Manual
de informática y de tecnologías
para la traducción

Mikel L. Forcada
Felipe Sánchez Martínez
Juan Antonio Pérez Ortiz

Dep. de Llenguatges i Sistemes Informàtics
Universitat d'Alacant
E-03071 Alacant

{mlf,fsanchez,japerez}@dlsi.ua.es

http://www.dlsi.ua.es/~mlf
http://www.dlsi.ua.es/~fsanchez
http://www.dlsi.ua.es/~japerez

Edición 0.9.2, septiembre de 2017
Primera edición (0.9), febrero de 2016

Índice general

Capítulo 1

Introducción

Estas páginas cubren la mayor parte de los contenidos[1] de la asignatura *Tecnologías de la Traducción* que cursará el alumnado de segundo curso del grado en Traducción e Interpretación de la Universitat d'Alacant; también pueden ser útiles para asignaturas similares en otras universidades (por eso se ha incluido material más avanzado que no se estudia en Tecnologías de la Traducción). La lectura de este manual —que puede incluso contener algún error no detectado— no puede nunca sustituir el estudio de otros libros sobre la materia, algunos de los cuales se citan en este texto y se listan en la bibliografía.

Los contenidos de este manual se pueden dividir en dos partes: la primera presenta algunos conceptos básicos de la informática (capítulo 2) y de Internet (capítulo 3), sobre la entrada y el procesamiento de textos (capítulo 4) y sobre las bases de datos (capítulo 5); la segunda es una introducción a algunos aspectos generales de la traducción automática (capítulos 6 a 9) y a la traducción asistida por ordenador con memorias de traducción (capítulo 10). Finalmente, un apéndice discute la problemática de la traducción español–catalán y algunos de los sistemas existentes para este par de lenguas; esta información puede servir como ilustración en un caso concreto de lo que se ha estudiado sobre traducción automática. Los contenidos de esta tercera parte son, por lo tanto, complementarios.

Este manual se puede mejorar mucho, e iremos publicando versiones nuevas. Además, seguro que hay errores que deben corregirse. El texto está abierto, por supuesto, a sugerencias y a correcciones que lo hagan más útil, tanto para el alumnado de la asignatura como para otras personas que quieran saber sobre el tema. De hecho, aprovechamos para dar las gracias a todas las personas (alumnado, profesorado, etc.) que, con sus comentarios críticos, han ido mejorando este texto.[2] Son demasiada gente para mencio-

[1] Hay contenidos —mejor dicho, habilidades— que se aprenden como parte de las sesiones de laboratorio y que no figuran en este documento.

[2] Este libro está basado en una obra anterior, (Forcada y Pérez-Ortiz 2009), usada para la

narlos a todos, pero no queremos acabar sin agradecer las aportaciones de Raül Canals y Marote, que corrigió errores de versiones anteriores e hizo aportaciones en la parte de conceptos básicos de la informática, de Gema Ramírez Sánchez, particularmente en el capítulo de memorias de traducción, y de Sandra Montserrat, en la discusión sobre divergencias lingüísticas español–catalán del apéndice.

Los ficheros fuente (LaTeX, .eps, etc.) necesarios para volver a generar el libro están disponibles en un *repositorio* público,[3] de forma que, si lo deseáis, los podéis modificar para generar un texto nuevo y publicarlo vosotros, pero siempre de acuerdo con las condiciones de la versión 3 de la Licencia General Pública de GNU[4] o de la licencia Creative Commons Reconocimiento-CompatirIgual 4.0 Internacional.[5] Estas licencias os obligan a publicar cualquier trabajo derivado de éste con la misma licencia. Así garantizamos que nuestro trabajo está siempre accesible para cualquier persona que lo considere útil para la enseñanza o el estudio personal. Si este es vuestro caso, os estaremos muy agradecidos si nos mandáis un mensaje de correo electrónico diciéndonos para qué asignatura lo estáis usando a `fsanchez@dlsi.ua.es`.

Este manual es una traducción; podéis descargar la versión original en catalán desde `http://rua.ua.es/dspace/handle/10045/53085`; en caso de duda la versión de referencia es el original en catalán. Queda pendiente la traducción al español de muchos de los ejemplos que, a lo largo del libro, usan la lengua catalana, aunque algunos ya han sido traducidos.

