

Muestreo y representatividad: el script Redemú

Por: Jorge Vivaldi y Rogelio Nazar

Transcripción del seminario IULAterm
presentado por R. Nazar
el día 6 de noviembre de 2008
en la Universidad Pompeu Fabra.

Contenidos

- 1) Introducción
- 2) Explicación del algoritmo
- 3) Modo de empleo
- 4) Funciones de soporte
- 5) Conclusiones y trabajo futuro

Introducción

En este seminario vamos a presentar un programa informático que se llama Redemú. Este es el acrónimo de *Representatividad de la muestra*. Originalmente el programa se llamaba CET, por *Clasificador Estructural de Términos*, pero nos dimos cuenta de que el nombre no representaba justamente lo que el programa hace, porque no hay realmente un problema de clasificación.

La mayor parte del guión de la presentación es el protocolo normal de un paper, pero quisiera centrar la atención en esta separación entre los puntos dos y tres, en los que hablamos, por un lado, de cómo es el método y, por el otro, de la forma en que se tiene que utilizar esta implementación, es decir que el punto dos es más teórico y el punto tres más práctico. El punto cuatro, funciones de soporte, son pequeñas cosas que se han ido agregando en base a la solicitud de los primeros usuarios y que pueden resultar prácticas para todos.

El problema general que tenemos aquí es el del muestreo y la representatividad, que es un tema que trasciende lo que es estrictamente lingüístico, es un problema de la ciencia en general. Es un problema estadístico o meta-científico.

En este seminario sólo vamos a tratar un aspecto del muestreo que es el tamaño. Sin embargo, es sabido que ese no es el único factor que interviene en la representatividad de una muestra, ya que también es fundamental la calidad de dicha muestra. Podemos tener una muestra muy grande y también muy mala, dependiendo de la inteligencia con la que hemos seleccionado los datos.

Entonces, dejando de lado este aspecto más cualitativo del muestreo, lo que tenemos es una población teórica de la cual extraemos una muestra para estudiar un fenómeno determinado y la pregunta clave es si esa muestra es o no representativa de la población en relación a dicho fenómeno. ¿Por qué? Porque si es representativa esto significa que podemos generalizar el conocimiento que tengamos de esa muestra hacia la población.

Explicación del algoritmo

Tal vez convenga hacer una aclaración aquí acerca de cómo surgió este problema. La mayoría de nosotros teníamos una lista de términos y queríamos saber si esos términos eran suficientes o no, y no simplemente obtener una lista de mil términos porque ése sería un valor arbitrario. Entonces, queríamos tener un número y también algún tipo de fundamento teórico que justifique por qué mil y no mil quinientos. Queremos una tranquilidad de que hemos elegido un valor mínimo con un fundamento.



Figura 1: Una urna con bolas de colores.

Este es el problema de la mayoría de ustedes, pero hay algunos que tienen un problema muy similar o idéntico, que es, dado un corpus, saber si la muestra que han seleccionado es o no suficiente en relación a la variable que están estudiando. Quiero decir que el problema es exactamente el mismo, sólo que los individuos representarán cosas distintas. Luego lo veremos con más detalle. Ahora, para ofrecer una abstracción incluso mayor, vamos a dejar de hablar de corpus o de listas de términos y vamos a plantear una situación hipotética en la que tenemos una urna (figura 1) con un número indeterminado de bolas de un número también indeterminado de colores. En este caso, suponiendo que sólo podemos tomar una bola por vez, y que cada bola tiene la misma probabilidad de ser elegida, la pregunta es, cuántas bolas tengo que sacar de la urna para tener una muestra que sea representativa en cuanto al número de colores. Es decir que esta misma metodología se podrá exportar a cualquier otro tipo de muestreo.

