Pontificia Universidad Católica de Chile

Facultad de Letras

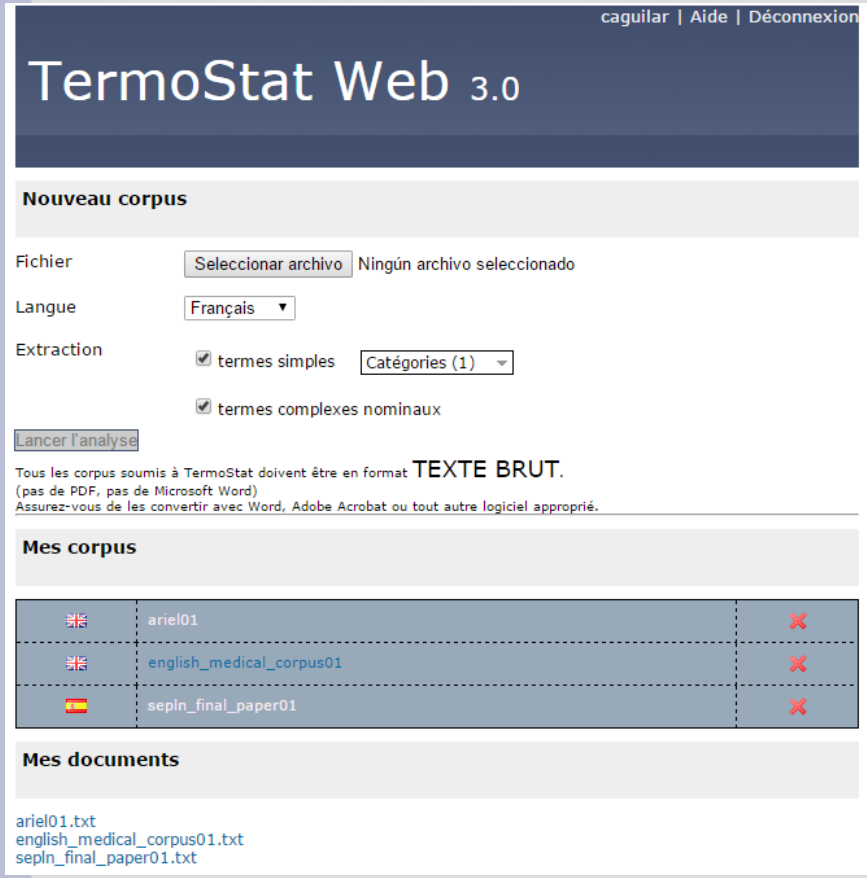
Curso de traducción automática de lenguas naturales

César Antonio Aguilar
Facultad de Lenguas y Letras
27/11/2015

Cesar.Aguilar72@gmail.com

Extracción de términos (1)

Vamos a cerrar el curso abordando un tema estrechamente relacionado con la traducción automática: la **extracción de términos** usando recursos computacionales.



La extracción de términos, para decirlo de manera muy breve, es una línea de investigación aplicada, derivada de la extracción de información. Su objetivo central es identificar candidatos a términos insertos en documentos científicos y técnicos.

Extracción de términos (2)

La idea de extraer términos usando recursos computacionales viene desde mediados del siglo XX, y fue planteada por **Eugen Wüster**, para muchos el padre fundador de la terminología.

¿Cómo se hace esto? De acuerdo con Wüster y otros terminólogos, el proceso consiste en identificar patrones lingüísticos y para-lingüísticos que,potencialmente, sean candidatos a términos. Para determinar tales candidatos, se hace uso tanto de análisis lingüísticos, de métodos estadísticos, así como de herramientas computacionales.