La traducción del original en catalán se llevó a cabo a partir de las memorias de traducción generadas por las alumnas Claudia Martínez Agulló, Ana Isabel Moreno Delgado, María Pérez Saura, Aída Rivera Soler, Saray García Martínez, Marina Berenguer Serrano, Aida Moreno Tremiño e Irene Pérez Navarro. Gracias a todas. Gracias también a Celia Millán Beltrán, Clara García Abiétar, José Hernández Conca, Carole Poncin, Lucia Mazón Miñano y Amanda De Craecker por detectar algunas de las erratas que contenía la primera edición de esta traducción. Si detectáis alguna ota errata o error de traducción os estaremos muy agradecidos si nos lo hacéis saber mandando un mensaje de correo electrónico a `fsanchez@dlsi.ua.es`. Cualquier otra sugerencia para mejorar la traducción a español también será bienvenida.

---

licenciatura en Traducción e Interpretación.

[3] Repositorio GitHub: `https://github.com/mlforcada/llibre-tecnol-trad`

[4] Descrita en `http://www.gnu.org/licenses/gpl-3.0.txt`

[5] Descrita en `http://creativecommons.org/licenses/by-sa/4.0/`

Capítulo 2

Ordenadores y programas

Todos los sistemas informáticos[1] se pueden dividir en dos partes: *hardware* y *software*.

Hardware: el equipo físico que se puede ver y tocar. Por ejemplo, la pantalla, el procesador central, el teclado, el ratón, los chips[2] de memoria y las impresoras.

Software: uno o más *programas* (y los datos asociados) que realizan alguna función útil para el usuario o para otro *programa*. Por ejemplo, un procesador de textos como LibreOffice o Microsoft Word puede estar compuesto por más de un *programa*. Un *programa* es una secuencia (lista o conjunto ordenado) de instrucciones que el hardware sigue o ejecuta, de forma que realizan alguna tarea determinada.[3] Normalmente, los ordenadores están organizados alrededor de un *procesador central* (véase más adelante) que es capaz de comprender y ejecutar instrucciones básicas tomadas de un conjunto determinado (el *conjunto de instrucciones* del procesador). Los programas pueden estar guardados en un disco o cargados en la memoria del ordenador mientras el procesador los ejecuta.

A continuación se considerarán el hardware y el software con más detalle.

[1] Es decir, todas las instalaciones basadas en ordenadores

[2] El chip es el elemento básico de la microelectrónica y de la microinformática; se trata de uno o más circuitos integrados en una placa de silicio de dimensiones muy reducidas, que normalmente se coloca en una caja hermética con contactos metálicos.

[3] El uso de la palabra *programa* en informática (secuencia de operaciones o acontecimientos) es paralelo a muchos usos de esta palabra en la vida cotidiana: programa *de fiestas*, *de un concierto*, *de la lavadora*, etc.; aunque para el usuario un programa de ordenador es más parecido a una especie de caja de herramientas para hacer una tarea determinada, como, por ejemplo, editar un documento de texto.

2.1. Hardware

Todos los sistemas informáticos tienen hardware de las siguientes clases:

Procesamiento: Los dispositivos de procesamiento son los que hacen realmente el trabajo. La mayoría de los sistemas contienen un CPU (*central processing unit*, unidad central de procesamiento), o sencillamente, un *procesador* que es el responsable de ejecutar todas las instrucciones de programa, de procesar datos, y de controlar el funcionamiento de otros componentes del hardware. En los ordenadores personales, la unidad central es un único chip de silicio. Además, la mayoría de los sistemas actuales contienen también una GPU (*graphics processing unit*, unidad de procesamiento de gráficos), una CPU especializada en el tratamiento de imágenes, pero que también se puede usar para otras tareas computacionalmente intensivas.

La velocidad a la que una CPU ejecuta las instrucciones básicas de un programa se mide en megahercios (MHz) o gigahercios (GHz; un gigahercio son 1000 megahercios). Un megahercio equivale a un millón de hercios (Hz), es decir, un millón de ciclos de procesamiento de información por segundo. Cada ciclo de procesamiento de información se corresponde con un *tic* del reloj que todos los dispositivos de procesamiento tienen para sincronizar todos los circuitos del ordenador. Normalmente, una instrucción requiere unos cuantos ciclos de procesamiento para ser ejecutada, aunque algunos sistemas son capaces de procesar más de una instrucción al mismo tiempo. La velocidad típica de la CPU de un ordenador en la actualidad es de 3 Ghz.