La tabla 1 es un ejemplo del tipo de individuo que tenemos y el tipo de valor que asociamos a cada individuo. Esta es una muestra de términos que tienen asociado como valor su patrón morfológico. ¿Por qué tenemos que asociar un valor? Porque si no tenemos ningún tipo de valor asociado no podemos hacer ninguna estimación ni podemos plantear el problema. En este caso, trasladando al ejemplo de las bolas de colores, lo que en la primera celda de la izquierda tendríamos sería "bola número 1" y en la celda de la derecha, el color. Así, entonces, en la siguiente fila tendríamos: "bola número 2" -> amarillo; "bola número tres" -> azul, etc. En el caso de un corpus, si lo que nos interesa es el léxico, lo que tendríamos en esta tabla (tabla 2) serían posiciones en el corpus y en la celda de al lado la forma correspondiente. En un ejemplo cualquiera, tendríamos "palabra número 1 del corpus" -> forma: "La"; y en la fila siguiente: "palabra número dos" -> "mayoría"; y en la siguiente: "palabra número tres" -> "de"; y así sucesivamente.

Término	T.Funcional
linfoma inmunoblástico	NA
retenedor de matriz	NPN
armazón balcánico	NA
sonda de Foley	NPN
terapia gaseosa	NN
defectuoso	A
síndrome de la válvula oscilante	NPDNA
barotrauma ótico	NA
singenesia	N
lívido	N
radiación corpuscular	NA

Tabla 1: Ejemplo del tipo de individuo y el tipo de valor asociado.

Posición	Token
1	La
2	mayoría
3	de
4	las
5	áreas
6	de
7	descarga
8	estaban
9	colonizadas
10	por
...	...

Tabla 2: Otro ejemplo de individuos con un valor asociado.

A continuación, la gráfica de la figura 2 representa el crecimiento de la diversidad a distintos tamaños de la muestra. Entonces tenemos en el eje horizontal la cantidad de bolas y en el eje vertical la cantidad de eventos diferentes, que, nuevamente, en el ejemplo de las bolas sería cantidad de colores. Transladando al ejemplo de los términos, tenemos en el eje horizontal distintos tamaños de la lista de términos y en el eje vertical el número de valores diferentes. Hemos elegido en este caso el patrón morfológico pero podría ser cualquier otro valor que tenga una mínima diversidad, porque si fuese una variable que puede adoptar sólo unos pocos valores distintos, esa variabilidad se agotaría muy rápido. La idea detrás de esta gráfica es encontrar, con algún algoritmo, un punto en que este crecimiento deje de ser tan acusado, es decir que la curva se vuelva asintótica.

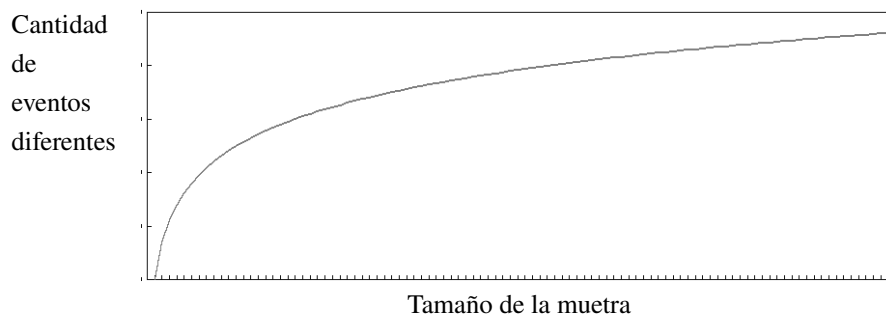


Figura 2: Función del crecimiento de la diversidad.

¿Cómo podemos hacer para medir el crecimiento de una función? Un método, bastante intuitivo o gráfico, es trazar en cada punto de la curva de crecimiento una recta tangente. Y la manera de cuantificar este crecimiento es medir la pendiente que tiene cada recta. En la figura 3 se ve de nuevo el crecimiento de la función en los dos puntos *a* y *b*. En el primer punto vemos que tenemos un crecimiento acusado porque la recta tiene una pendiente muy alta y en el segundo punto el crecimiento es menor. La manera de cuantificar este crecimiento entonces es ver qué pendiente tiene la tangente en cada punto.

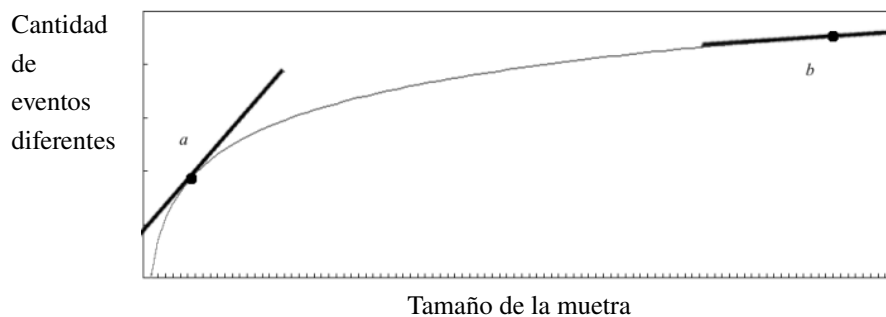


Figura 3: Función del crecimiento de la diversidad.