Eugen Wüster (1898-1977)

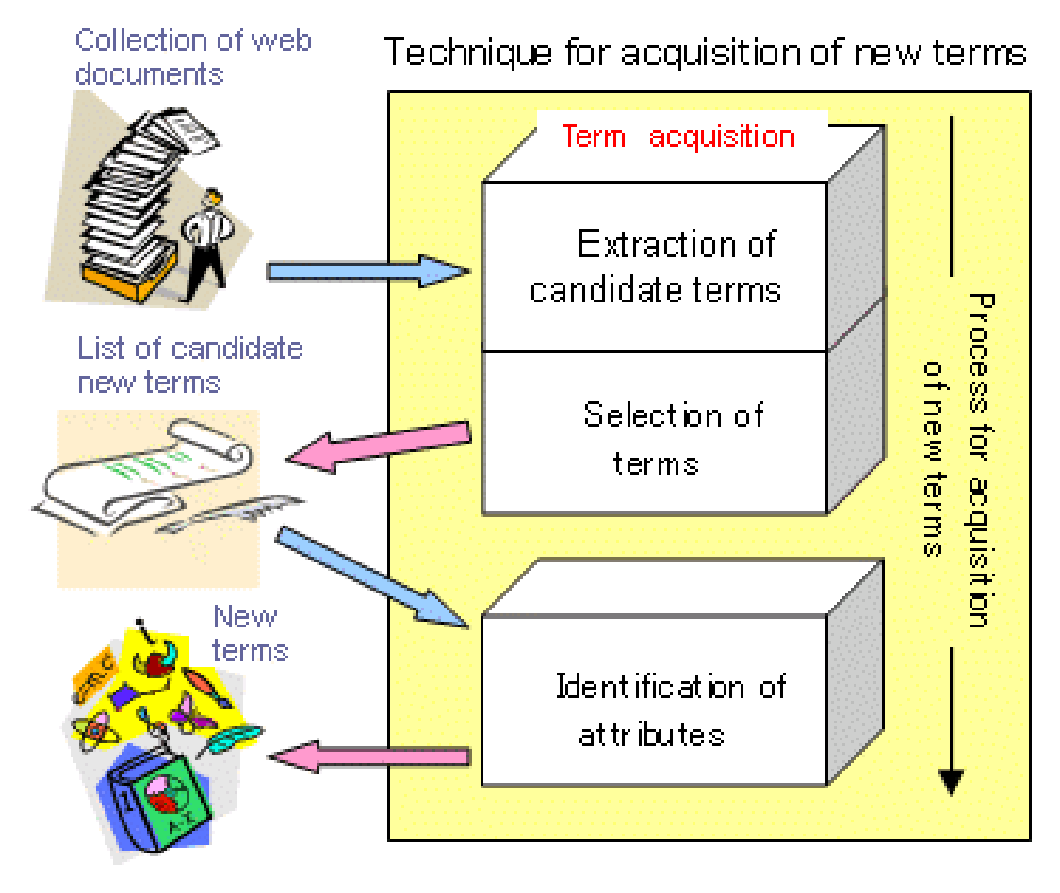
¿Qué es un término? (1)

Claro está, antes de realizar este proceso, primer hay que determinar qué es lo que entendemos por un término. Una definición útil es la siguiente:

- Unidad de significación especializada dentro de un dominio de conocimiento.
- Cuenta con una capacidad referencial y nominativa concreta, así como un significado especializado.
- Representación en lenguaje natural: frases nominales, frases adjetivas, frases prepositivas y construcciones verbales en una función nominal

[cuisine]	PESTO
	PÂTE FEUILLETÉE
	TASSE À MESURER
	APPAREIL
[Internet]	LOGICIEL DE NAVIGATION
	LIEN
	MOTEUR DE RECHERCHE
	PAGE D'ACCUEIL
[terminologie]	CONCEPT
	ONOMASIOLOGIE
	FICHE DE TERMINOLOGIE
	DOMAINE DE SPÉCIALITÉ

¿Qué es un término? (2)



¿Cómo determinamos que algo es un término o no? Esta decisión no es del todo sencilla. Por lo general, la extracción terminológica lo que hace es generar listados de **candidatos a términos**, los cuales posteriormente, son validados por expertos del área, quienes cuáles equivalen a un vocablo especializado.

¿Cómo se reconocen automáticamente? (1)

Un avance sustancial que se la logrado respecto a determinar qué tan especializada es una palabra (o una secuencia de palabras) dentro de un dominio de conocimiento es hacer un contraste entre dos valores concretos: su grado de *unicidad* (ing. *Unithood*) como patrón morfológico y/o sintáctico; *versus* su grado de *termicidad* (ing. *Termhood*), es decir, qué tanto este patrón es usado como un término auténtico.

unithood

how it is grammatically defined? (e.g. noun phrases)

termhood

does it belong to a domain?