Almacenamiento: los dispositivos de almacenamiento se pueden dividir en dos grupos:

Memoria primaria: memoria rápida de corto plazo, volátil (se borra cuando se apaga el ordenador), que sirve para guardar los programas y datos mientras el ordenador está en funcionamiento; si los programas y los datos no caben en la memoria primaria, el sistema operativo —véase el apartado 2.2— se encarga de copiarlos desde la memoria al disco duro cuando no se están usando y copiarlos de vuelta desde el disco duro a la memoria cuando son necesarios, operación que se llama *intercambio*.[4] La memoria primaria normalmente consiste en chips RAM (*random-access memory*, memoria de acceso aleatorio[5]) de silicio.

[4] En inglés *swapping*. Como el disco duro es más lento que la memoria primaria, el intercambio hace que el ordenador vaya más lento: por eso, ampliar la memoria primaria suele hacer que el ordenador vaya más rápido.

[5] Nótese la diferencia entre *acceso aleatorio* (a voluntad) y *acceso secuencial*. Un CD-ROM

Memoria secundaria: memoria de largo plazo, permanente. Ejemplos: los antiguos disquetes, discos fijos o duros internos y externos, memorias USB (también llamadas *pendrive*) y diversas formas de ROM (*read-only memory*, memoria de sólo lectura), como los chips ROM, los CD-ROM o los DVD.

Los discos fijos (y los antiguos disquetes) son dispositivos de almacenamiento magnético, más o menos como lo eran los antiguos casetes. La información se almacena haciendo servir las propiedades magnéticas de determinados materiales magnetizables. En la actualidad el tamaño típico de un disco fijo es de 500 GB o 1 TB (véase el apartado 2.3 para conocer las medidas de almacenamiento de información).

La memoria USB es un dispositivo de memoria flash, un chip de memoria que mantiene su contenido en ausencia de alimentación, que se conecta al puerto USB del ordenador. El tamaño de estas memorias puede llegar hasta 1 TB, a pesar de que los tamaños más típicos son 16, 32 y 64 GB.

La memoria ROM suele estar hecha de chips de silicio. Los CD-ROM (*compact dics read-only memory*) —idénticos en apariencia y similares en muchos aspectos a los CD de música— almacenan la información ópticamente.[6] El tamaño de un CR-ROM suele ser de 650 MB o de 700 MB.

El DVD (*digital versatil dics*)[7] es un tipo más avanzado de sistemas de almacenamiento basado en discos ópticos; básicamente, se trata de un CD más rápido y con más capacidad, que ha desplazado casi completamente a los CD-ROM. El tamaño de un DVD depende del tipo de DVD y suele estar entre 4,7 GB y 17 GB.

La manera más común de almacenar los datos en una memoria secundaria es organizarlos en *ficheros* o *documentos* organizados en *directorios* o *carpetas*; la sección 2.4 explica estos conceptos en detalle.

Entrada: La función primaria de los dispositivos de entrada es que el usuario pueda interactuar con la máquina y con los programas que ejecuta con el fin de *introducir* datos o información. Los dispositivos de entrada más comunes son el teclado, el ratón, la pantalla táctil, la palanca

---

de música es de acceso aleatorio porque podemos acceder a la decimosexta canción directamente; en cambio, un casete (cinta magnética) es de acceso secuencial porque para acceder a la decimosexta canción directamente tenemos que pasar por las 15 anteriores.

[6] Otros términos habituales son CD-R (*compact disc recordable*) —que identifica los CD en los que se puede escribir información solo una vez con la ayuda de dispositivos conocidos como grabadoras— y CD-RW (*compact disc rewritable*) —utilizado para los CD que se pueden borrar y reescribir un número ilimitado de veces.

[7] Como en el caso de los CD, podemos hablar de DVD-ROM, DVD-R y DVD-RW.

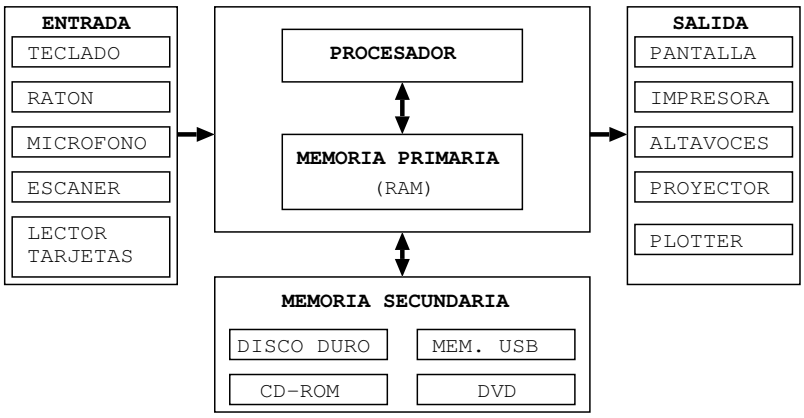


Figura 2.1: Esquema del hardware de un ordenador.

de mando (*joystick*), las cámaras de fotos y *webcams* o el escáner —un dispositivo que lee una imagen impresa y la convierte en un fichero (véase el apartado 2.4) que contiene la imagen digitalizada.[8]

Salida: Esta es la familia de los dispositivos que el ordenador usa para comunicar datos o información al usuario. El monitor (la pantalla) es el más común. Otros dispositivos de salida son las impresoras, los altavoces, los vibradores de los dispositivos móviles, etc.