En la práctica esto significa que debemos decidir cuál será el valor de pendiente que actúe como umbral. Es decir, una pendiente que una vez alcanzada nos permita considerar que la lista de términos resultante es representativa de la población. Una recta que forme un ángulo de seis grados respecto al eje horizontal va a ser suficiente para nosotros, pero es necesario ser concientes de que podría ser un umbral más o menos exigente. Entonces la representatividad en este caso tiene un costado problemático.

En la tabla 3 volvemos al ejemplo de los términos. Aquí ya la forma de los términos no nos va a importar. Lo que vamos a rescatar es solamente la posición de este término en la lista. En la primera columna tenemos el número de orden del término; en la segunda columna tenemos el patrón morfológico correspondiente y en la tercera la cantidad de patrones distintos. Observemos que para el término número 4 el número de patrones distintos sigue siendo 3 porque los patrones correspondientes a los términos 2 y 3 son iguales. Se repite el patrón Nombre + Preposición + Nombre, como en *papel de cera* o *nanotubo de carbono*. En la figura 4 representamos en forma gráfica los primeros valores de esta tabla. Y la manera de calcular las pendientes de las tangentes es preguntándonos cuántas posiciones tenemos que desplazarnos en el eje x para poder incrementar

una posición en el eje y. Es decir, cuántos términos tengo que agregar para poder ver un patrón nuevo. Si tengo que agregar muchos términos para encontrar un patrón nuevo, voy a a tener una recta con una baja pendiente.

nº	patrón	nº patrones distintos
1	VN	1
2	NpN	2
3	NA	3
4	NpN	3
5	NN	4
6	A	5
7	NpDNA	6
8	NA	6
9	N	7
10	V	8
11	NA	8
12	N	8
13	N	8
14	NVA	9
15	NpN	9
16	VV	10
17	N	10
18	NpN	10

Tabla 3: Crecimiento de la diversidad de patrones.

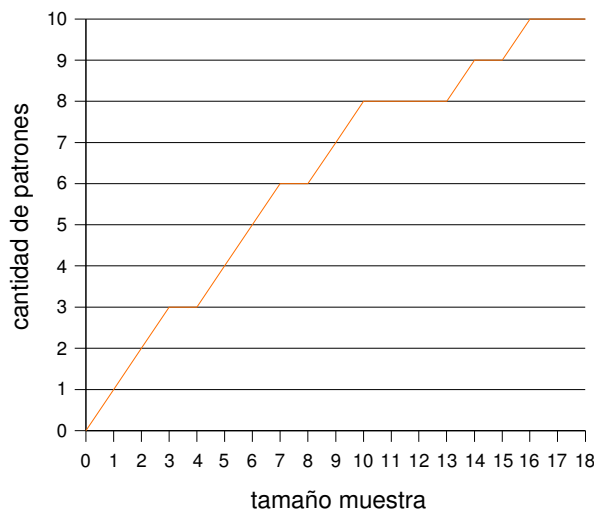


Figura 4: Representación gráfica del crecimiento.

Ahora vamos a entrar en un nivel de complejidad más porque queremos estar a salvo del factor azar. Es decir, tenemos la preocupación de haber encontrado un punto en que esa función deja de crecer por una simple casualidad. Entonces, como conocemos el comportamiento de las curvas de frecuencias, que cumplen con la Ley de Zipf, sabemos que no solamente tenemos que buscar un punto, sino un intervalo, es decir una serie de puntos contiguos. Buscamos un segmento de n puntos en los que podamos trazar rectas tangentes que, además de ser inferiores al umbral, sean cada vez menores, porque esto nos proporciona una seguridad extra.