¿Cómo se reconocen automáticamente? (2)

¿Por qué necesitamos esto? Porque nos interesa determinar qué secuencias de palabras tienen un valor auténtico como término, en contraste con aquellas que son parte de un vocabulario más general, y que no nos aportan un conocimiento especializado relevante. Veamos un ejemplo de esto. Si leen el siguiente fragmento, ¿para ustedes qué palabras son términos, y por qué lo son?:

Tal y como sale de fábrica el disco duro no puede ser utilizado por un sistema operativo.

Antes tenemos que definir en él una o más particiones y luego hemos de darles un formato que pueda ser entendido por nuestro sistema.

¿Cómo se reconocen automáticamente? (3)

Si ya propusieron sus candidatos a términos, ahora tratemos de hacer esto automáticamente, y para ello, vamos a seguir la recomendación de un lingüista computacional alemán llamado **Ulrich Heid**. Lo que podemos hacer, si trabajamos con un corpus lingüístico, es lo siguiente:

Ulrich Heid

Tokenización: identificación de límites entre palabras y oraciones

Análisis morfosintáctico: identificación de unidades morfológicas y sintácticas, así como análisis de su distribución (chunking y parsing).

Etiquetado de partes de la oración (POS): colocación de etiquetas morfológicas y sintácticas en las palabras y oraciones localizadas.

¿Cómo se reconocen automáticamente? (4)

Concentrémonos en la última parte, y tratemos de identificar candidatos a términos usando un etiquetado morfosintáctico. Así se verá nuestro texto con esta clase de anotado:

Tal_tal/QU y_y/CC como_como/CSUBX sale_salar/VLfin de_de/PDE fábrica_fábrica/NC el_el/ART disco_disco/NC duro_duro/ADJ no_no/NEG puede_poder/VMfin ser_ser/VSinf utilizado_utilizar/VLadj por_por/PREP un_un/ART sistema_sistema/NC operativo_operativo/ADJ ._./FS

Antes_antes/ADV tenemos_tener/VLfin que_que/CQUE definir_definir/VLinf en_en/PREP él_él/PPX una_un/ART o_o/CC más_más/ADV particiones_particiones/NC y_y/CC luego_luego/CSUBF hemos_haber/VLfin de_de/PDE darles_dar/VLinf un_un/ART formato_formato/NC que_que/CQUE pueda_poder/VMfin ser_ser/VSinf entendido_entender/VLadj por_por/PREP nuestro_nuestro/PPO sistema_sistema/NC ._./FS

El algortimo C-Value/NC-Value (1)

Una vez que tenemos etiquetado nuestro corpus, podemos usar varias técnicas para reconocer posibles palabras o frases que funcionen como términos. Una de esas técnicas es el algortimo **C-Value/NC-Value**.

Este algoritmo fue desarrollado por Sophia Ananiadou y sus colegas del **National Center of Text Mining** (**NaCTeM**) de Manchester, Reino Unido, para el reconocimiento de términos en inglés.

Sophia Ananiadou

NaCTeM
The National Centre for Text Mining

www.nactem.ac.uk

El algortimo C-Value/NC-Value (2)

Este algoritmo es un tipo de método híbrido, diseñado para el reconocimiento de términos en el área de biomedicina, precisamente usando documentos obtenidos a través de **MedLine Plus**. En concreto, cubre dos fases:

lingüística

- Etiquetado de partes de la oración
- Detección de candidatos a término
- Eliminación de candidatos por medio de una lista de paro

estadística

- Ordenación de candidatos con base en su frecuencia y longitud

El algortimo C-Value/NC-Value (3)

Vamos a detectar secuencias que se correspondan con nombres o frases nominales que consideremos como posibles candidatos, construyendo patrones basándonos en una **gramática regular**:

<NC | NP | PE>$^{+}$

servidor, tarjeta madre, MB, arquitectura von Neumann

<NC> <ADJ> (<PDE> <NC | NP>)*

sistema operativo, unidad central de procesamiento

<NC> <PDE> <NC | NP | NMEA>

ángulo de visión, ancho de banda, tasa de transferencia

El algortimo C-Value/NC-Value (4)

Y como todo en la vida, también consideramos casos excepcionales:

<VLFIN | VLINF>

compilar, descifrar, ensamblar

<NC>? <ACRNM>

DDR2, IP, slot ISA

<NC> <PDE> ((<NC> <ADJ>) | (<ADJ> <NC>))

computadora de uso general

El algortimo C-Value/NC-Value (5)

Volvamos a la pregunta inicial: ¿cuáles serían nuestros posibles candidatos a términos? Veamos:

Tal y como sale de fábrica el disco duro no puede ser utilizado por un sistema operativo.

Antes tenemos que definir una o más particiones en el disco duro y formatearlo para que pueda ser entendido por nuestro sistema.

El algortimo C-Value/NC-Value (6)

Para mejorar el proceso de detección de candidatos términos, el algoritmo considera el uso de una **lista de paro** (ing. **Stop Words**), la cual agrupa todas aquellas palabras que no son términos considerados en el área, p. e., en un área como las ciencias de la computación, las siguientes difícilmente serán consideradas como tales:

Detalle, elegante, importante, mayoría, mercado, opción, tamaño, vez, etc.

El algortimo C-Value/NC-Value (7)

Usando estas listas de paro, logramos filtrar palabras que no tienen nada que ver con posibles candidatos a términos:

~~fábrica~~	partición
disco duro	formato
sistema operativo	sistema

Calculando valores de *unithood* y *termhood* (1)

Una vez concluida la fase de análisis lingüístico, pasamos a otra en donde se realizan una serie de cálculos estadísticos para determinar cuáles son los mejores candidatos a términos, tomando en cuenta el contraste entre *unithood* y *termhood*. Aquí lo que consideramos son 4 parámetros:

1. Frecuencia total de ocurrencia de una frase nominal en el corpus
2. Frecuencia total de ocurrencia de una frase nominal como parte de secuencias más largas
3. Número de dichos candidatos de mayor longitud
4. Longitud del candidato a término

Calculando valores de *unithood* y *termhood* (2)

Sin entrar en demasiados detalles, tratemos de ver la lógica que sigue este algoritmo para calcular dichos valores. Supongamos que tenemos los siguientes candidatos:

candidato	longitud	frecuencia
disco duro	2	3
sistema operativo	2	2
sistema	1	2
partición	1	1

Pregunta: intuitivamente, ¿cuáles creen que sean nuestros mejores candidatos?

Calculando valores de *unithood* y *termhood* (3)

En términos un poco más técnicos, esto es lo que hace el C-Value para determinar que una frase nominal de longitud 2, y con una frecuencia de aparición de 3 casos es un buen candidato:

candidato	longitud	frecuencia
sistema operativo	2	2

$$C\text{-}value = (1 + log_2|a|) * f(a)$$

$$C\text{-}value(sistema\ operativo) = (1 + log_2(2)) * 2$$

$$C\text{-}value(sistema\ operativo) = 4$$

Calculando valores de *unithood* y *termhood* (4)

Ahora bien, si bien el algoritmo C-Vaue ya nos dio buenas pistas para detectar buenos candidatos, todavía podemos establecer otra fase de filtrado, tratando de reconocer palabras adyacentes a los nombres considerados como términos, p. e:

Un **llavero USB** es un **pequeño dispositivo de almacenamiento** que **utiliza** la **memoria flash** para guardar la información sin necesidad de pilas.

Tienen una capacidad de almacenamiento que va desde algunos megabytes hasta 8 gigabytes.