En la figura 2.1 se resume esquemáticamente el hardware de un ordenador.

2.2. Software

Hay tres tipos básicos de software:

Sistemas operativos y *firmware*: son los programas que permiten el funcionamiento básico del ordenador. Se denomina *firmware* al software del sistema que se usa tan frecuentemente que se almacena permanentemente en chips ROM. Este software ofrece al sistema operativo servicios básicos de acceso a los dispositivos de entrada y de salida más habituales.

[8]Cuando la imagen es la de un texto impreso, un programa de *reconocimiento óptico de caracteres* (OCR, *optical character recognition*) la puede convertir en una representación del texto adecuada para ser manipulada con un procesador de textos (véase la sección 4.6), generalmente con algunos errores tipográficos menores.

Cuando conectamos el ordenador, el primer programa en ejecutarse es el *firmware*, el cual se encarga de hacer algunas comprobaciones, como por ejemplo que hay un teclado conectado al ordenador o que la memoria RAM no tiene defectos, y de cargar el sistema operativo.

El sistema operativo, por un lado, permite que el usuario ejecute programas y gestione los ficheros de datos, etc.; para eso, ofrece una *interfaz de usuario* (véase más adelante). Por otro lado, el sistema operativo ofrece servicios básicos (véase más abajo) a los programas de aplicación que se ejecutan en el ordenador (los cuales pueden tener su propia interfaz de usuario).

En cuanto a la *interfaz de usuario*, es decir, la apariencia y la forma de interaccionar con el usuario, la mayoría de los sistemas operativos son *gráficos*, es decir, basados en ratón o pantalla táctil, punteros, ventanas, etc. (GNU/Linux, Windows, MacOS, iOS, Android); antiguamente, los sistemas operativos eran de *línea de órdenes*, es decir, basados en texto (Unix primigenio, MS-DOS).

Los sistemas más antiguos eran a veces *monousuario* (MS-DOS, Windows 3.11), es decir, sólo podían ser usados por un único usuario a la vez, o *monotarea*, es decir, no podían ejecutar más de un programa al mismo tiempo. La mayor parte de los actuales sistemas operativos son *multiusuarios*, es decir, pueden ser usados por más de un usuario a la vez, y *multitarea* (GNU/Linux, versiones recientes de Windows, MacOS). La mayor parte de los sistemas operativos actuales están además preparados para interaccionar con otros dispositivos a través de diferentes tipos de *redes*.[9]

Algunas de las operaciones básicas que hacen los sistemas operativos son:

- Controlar el hardware del ordenador donde se ejecutan.
- Copiar, mover y borrar ficheros de datos.
- Crear, mover y borrar directorios de ficheros.
- Establecer conexiones entre ordenadores.
- Ejecutar programas y controlar su ejecución.
- Establecer conexiones con otros ordenadores o dispositivos en red.

De hecho, los programas de aplicación suelen estar escritos para ser ejecutados *sobre un sistema operativo*, es decir, los programas de apli-

[9] La organización de los ordenadores en una red local permite la comunicación de información entre ellos y que se compartan recursos, como por ejemplo una impresora. Internet (véase el capítulo 3) no es más que una gran red global que interconecta muchas redes más locales.

cación *asumen* que el sistema operativo hará todas estas operaciones sencillas y no contienen instrucciones de programa para hacerlas, sino sólo instrucciones para invocar los programas correspondientes del sistema operativo, cosa que simplifica enormemente la escritura de programas por parte de los programadores. Por eso, cuando se especifican las características de un programa de ordenador se tiene que decir para qué sistema operativo está escrito, puesto que cada sistema operativo ofrece servicios diferentes e interacciona de manera diferente con los programas de aplicación.

Programas de aplicación: software diseñado específicamente para satisfacer las necesidades de los usuarios (a veces se denominan simplemente *aplicaciones*). Se podrían hacer dos grupos:

Software de uso específico: software diseñado para un usuario muy concreto con unas necesidades muy específicas: por ejemplo, el programa que gestiona los préstamos, las cuotas y las adquisiciones de un videoclub, hecho a medida para él.