Un segundo seguro contra el potencial efecto de la casualidad es no hacer este experimento una sola vez sino mil veces. Mil porque es un número grande. Es decir, repetir el ensayo muchas veces ordenando cada vez los términos en forma aleatoria. Cuando nosotros hacemos el ensayo una vez, el resultado es binario, porque el programa nos dirá que la muestra es representativa o que no lo es. Si hacemos el ensayo mil veces, la máquina nos dirá: "de los mil ensayos realizados, en 800 (o k) veces es posible extraer una muestra representativa. Entonces, para medir el efecto de la casualidad, lo que hacemos es calcular cuál es la probabilidad de tener por casualidad 800 (o k) ensayos exitosos a partir de mil (o n) ensayos totales.

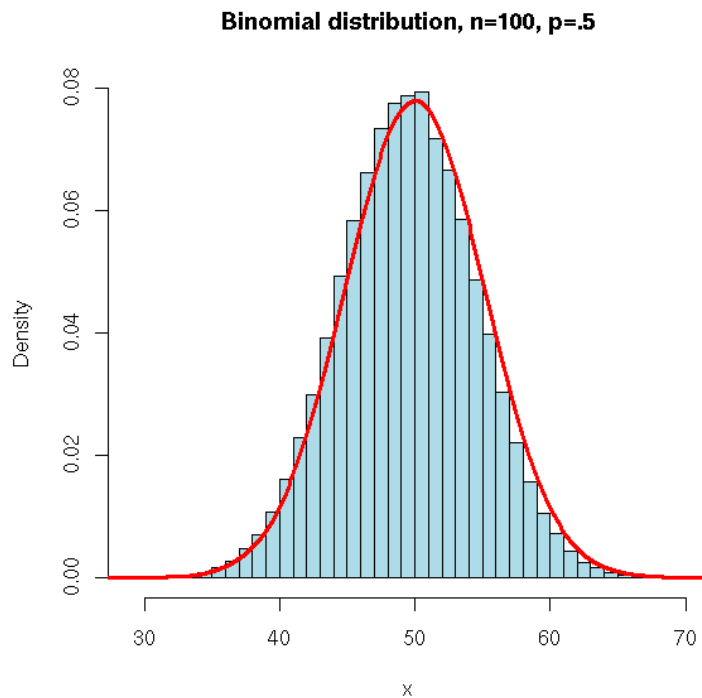


Figura 5: Test binomial. Probabilidad de tener por casualidad k éxitos en n ensayos. (Ejemplo tomado de la web http://zoonek2.free.fr/UNIX/48_R/07.html)

La curva que aparece en la figura 5 es la curva de la distribución binomial y es un ejemplo de qué pasa cuando uno tira una moneda 100 veces. Tirar una moneda puede tener solamente dos resultados (a menos que caiga de canto y se mantenga así, pero esa es una probabilidad remota). Si es una moneda normal tiene una probabilidad del 50% de salir cara o cruz. Si tiramos 100 veces la moneda, ésta es la probabilidad de tener distinto número de veces cara o cruz. Lo más probable es que salga 50 veces cara y 50 veces cruz, pero también puede ser que salga 40 y 60, etc, aunque la probabilidad de este resultado es mucho menor. Ahora, si la moneda sale 70 veces cara, entonces la probabilidad es mínima y sospecharíamos que hay algo raro, es decir, que no es una probabilidad del 50%.

Esta misma lógica es la que aplicamos a los mil ensayos anteriores. Es decir, hacemos mil ensayos y obtenemos 700 resultados positivos, lo que quiere decir que la probabilidad de que nos haya salido ese resultado por casualidad es mínima. Por lo tanto, tenemos una seguridad de que la muestra es representativa. Esta es la p , la probabilidad de la hipótesis nula que sería que nuestro ensayo es el resultado de una variable aleatoria.

Sin embargo, lo que tenemos hasta el momento sigue siendo una respuesta del tipo SI/NO. Es decir, yo he seleccionado una muestra, someto esa muestra al análisis y el resultado puede ser "es representativa" o "no lo es". Y en realidad yo no quiero saber solamente eso sino también cuál es la muestra mínima, porque tener 3000 términos para analizar es muy laborioso. Entonces queremos saber si el mismo resultado se podría tener con 800 o con 500 términos.