Calculando valores de *unithood* y *termhood* (5)

Una opción para mejorar nuestra búsqueda es identificar **palabras de contexto** (esto es, aquellos nombres, adjetivos y verbos que aparecen asociados a un término), podemos corroborar si nuestros candidatos son buenos o no:

candidato	palabras de contexto
disco duro	formatear, guardar, GB
sistema operativo	instalar, configurar, cargar
partición	disco duro, crear, swap
sistema	levantar, iniciar, apagar

Calculando valores de *unithood* y *termhood* (6)

Ahora, establezcamos la fórmula para hacer el cálculo de las palabras en contexto:

$$weight(w) = \frac{t(w)}{n}$$

w es la palabra de contexto analizada

weight(w) es el peso asignado a la palabra *w*

t(w) es el número de candidatos con los que aparece la palabra *w*

n es el número total de candidatos considerados (para expresarlo como una probabilidad)

Calculando valores de *unithood* y *termhood* (7)

En el siguiente ejemplo, podemos ver cómo se aplica este cálculo:

"es necesario **formatear** el *disco* y ***crear*** dos **particiones**"

"el tipo de **partición** para el sistema operativo **Linux es ext3** "

$$weight(formatear) = \frac{1}{2} \qquad weight(particion) = 1$$

Calculando valores de *unithood* y *termhood* (8)

Comparemos entonces ambos valores, C-Value y NC-Value:

candidato	frecuencia	C-value	NC-value
disco duro	3	6	5.3
sistema operativo	2	4	5.0
formato	2	2	1.8
partición	1	1	1.4
sistema	2	1	0.8

Pregunta: de acuerdo con estos valores, ¿cuáles son los mejores candidatos a términos?

Extractores automáticos

C-Value/NC-Value es uno de muchos algoritmos que pueden ser usados para realizar una tarea de extracción terminológica. La función principal de estos algoritmos es ayudar a evaluar qué tan eficiente es un sistema de extracción. Igualmente, hay muchos sistemas de extracción, académicos y comerciales:

wandora

YATE on the Web

SDL TRADOS

Sketch Engine

Ejercicio (1)

Vamos a realizar una tarea de extracción de términos, y vamos a tratar de comparar el desempeño de dos sistemas de este tipo, uno llamado **TermoStat**, desarrollado por **Patrick Drouin** (Université de Montréal), y **Sketch Engine.** Empecemos con **TermoStat**.

Patrick Drouin

TermoStat Web 3.0

http://termostat.ling.umontreal.ca/

Ejercicio (2)

Lo primero que hay que hacer es suscribirse, para lo cual tienen que llenar la ventana asociada con la opción **Inscription**:

Ejercicio (3)

TermoStat les permite trabajar con 5 lenguas diferentes: español, francés, inglés, italiano y portugués.

Vamos a trabajar entonces con dos *corpora*, uno en inglés llamado *english_medical_corpus01*, y otro en español nombrado *spanish_medical_corpus01.* Veamos el primer caso. Seleccionemos nuestro archivo en TXT:

Ejercicio (4)

¿Qué resultados obtuvieron? Veamos una fracción:

Corpus >> **english_medical_corpus01** caguilar | Aide | Déconnexion

Résultats

Liste des termes | Nuage | Statistiques | Structuration | Bigrammes

Candidat de regroupement	Fréquence	Score (Spécificité)	Variantes orthographiques	Matrice
patient	404	100.31	*patient* *patients*	Nom
hospitalization	44	91.78	*hospitalization* *hospitalizations*	Nom
complication	82	90.15	*complication* *complications*	Nom
ibd	32	85.61	*ibd*	Nom
ccd	31	82.88	*ccd*	Nom
diarrhea	44	80.91	*diarrhea*	Nom
morbidity	30	75.52	*morbidity*	Nom
nosocomial case	25	75.33	*nosocomial cases*	Adjectif Nom
rotavirus	23	72.12	*rotavirus*	Nom
bariatric surgery	21	68.77	*bariatric surgery*	Adjectif Nom
hospital stay	23	66.52	*hospital stay*	Nom Nom
mortality	42	62.79	*mortality*	Nom
cholecystitis	17	61.52	*cholecystitis*	Nom
abdominal pain	23	61.04	*abdominal pain*	Adjectif Nom
hospital admission	22	60.25	*hospital admission* *hospital admissions*	Nom Nom
diarrheas	16	59.57	*diarrheas*	Nom
study period	17	58.16	*study period* *study periods*	Nom Nom
incidence	47	57.63	*incidence*	Nom
admission	76	55.97	*admission* *admissions*	Nom

Preguntas: ¿Consideran que son buenos resultados o no? ¿Por qué lo creen?