Software específico para profesionales de la traducción: sistemas de traducción automática (capítulos 6 a 9), sistemas de traducción asistida basados en memorias de traducción (capítulo 10) y bases de datos terminológicas (capítulo 5).

Software de uso general: software diseñado para hacer tareas más genéricas, interesantes para muchos tipos de usuarios. Aquí tenéis algunos ejemplos:

Editores y procesadores de texto para preparar, modificar, almacenar e imprimir documentos de texto (véase la sección 4.6).

Hojas de cálculo que permiten automatizar cálculos que se repiten sobre un conjunto más o menos grande de datos (por ejemplo, para calcular la nota media de cada estudiante de una clase entera a partir de las notas parciales), y presentar los resultados de varias maneras, por ejemplo, en gráficos de muchos tipos.

Gestores de bases de datos que sirven para almacenar, organizar y gestionar de varias maneras la información contenida en *bases* o bancos de datos (véase el capítulo 5).

Navegadores de Internet: programas que permiten acceder de manera sencilla a los documentos de Internet en máquinas conectadas a esta red.[10] Véase la sección 3.5.

Juegos de muchas clases.

[10]El nombre *navegador* se usa por la analogía —débil— existente entre los mecanismos de acceso a los documentos de Internet y la navegación mediante un mapa en una zona desconocida.

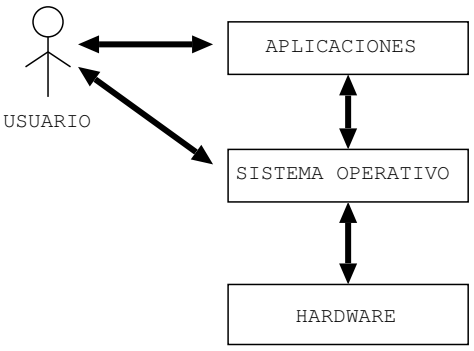


Figura 2.2: Esquema de la interacción entre el usuario, el sistema operativo y los programas de aplicación.

Los programas de aplicación los activa el usuario por medio del sistema operativo, utilizan el sistema operativo para acceder a los recursos del sistema (hardware y otros programas) e interaccionan con el usuario mediante los dispositivos de entrada y de salida (véase la figura 2.2).

Para saber más sobre software

Como ya se ha dicho más arriba, un software es un conjunto de programas, cada uno de los cuales consiste en una lista de instrucciones válidas (ejecutables por el ordenador) que se ejecutan en el orden indicado, de la primera a la última, excepto cuando se presenta alguna instrucción de *salto* que indica cuál es la siguiente instrucción que se tiene que ejecutar.

Por ejemplo, un programa que suma todos los números enteros del 1 al 10 podría ser el siguiente, el cual usa dos posiciones de memoria RAM para guardar valores necesarios para el cálculo. Cada una de las órdenes se corresponde con una instrucción básica de las que puede entender cualquier procesador.

1. Haz que el acumulador (un registro de la memoria interna del procesador) valga 1.
2. Guarda el valor del acumulador en una posición de memoria que denominaremos *índice*.
3. Haz que el acumulador valga 0.
4. Guarda el valor del acumulador en una posición de memoria que denominaremos *suma*, la cual contendrá la suma total.
5. Carga el valor de *suma* en el acumulador.
6. Suma el valor de *índice* al acumulador.
7. Guarda el valor del acumulador en *suma*.

8. Carga el valor de *índice* en el acumulador.
9. Compara el valor del acumulador con 10.
10. Si es igual, salta a la instrucción 14
11. Incrementa en 1 el valor del acumulador.
12. Guarda el valor del acumulador en *índice*.
13. Salta a la instrucción 5.
14. Para.

Muchas veces se usan nombres cortos (en inglés *mnemonics*) para las instrucciones del procesador y también nombres elegidos por el programador para referirse a posiciones del programa (esta notación se suele denominar *lenguaje ensamblador*). El programa de arriba tendría la apariencia siguiente:

```
         mov #1,A
         mov A,index
         mov #0,A
         mov A,suma
 otro:   mov suma,A
         add A,index
         mov A,suma
         mov index,A
         cmp A,#10
         jeq final
         inc A
         mov A,index
         jmp otro
 final:  hlt
```