Lo que tenemos en la figura 6 es un histograma, muy similar al anterior, que es el resultado de un experimento que hicimos con una lista de términos de un diccionario de medicina que tiene 32.000 entradas. Es decir que sometimos una lista de 32.000 términos a este experimento (en el eje horizontal los números están divididos por 1000 para facilitar la lectura, pero donde dice 1 debería leerse 1000, etc.) y lo que este histograma nos informa es la cantidad de veces que el programa declaró como representativa una muestra a distintos tamaños. Lo que hacemos aquí entonces es calcular la media de los tamaños seleccionados.

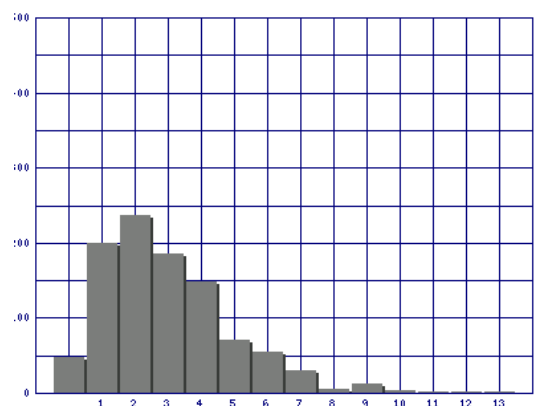


Figura 6: Histograma de muestras mínimas.

La media en este caso es de 3339. La media es muy fácil de calcular. Simplemente sumamos todos los valores de la muestra y dividimos por la cantidad de valores que tenemos:

$$\bar{x} = \frac{1}{n} \cdot \sum_{i=1}^n x_i$$

Como otro control, para saber justamente qué tan representativa es la media respecto al resto de los valores de la muestra, calculamos también el error estándar, que nos dice qué tan alejados están el resto de los valores de la muestra respecto a esa media:

$$SE_{\bar{x}} = \frac{s}{\sqrt{n}}$$

El símbolo s es la desviación estándar, a su vez definida como:

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$$

Si los valores tienen tendencia a concentrarse alrededor de un punto, entonces el error estándar será menor. Si, en cambio, se reparten de una manera homogénea, la media será poco significativa.

Modeo de empleo

Hasta aquí hemos hablado de cuál sería la lógica para decidir si una muestra es o no representativa. El resto de este seminario va a estar centrado en cómo utilizar el programa. Ahora mismo está implementado como una aplicación web, y la dirección es la siguiente:

<http://rc16.upf.es/redemú>

Esta es la dirección de mi propio ordenador, y no es un ordenador muy potente, es decir que si todos se conectan al mismo tiempo se va a notar el deterioro en la velocidad de procesamiento. De cualquier manera funciona bastante rápido para la complejidad del proceso.

La figura 7 muestra el aspecto que tiene la interfaz. Tiene un primer *login* para que cada usuario tenga su propia carpeta y que no se mezclen sus resultados con los de los demás. Si ustedes ya tienen un usuario en el programa Jaguar pueden usar el mismo. Y si no lo tienen se pueden dar de alta con un simple formulario, como se muestra en la figura 8. La figura 9 muestra la forma en la que se disponen los botones y cómo se apilan los archivos con cada uno de los procesos. Cuando uno somete un archivo a un determinado proceso en este programa, el resultado se crea en un nuevo fichero a cuyo nombre se agrega, como un sufijo, el tipo de proceso al que se ha sometido. Así, por ejemplo, si un archivo se llama "terms.txt", y sometemos el archivo al proceso "TAGGER", el resultado será otro archivo que se llamará "terms.txt_TAGGED".



Figura 7: Página de inicio del programa.

