forma crítica, forma normalizada y lema).

De esta manera, para analizar las palabras clave centrales, observamos las palabras adyacentes y sus categorías y las últimas se reproducen a partir de las correspondientes a las palabras clave. Por lo tanto, la relación entre las dos es interdependiente y circular, y el trabajo de los analistas también lo es. De este modo, realizamos la revisión de los datos varias veces cambiando las categorías de la palabra clave con un procesamiento automático de reproducir las catergorias de formas adyacentes y haciéndolas cada vez más renovadas y correctas. Tras realizar múltiples revisiones sistemáticas, llegamos a conseguir que los datos sean lo más fiables posible. Nuestro trabajo es semiautomático en el sentido de que combinamos las operaciones mecánicas y las manuales, a diferencia de las plataformas automáticas existentes (Gómez Díaz, 2005).

El carácter circular también lo mantenemos en la serie de trabajo del inventario léxico, citada en la Introducción. Cada vez que realizamos el mismo trabajo aprovechamos los datos anteriores, así como la experiencia previa, y el trabajo actual va a ser la base del próximo proceso de lematización (Ueda / Sánchez-Prieto Borja / Moreno Sandoval, 2020). Sin embargo, el trabajo, lejos de considerarse repetitivo, es el mejor posible, pues avanzamos hacia datos cada vez mayores[8], cada vez mejores. Por lo tanto, la circularidad no significa necesariamente la repetición, sino la innovación en forma de espiral ascendente.

Los criterios lingüísticos todavía no son definitivos. Nunca lo pueden ser, puesto que siempre se presentan mejores soluciones. Entre varios

[8] Actualmente, n el corpus CODEA a contamos con dos millones y medio de formas (2.528.259 formas).

colaboradores, hemos discutido mucho durante estos meses sin llegar todavía a una conclusión decisiva. Afortunadamente, los datos no están impresos en papel en forma del diccionario, sino que son digitales en forma de base de datos. Podemos cambiar los criterios lingüísticos en cada etapa. Se reeditan trabajos, pero con mejoras constantes. Nuestra idea es ofrecer estos datos de la mejor manera posible a investigadores interesados.

Programas en R

A continuación, ofrecemos los códigos de funciones utilizadas en la sección 2. Está en el lenguaje R con paquetes indicados por *library*.

4.1. Datos de referencia

```
MakeRef=function(D){
  Ft=function(E){
    library(dplyr) #count, distinct
    print('1/3'); nc=ncol(E); E=count(E,E[,1:nc]) #UniqSum
    print('2/3'); A=c(rep(F,nc-1),T); E=SortM(E,1:nc,A)
    print('3/3'); E=distinct(E,E[,1:nc], .keep_all=T)
    print('sort'); E=SortM(E,nc+1,F)
  }; Li=list(); Col=list(1:5,1:4,2:5,2:4,1:3,3:5,2:3,3:4,3)
  for(i in 1:9){
    print(paste('File:',i)); Li[[i]]=Ft(D[,c(Col[[i]],6:9)])
  }; Li
} #IL
```

4.2. Del texto a las palabras

```
Text2Word=function(D){
  Trim=function(str) gsub('^([[:space:]]+)|([[:space:]]+)$','',str)
  A=D[,2]; A=gsub('([^0-9a-zñçáéíóúüçâòôàèì&êë&<>¥r¥n-])',' ¥¥1 ',A,T)
  A=Trim(A); A=gsub(' {2,}',' ',A,T); D[,2]=A
  library(tidyr); W=separate_rows(D, Tx, sep = " ")
  data.frame(N=1:nrow(W),W)
}
```

4.3. Información gramatical

```
AddInf=function(X,RR){
  SS=function(W,U,Col){
    Combine=function(D,C=NULL) {
      if(length(C)==1) return(D[,C])
      apply(D[,C],1,paste,collapse='-')
    }
    WhereAB=function(A,B) {Pos=match(A,B); Pos[is.na(Pos)]=0; Pos}
    Na2StrD=function(D,str){
      library(dplyr); mutate_all(D, ~ ifelse(is.na(.), str, .))
    }
    print('SS: 1/4'); V=Combine(U,1:length(Col)); U=cbind(V,U[,-
    (1:length(Col))])
    print('SS: 2/4'); A=Combine(W,Col); Wh=WhereAB(A,V)
    R=rep(0,nrow(W)); E=data.frame(Crit=R,Norm=R,Lema=R,Cg=R,n=R)
    print('SS: 3/4'); indices=which(Wh > 0); E[indices, ]=U[Wh[indices], -1]
    print('SS: 4/4'); E=Na2StrD(E,0); cbind(W,E)
  }
  Li=list(); Col=list(1:5,1:4,2:5,2:4,1:3,3:5,2:3,3:4,3)
  for(i in 1:9){
    print(paste('File:',i))
    R=SS(X,RR[[i]],Col[[i]])
    Li[[i]]=Grep(R,'Crit',0,T)
    Li[[i]]=cbind(Li[[i]], g=rep(10-i,nrow(Li[[i]])))
    X=Grep(R,'Crit',0,F)
    if(i<9) X=X[,1:7] else Li[[10]]=cbind(X,g=rep(0,nrow(X)))
  }
  do.call(rbind, Li)
}
```