Procesadores de lenguajes de programación: las instrucciones que ejecuta el procesador central de un ordenador son demasiado sencillas para que un programador humano haga programas útiles; sería largo y engorroso, como hemos visto en el ejemplo de programa que sumaba los enteros del 1 al 10. Los programadores normalmente escriben sus programas en *lenguajes de programación* basados en instrucciones más potentes (como por ejemplo BASIC, Java, C, C++, Pascal, Perl o Python) y usan programas especiales —los procesadores de lenguajes— para traducirlos a las instrucciones sencillas que entiende la máquina.[a] Casi todos los programas que se ejecutan en un ordenador han sido escritos en algún lenguaje de programación. El programa que suma los números del 1 al 10 quedaría así en el lenguaje Pascal:

```
program SUMA;
var
   index, suma: integer;
begin
   suma:=0;
   for index:=1 to 10
      suma:=suma+index;
end.
```

[a] Hay dos familias básicas de procesadores de lenguajes: los *compiladores*, que traducen todo el programa al lenguaje de la máquina antes de ejecutarlo, y los *intérpretes*,

que leen el programa línea a línea y ejecutan pequeños programas ya escritos en el lenguaje de la máquina y que se corresponden con las sentencias del lenguaje de programación.

2.3. Memoria

Toda la información — instrucciones de programa o datos — que se almacena en la memoria de un ordenador se guarda en forma binaria, es decir, cada dato es una cadena de dígitos binarios o *bits*. Un bit puede tener dos valores: 0 (apagado, inactivo) o 1 (encendido, activo); esto es porque el dispositivo electrónico correspondiente puede estar en dos estados. Si necesitamos guardar objetos o unidades de información que tienen más de dos valores posibles, no tendremos suficiente con 1 bit; tendremos que combinar más de un bit. Por ejemplo, si tenemos una unidad de información que puede presentarse en 778 formas diferentes,[11] necesitaremos 10 bits, porque con 9 bits solo podemos hacer

$$2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 2^9 = 512$$

combinaciones diferentes, pero con 10, ya podemos hacer suficientes, porque $2^{10} = 1\ 024$ (quedarían $1\ 024 - 778 = 246$ combinaciones sin usar).

Los bits se agrupan normalmente en grupos de ocho, llamados *bytes*. Un byte puede estar, por lo tanto, en $2^8 = 256$ estados diferentes. Por ejemplo, los caracteres y símbolos más comúnmente usados en textos se guardaban históricamente cada uno en un byte, usando el código ASCII (*American Standard Code for Information Interchange*), donde el código de la "`A`" es "`01000001`" o el de la "`z`" es "`01111010`" (véase el epígrafe 4.1). El código ASCII fue el primer código estándar para almacenar textos; cuando los textos son más ricos y contienen información sobre tipos y tamaño de letra, diagramación, notas a pie de página, etc., se usan formatos más avanzados que se explican en el epígrafe 4.1. Un byte puede contener, por tanto, muy poca información (un carácter, una instrucción sencilla del procesador central, un número del 0 ("`00000000`") al 255 ("`11111111`"), etc.). Por ejemplo, un documento de texto como éste tiene decenas de miles de caracteres, y una enciclopedia, cientos de millones. En las imágenes en blanco y negro, cada punto es un bit, una pantalla de ordenador contiene más o menos un millón. Si son de colores, hay más de un bit por cada punto. Las instrucciones de los programas que ejecuta el procesador central también se almacenan en bytes.[12]

Como un byte puede contener poca información, normalmente se habla de:

[11] Cómo, por ejemplo, los signos de algún sistema de escritura no alfabético.

[12] En el ejemplo de la sección anterior, la instrucción `inc A`, que incrementa el valor almacenado en el acumulador en 1, podría ser el byte `11010110`.

- *kilobytes* (kB), o miles de bytes. De hecho, por fidelidad al sistema binario, un kilobyte no tiene 1.000, sino 1.024 bytes (2^{10} es 1.024), es decir, 1.024×8=8.192 bits.
- *megabytes* (MB), o millones de bytes. De hecho, como en el caso de los kilobytes, no exactamente:

$$1 \text{ MB} = 1.024 \times 1.024 \text{ bytes} = 1.048.576 \text{ bytes}.$$

- *gigabytes* (GB), o miles de millones —un poco más— de bytes:

$$1 \text{ GB} = 1.024 \text{ MB} = 1.048.576 \text{ kB} = 1.073.741.824 \text{ bytes}.$$

- *terabytes* (TB), o billones (millones de miliones) —de nuevo, un poco más— de bytes:

$$1 \text{ TB} = 1.024 \text{ GB} = 1.048.576 \text{ MB} = 1.073.741.824 \text{ kB} = \\ = 1.099.511.627.776 \text{ bytes}.$$

Puesto que los prefijos *k*, *M*, *G* y *T* se usan en el resto de las disciplinas científicas para expresar potencias exactas de 10 (de 1.000), hay quien prefiere hablar de *kibibytes* (kiB), *mibibytes* (MiB), *gibibytes* (GiB) y *tebibytes* (TiB) para referirse a las unidades de capacidad de almacenamiento basadas en múltiplos de 1.024.