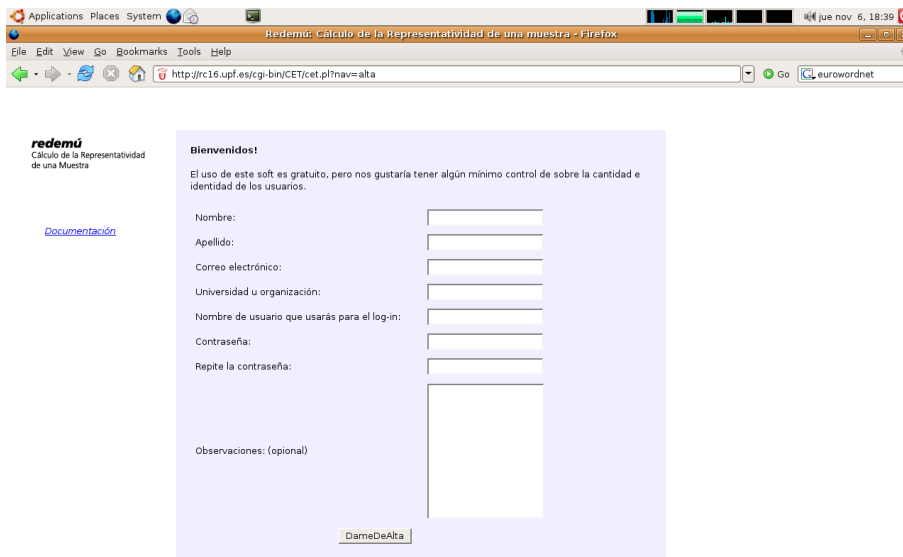


Figura 8: Formulario de registro.

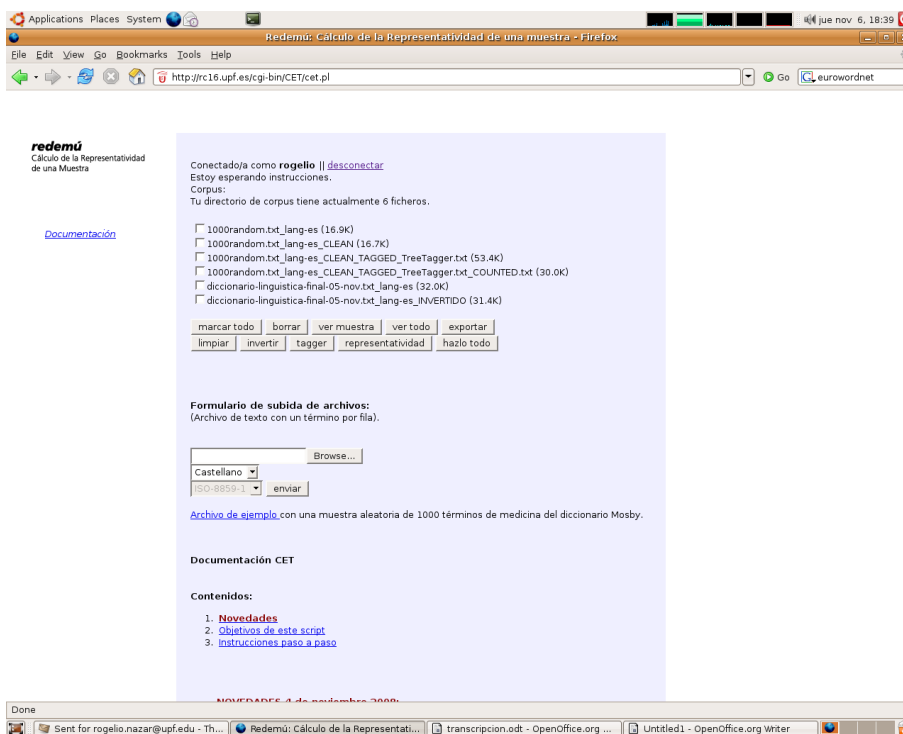


Figura 9: Disposición de botones y presentación de ficheros.

En primer lugar hay que seleccionar una lista de términos en formato txt. Este es el aporte del usuario. Es importante resaltar que se trata de un archivo en formato texto, porque cuando uno que no es informático dice "un archivo de texto" lo que se imagina es un archivo de Word o un pdf, etc. Debería haber sido nuestra previsión tener un conversor automático de Word o pdf a txt, y lo

agregaremos. Pero de momento, lo que pedimos es un archivo de texto ASCII que tenga un término por línea, en la codificación de caracteres iso-8859-1, que es la más común. Es importante que los términos estén libres de toda información adicional, como marcas de flexión, etc.

La manera de subir el archivo es con un botón de *upload*, seleccionándolo desde nuestro ordenador local. Debemos seleccionar manualmente la lengua entre las tres posibles, que son catalán, castellano o inglés (este reconocimiento de la lengua también será automático en un futuro). Entonces el archivo pasa a visualizarse en la interfaz del programa.