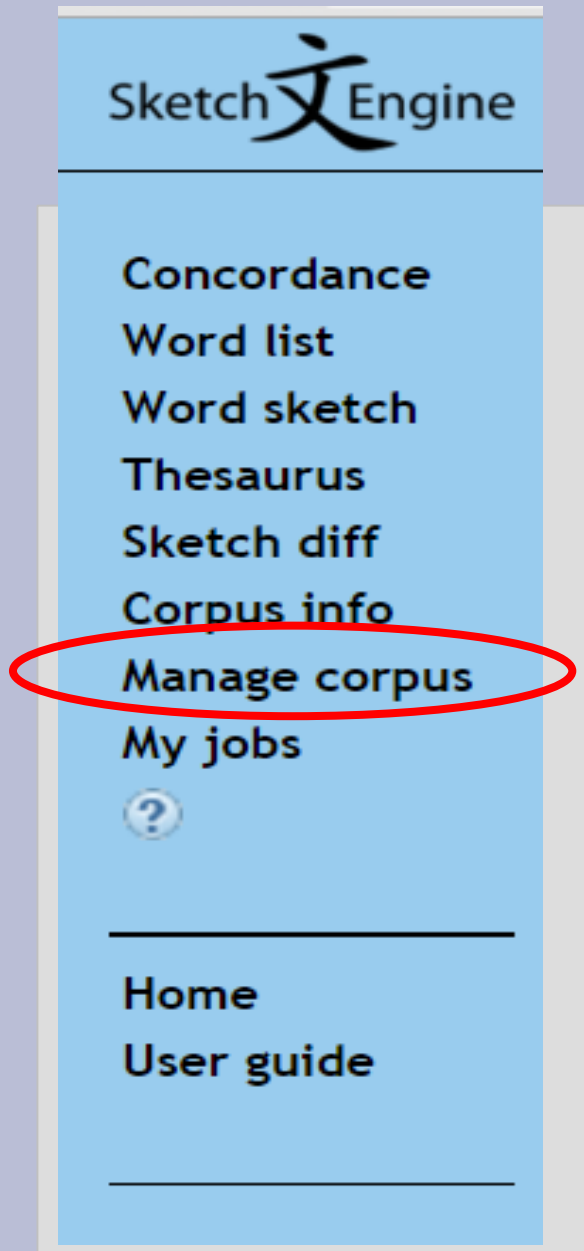
Ejercicio (5)

Ahora, vamos a hacerlo con **Sketch Engine**. Primero, accedan al sistema, y luego dentro de la opción *My own*, elijan el corpus llamado *English_Medical_Corpus01*:

Corpora: Recent | My own | Featured | Parallel | All

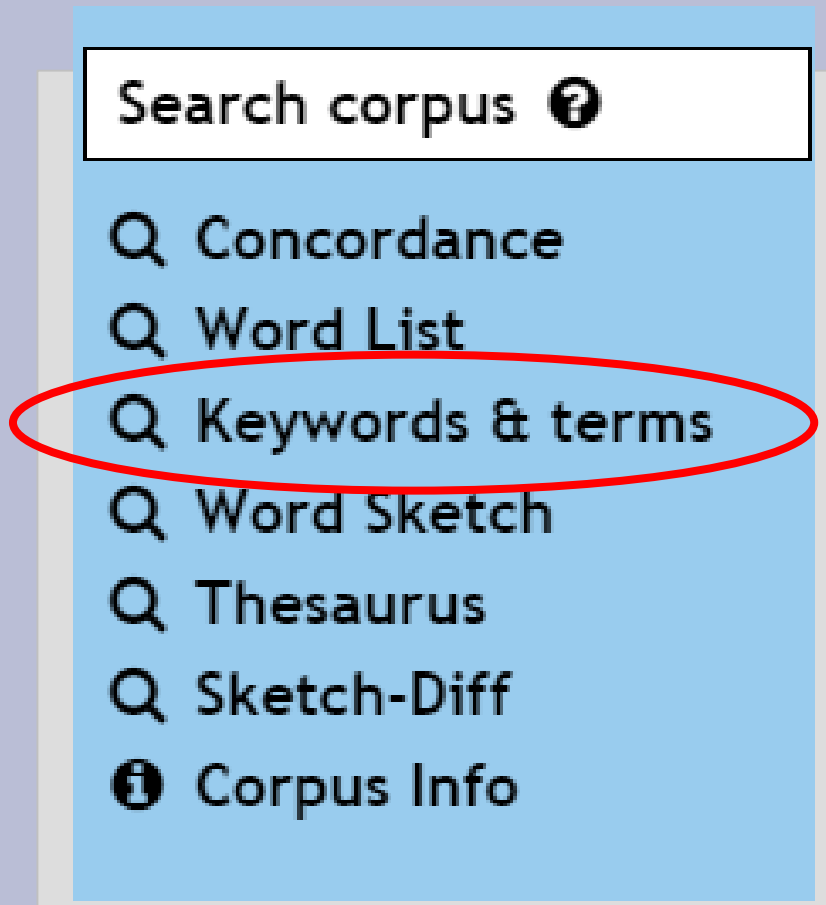
Language	Name	Words
English	art_medicina_ingles	264,397
English	articulo_medicina	4,402,227
English	compound_adjectives_EnTenTen2012	2,087,447
English	compuestos_irregulares	273,954
English	compuestos_irregulares(todos)	2,945,810
English	Corpus Prueba_Inglés	41
English	Corpus_Ingles01	43
English	English_Medical_Corpus01	26,304
English	HTTPS	2,118
English	irregulares atributivos	–
English	irregulares_depurados	224,108

Ejercicio (6)



Una vez que lo hayan hecho, vayan a la parte izquierda de su menú de opciones, y seleccionen el botón **Manage corpus:**

Ejercicio (7)



Hecha la elección anterior, ahora busquen el botón **Search corpus**, y elijan la opción **Keywords & terms**, para que les genere un listado de candidatos a términos:

Ejercicio (8)

Veamos un ejemplo de los resultados que nos genera Sketch Engine:

Keywords		Score	F	RefF
diag	W	1,226.60	46	1,256
nosocomial	W	1,146.71	44	1,586
nager	W	1,107.26	39	393
rotavirus	W	1,053.58	46	3,592
mbds	W	1,050.22	36	36
ibd	W	1,004.33	51	6,291
ccd	W	601.42	50	18,563
spp	W	599.28	39	11,719
pmrs	W	519.58	18	187
hinchey	W	518.62	19	942
clavien	W	497.58	17	7
difficile	W	494.70	22	3,913
campylobacter	W	470.28	19	2,372
diarrheas	W	464.00	16	129
cholecystitis	W	451.98	17	1,316
dge	W	449.32	16	557

Terms		Score	F	RefF
hospital stay	W	426.35	23	7,518
bariatric surgery	W	384.94	21	7,744
study period	W	343.81	16	4,719
diverticular disease	W	340.48	12	432
primary anastomosis	W	321.52	11	40
abdominal pain	W	305.31	23	15,633
clinical process	W	287.23	10	284
hospital admission	W	259.75	10	1,673
retrospective study	W	237.87	9	1,435
pancreatic fistula	W	234.11	8	39
peritoneal lavage	W	233.87	8	63
endoscopic treatment	W	233.41	8	80
seasonal distribution	W	232.71	8	125
reference method	W	230.88	8	232
predictive value	W	226.25	9	2,172
complete response	W	215.03	8	1,197

Ejercicio (9)

Preguntas:

1. ¿Qué diferencias hay entre lo que es un término de una *keyword*?

2. ¿Quién dio, según su criterio, mejores candidatos a términos: **TermoStat** o **Sketch Engine**?

Tarea:

- Apliquen el mismo proceso al corpus médico en español, y comparen resultados

- ¿Según su comparación, los resultados son peores o mejores? ¿El factor *lengua* afecta o no el desempeño de ambos sistemas?

Gracias por su atención

Blog del curso:
http://cesaraguilar.weebly.com/traduccioacuten-automaacutetica.html