2.4. Ficheros y directorios

Como ya se ha dicho en la página 5, es habitual que los datos —de cualquier clase: textos, instrucciones de programas, datos gráficos, sonido, vídeo, etc.— almacenados en la memoria secundaria estén organizados en *ficheros*, también llamados *documentos* o *archivos*. Los ficheros son conjuntos de datos con un nombre que los identifica y que se manipulan —se abren, se cierran, se copian, se borran— como un todo. En discos grandes, sería muy incómodo tener todos los archivos uno tras otro, por lo que es normal que los archivos estén organizados en *directorios*, también llamados *carpetas*. Los directorios son ficheros especiales que agrupan los nombres y las características de otros ficheros; de hecho, los directorios pueden contener cero o más ficheros o también cero o más directorios (sin restricciones de cantidad), y así sucesivamente, de forma que la persona usuaria puede establecer una estructuración jerárquica o arbórea de sus ficheros en el disco.

Normalmente, cada disco tiene un *directorio principal* o *directorio raíz* (el más elevado en la jerarquía de directorios), dentro del cual se encuentra el resto de directorios. Dos ficheros — también dos directorios — sólo pueden tener el mismo nombre si se encuentran en directorios diferentes.

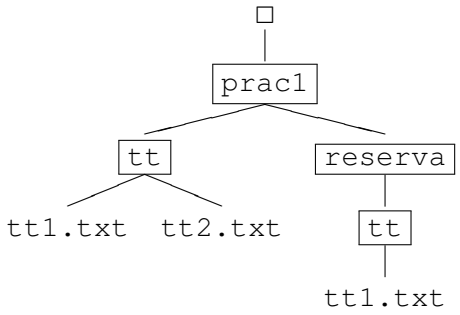


Figura 2.3: Ejemplo de estructura de ficheros y directorios en un dispositivo de almacenamiento. El directorio principal o raíz está representado por el símbolo □.

Por razones históricas, los nombres de ficheros suelen tener dos partes: el *nombre* propiamente dicho y la *extensión*, separadas por un punto (por ejemplo, `alacant.txt`). El nombre suele ser libre, pero la extensión suele ser corta (entre una y cuatro letras) y el sistema operativo suele usarla para identificar el programa que se debe usar para procesarlo o el formato en que se encuentran los datos que contiene (por ejemplo, la extensión `.txt` identifica normalmente un fichero de texto plano, véase el apartado 4.1; la extensión `.exe` se usa para los programas de ordenador, etc.).

La secuencia de los nombres de las carpetas que hay que ir abriendo hasta que lleguemos a un fichero se conoce como la *trayectoria* o *ruta* del fichero. De hecho, conviene considerar la trayectoria como parte del nombre del archivo, lo que nos permitiría decir, sencillamente, que en un disco no puede haber dos archivos con el mismo nombre.

Todos estos conceptos se ven quizás más claros con el ejemplo de la figura 2.3 en que se muestra la estructura de ficheros y directorios en un dispositivo de almacenamiento cualquiera. En este dispositivo, el directorio principal o raíz (representado con el símbolo □) contiene un único (sub)directorio `prac1`; este directorio contiene dos (sub)directorios, `tt` (que contiene los ficheros `tt1.txt` y `tt2.txt`) y `reserva`. El directorio `reserva` contiene un subdirectorio `tt` (que contiene el archivo `tt1.txt`). Fijaos que dos carpetas diferentes contienen archivos con el mismo nombre `tt1.txt`; esto no es problema si consideramos la trayectoria completa como nombre del fichero. Si el disco se llama `C:` (típico en el caso del disco duro de un PC con sistema operativo Windows), las trayectorias de estos dos ficheros serían `C:\prac1\tt\tt1.txt` y `C:\prac1\reserva\tt\tt1.txt`, y, por lo tanto, serían diferentes. En el caso del sistema operativo GNU/Linux las trayectorias de estos dos ficheros serían `/prac1/tt/tt1.txt` y `/prac1/reserva/tt/tt1.txt`. Fijaos que cada sistema operativo usa

un símbolo diferente para el directorio principal o raíz y para separar los nombres de los directorios y archivos dentro de la ruta o trayectoria.