En esta interfaz vemos varios botones, que son las que anteriormente llamé "Funciones de soporte", que veremos más tarde. Entre los botones más importantes tenemos el "tagger", que es el que hace el etiquetado morfológico de los términos, y el botón "representatividad", que es el que dice cuál es el tamaño mínimo de la muestra. La secuencia de pasos a seguir es: subir el archivo al servidor, hacer el etiquetado morfológico y luego el análisis de la representatividad.

Hay una novedad de último momento respecto a este programa. Si bien tiene una serie de funciones distintas que le da una gran capacidad de juego, el coste de esto es que el programa es más complicado de usar y entonces, examinando el comportamiento de los usuarios, hemos decidido poner un botón que se llama "Hazlo todo". Así, el modo de uso se limita a llevar la flecha del mouse y hacer clic en ese botón. Si toco este botón, el programa va a hacer cada uno de los pasos de manera secuencial. Hace la representación de la curva de crecimiento, hace mil veces el ensayo y finalmente dice si se alcanza la representatividad y con qué tamaño.

Hay otro cambio de último momento respecto a la nomenclatura elegida para las categorías morfológicas. Recordarán que vimos antes que los patrones morfológicos corresponden a la estructura interna de los términos, cuando la nomenclatura que habíamos elegido inicialmente no era tal, porque tenía una sola categoría para la totalidad del término. Veamos un ejemplo. En el caso de nombres, que pueden ser femeninos o masculinos, como el término *pulso periférico*, que inicialmente tenía la etiqueta NA, por Nombre+Adjetivo, y ahora tiene la etiqueta "m", que es la que corresponde a "sintagma nominal masculino". Lo que hemos hecho es agregar una especie de traductor interno que analiza la estructura del término, identifica el núcleo (dependiendo de la lengua) y asigna, con cierto margen de error, por supuesto, la categoría que corresponde. La única cosa que todavía no podemos resolver es, en cuanto a la categoría "verbo", que sean transitivos, intransitivos o pronominales, porque es un problema algo complejo. Es seguro que hay marcas formales que pueden ayudarnos a clasificar automáticamente un verbo en estas categorías. Lo que ocurre es que para poder hacerlo necesitamos analizar los términos en sus contextos de aparición. De momento asignamos la categoría "v" para todos y quedará en la responsabilidad de cada uno completar esa información, si realmente es relevante.

Esto es todo en cuanto al modo de empleo. Antes era más complicado, había que ir haciendo una serie de cosas, y ahora se ha simplificado considerablemente.

Pregunta de una asistente: ¿Qué tenemos que hacer si el programa nos dice que nuestra muestra no es representativa?

Respuesta: Repetir el experimento con un tamaño de muestra más grande.

J. Vivaldi: Recordemos que el tagger es aquí una fuente potencial de error. Como tenemos sólo un

término por línea, el tagger no tiene contexto y entonces puede darse el caso de que haya términos que estén mal etiquetados.

Pregunta de otra asistente: Si nosotros ya tuviésemos el texto etiquetado, ¿se podría subir directamente ese texto?

J. Vivaldi: Sí, pero el programa no tiene previsto aceptar texto etiquetado.

R. Nazar: Se podría siempre y cuando el formato sea el mismo que espera el programa, es decir, una tabla donde hay un término en cada línea y la etiqueta morfológica separada por un tabulador. Hay una función prevista que es la de la mayoría de los usuarios y después hay una serie de cosas que se podrían hacer.

J. Vivaldi: Lo que sí está previsto es que corrija manualmente la lista de términos y la vuelvas a subir.

R. Nazar: El tema de los errores en el etiquetado ha generado una nueva línea de investigación que es un nuevo tagger que está basado en el TreeTagger de la Universidad de Stuttgart y en FreeLing, de la Universidad Politécnica de Cataluña. Lo que hace este nuevo etiquetador es supervisar el trabajo de los dos anteriores y cuando ve que los dos no están de acuerdo y dan informaciones distintas, entonces identifica que se trata de un caso problemático; busca en internet contextos de aparición del caso en cuestión y les proporciona esos contextos a los etiquetadores para que tengan más información. Esto hace que los etiquetadores aumenten su precisión porque resuelve este problema de que los etiquetadores estén trabajando sobre términos fuera de contexto. De todas maneras esto ya es otra investigación.