4.4. Formas adyacentes

```
Adjacent=function(A,sel) {
  n=length(A)
  if(sel==1) return(c("*",A[-n]))
  if(sel==2) return(c("*","*",A[-c(n,n-1)]))
```

```
  if(sel==3) return(c(A[-1],"*"))
  if(sel==4) return(c(A[-(1:2)],"*","*"))
}
```

4.5. Reordenación

```
SortM=function(D, K=1, A=NULL) {
  D=as.data.frame(D)
  if(is.null(A)) A=rep(T,length(K))
  for(i in length(K):1) D=D[order(D[,K[i]]),]; D
} #D: data.frame, K: key, A=F: ascendente, T: descendente
```

4.6. Contextos

```
Context=function(A,sel,f=3,t=10){
  Li=list(); n=length(A)
  if (sel==1) for(i in f:t) Li[[i]]=c(rep('*',i), A[-(n:(n-(i-1)))])
  if (sel==2) for(i in f:t) Li[[i]]=c(A[-(1:i)], rep('*',i))
  if (sel==1) D=do.call(data.frame, Li[t:f])
  if (sel==2) D=do.call(data.frame, Li[f:t])
  apply(D,1,paste,collapse=' ')
}
```

4.7. Forma inversa

```
Reverse=function(A) {
  A=sapply(strsplit(A, ""),function(x)paste(rev(x),collapse=""))
  A=gsub("([A-ZÁÉÍÓÚÜÑ])",'#¥¥1',A); gsub('([áéíóúü])','¥¥1@',A,T)
}
```

4.8. Atributos

```
VLookUp <- function(D1, D2, c1=1, c2=1) {
  cn=names(D1)[c1]; names(D1)[c1]="v1"; names(D2)[c2]="v2"
  D2=distinct(D2, v2, .keep_all = TRUE)
  library(dplyr);R=left_join(D1,D2,by=c("v1"="v2"))
  names(R)[c1]=cn; R
```

}

Referencia

Agujetas, María / Sánchez-Prieto Borja, Pedro / Ueda, Hiroto. 2022a. *Inventario léxico de Castilla la Nueva. CORPUS CODEA.* https://h-ueda.sakura.ne.jp/lyneal/il/cn/

Agujetas, María / Sánchez-Prieto Borja, Pedro / Ueda, Hiroto. 2022b. *Inventario Léxico de Andalucía. CORPUS CODEA.* https://h-ueda.sakura.ne.jp/lyneal/il/an/

Barnbrook, Geoff. 1996. *Language and Computers.* Edinburg University Press.

Etxeberría Murgiondo, Juan / García Jiménez, Eduardo / Gil Flores, Javier y Rodríguez Gómez, Gregorio. 1995. *Análisis de datos y textos*. Madrid. RA-MA.

Gómez Díaz, Raquel. 2005. *La lematización en español: una aplicación para la recuperación de información.* Gijón. Ediciones Trea.

Halliday, M. A. K. / Teubert, Wlfgang / Yallop, Colin / Čermáková, Anna. 2005. *Lexicology and Corpus Linguistics. An Introduction.* London. Continuum.

McEnery Tony and Hardie, Andrew. 2012. *Corpus Linguistics. Methods, Theory and Practice.* Cambridge University Press.

R Core Team. 2023. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.

Sánchez-Prieto Borja, Pedro. 2011. La edición de textos españoles medievales y clásicos. Criterios de presentación gráfica. San Millán de la Cogolla. Cilengua.

Sánchez-Prieto Borja, Pedro / Ueda, Hiroto. 2018. *Inventario léxico del corpus CODEA. I. Castilla la Vieja*. https://h-ueda.sakura.ne.jp/lyneal/ilc-cv.htm

Sánchez-Prieto Borja, Pedro / Moreno Sandoval, Antonio, Ueda, Hiroto. 2020. *Las voces de Madrid en su historia. Inventario léxico del corpus ALDICAM.* DOI: https://doi.org/10.37536/LEXALDICAM.2020 https://h-ueda.sakura.ne.jp/lyneal/il/aldicam/

Ueda, Hiroto / Sánchez-Prieto Borja, Pedro / Moreno Sandoval, Antonio. 2020. "Lematización y visualización cartográfica del corpus CODEA. Formas de la conjunción ‘y’ en el norte de Castilla medieval”, *Estudios de Lingüística del Español*. 42, pp. 245-261.

FIN