Funciones de soporte

Las funciones de soporte son un conjunto de simples reglas de transformación. Es decir que cuando el programa recibe términos que están en un formato que consideramos inadecuado (porque producen ruido en la fase de etiquetado) los convierte automáticamente al formato correcto. Estas marcas inadecuadas pueden ser por ejemplo marcas de flexión, que el programa eliminará, tal como muestra el ejemplo de la figura 10.

Acelerador/a => Acelerador
Acelerador -a

Figura 10: Eliminar marcas de flexión.

A veces algunos usuarios en lugar de subir una lista de términos con un término por fila nos daban una lista de términos separados por comas. Entonces el programa cuando ve esto convierte automáticamente las comas en retornos de carro, es decir que el resultado es un término por línea (figura 11). Todo lo que está entre paréntesis también va a ser eliminado por el programa (figura 12).

Activar, habilitar, permitir, poner en servicio => Activar
habilitar
permitir
poner en servicio

Figura 11: Separar de términos en líneas.

Actualizar (diferentes versiones), mejorar => Actualizar
mejorar

Figura 12: Eliminar de paréntesis.

También tenemos la función Invertir, que invierte el orden de los componentes de un término cuando están separados por una coma o por cualquier otro caracter que el usuario elija (figura 13).

Acuático, alimento
Agro-acuícola, explotación => Alimento acuático
Explotación agro-acuícola

Figura 13: Función "Invertir".

Conclusiones

Repasando los puntos principales de este seminario, tenemos por un lado la selección de una muestra mínima con un fundamento teórico. Esto no es un tema que esté resuelto en lingüística. Es un tema abierto, muy complejo, hay mucha investigación al respecto y todavía tenemos mucho que estudiar y aprender. Por otro lado, hay que destacar otra vez que si bien estamos tratando un tema muy específico, el muestreo de términos, también lo podemos exportar a cualquier otro problema de muestreo. Finalmente, resaltar la ventaja que significa una interfaz gráfica y amigable.

Como trabajo futuro, comentábamos antes sobre esta línea de un etiquetador morfológico que sea un poco más inteligente y que utilice el trabajo de los otros dos. Esto es un tema realmente urgente porque el trabajo de revisar manualmente el resultado de un etiquetado automático es muy laborioso. Por lo menos deseáramos tener un programa que diga "estos son los casos en los que tengo dudas", en lugar de que te obligue a revisar todo el corpus. Esta es una línea de investigación abierta y que me parece vale la pena continuar.

Eso es todo, gracias por vuestra atención y paciencia.

Pregunta de una asistente: Yo estoy analizando verbos y la variable que tengo asociada a esos verbos no es la categoría morfológica que sale automáticamente. Me refiero a variables sintácticas, papeles temáticos, etc. Entonces mi pregunta es si yo también puedo usar Redemú en este caso.

Respuesta: La pregunta clave en este caso es cuál es la cantidad de valores distintos que pueden adoptar tus verbos. Si las posibilidades son solo cinco o diez, es muy probable que agotes esa variabilidad muy rápido. Esto está pensado para un tipo de valor que es indeterminado, es decir que no se sabe ni cuántos ni cuáles son los valores que puede adoptar la variable.

Pregunta de otra asistente: En el caso de tener más de una variable asociada a cada individuo, es decir, una tabla con más de una columna, ¿todavía se podría utilizar el programa?

Respuesta: Sí, el programa acepta tablas con más de una columna. Pero lo que hace en este caso es juntarlas todas, es decir que al final a cada individuo le tocará un solo valor que será la combinación de los valores que se le da en cada una de las columnas.

Bibliografía Recomendada

(Sobre crecimiento del vocabulario y muestreo de corpus).

- BIBER, D. (1993). "Representativeness in Corpus Design". *Literary and Linguistic Computing*, Vol. 8 (4): 243-257.
- HEAPS, H. S. (1978) "Information Retrieval: Computational and Theoretical Aspects". New York, Academic Press.
- HERDAN, G. (1964); "Quantitative Linguistics". Washington. Butterworths.
- McENERY, T y WILSON, A. (2001). "Corpus Linguistics: An Introduction". Edinburgh University Press, 2001.
- MULLER, C. (1973); "Estadística Lingüística", Madrid, Gredos.