2.5. Tipos de ordenadores

Una clasificación no muy exhaustiva de los diferentes tipos de ordenadores que podemos encontrar hoy en día es la siguiente:

De sobremesa (en inglés *desktop*): Están formados por *una caja* con dispositivos de procesamiento y almacenamiento y un conjunto de periféricos (dispositivos de entrada o de salida) como por ejemplo el teclado, el ratón o la pantalla. Son, con diferencia, los ordenadores más habituales.

Portátiles (en inglés *laptop*, aunque cuando son pequeños se les llama a veces *notebook* o *netbook*): Tienen un tamaño menor que el de un maletín y un peso ligero que permite llevarlos sin demasiado esfuerzo de un lugar a otro. Su volumen reducido limita las posibilidades de hacer ampliaciones y, por lo tanto, su tiempo de vida puede ser más corto que el de los ordenadores de sobremesa.

Tabletas y *smartphones*: las tabletas (en inglés *tablets*) y los teléfonos móviles más modernos, denominados *smartphones*, son verdaderos ordenadores portátiles, con una pantalla táctil y sin teclado, con cámara, conectividad Wi-Fi y Bluetooth, receptor GPS, etc. Suelen venir con un sistema operativo gráfico: el más común es Android, pero los de la marca Apple usan iOS.[13]

Servidores: Los servidores son ordenadores que contienen y gestionan información que se utilizará en otros ordenadores ("clientes") conectados a ellos a través de una red interna o a través de Internet; no son muy diferentes de un ordenador de sobremesa, aunque normalmente son más potentes en cuanto a memoria, disco y procesador, pero como nadie tiene que sentarse delante de ellos no suelen tener pantalla, teclado o ratón y están pensados para ser ubicados horizontalmente en armarios especiales llamados *racks*. Estos ordenadores se pueden presentar en grupos conectados entre sí para ofrecer mayor potencia y capacidad.

Respecto a los ordenadores de sobremesa y los portátiles, a menudo se hace la distinción entre los ordenadores de tipo PC y los Macintosh (a

[13] Estos dispositivos han desplazado a los antiguos *handhelds* o dispositivos de mano, que eran una evolución de las antiguas agendas electrónicas y se solían denominar *PDA* por *Personal Digital Assistant*, 'asistente digital personal'.

menudo denominados Mac). Los PC son la evolución de los primeros ordenadores personales desarrollados por IBM, a pesar de que actualmente son fabricados por un número muy grande de empresas. Los Mac, en la actualidad, también son ordenadores de tipo PC, pero antiguamente eran ordenadores tipo PowerPC fabricados exclusivamente por la empresa Apple. Los Mac usan un sistema operativo propio (MacOS) y tienen una cuota de mercado más reducida entre el público general pero más grande en determinadas aplicaciones especializadas (por ejemplo, el diseño gráfico).

2.6. Configuración típica de un ordenador personal

La configuración clásica de un ordenador personal de sobremesa de 2015 suele ser más o menos la siguiente:

- La unidad base (la "caja" o la "torre") contiene:
  - Un procesador compuesto de cuatro núcleos o procesadores individuales (*quad-core*) o más, como por ejemplo un *Intel Core i5* o superior o un procesador equivalente de la marca AMD (véase el glosario, apartado 2.7) a 3 GHz.
  - La memoria RAM (por ejemplo, de 8 GB).
  - Una buena tarjeta gráfica, con su propia unidad independiente de procesamiento gráfico o GPU (*graphics processing units*).
  - Un disco fijo con una capacidad del orden de 1 TB.
  - Una unidad grabadora de DVD y de CD-ROM.[14]
  - Una tarjeta de sonido con altavoces y micrófono.
  - Una cámara (a veces llamada *webcam*).
  - Una o más tarjetas de comunicaciones incorporadas (con cables o inalámbricas, véase el glosario, apartado 2.7).
- Un monitor o pantalla, normalmente una pantalla LCD[15] de 17 o más pulgadas,
- Un teclado aparte y un ratón.
- Una impresora (de inyección o de tinta —la más típica —, o láser[16]).

[14] En las unidades de CD-ROM es importante la velocidad máxima de transferencia de datos, que se da como múltiple del estándar (la de un CD de música, del orden de unos 150 kilobytes por segundo): cuádruplo (4×), séxtuplo (6×), etc. Actualmente no es extraño que una unidad de CD-ROM tenga una velocidad punta de lectura y de escritura de 52× o más. De todas formas, las velocidades *medias* de todo un proceso de lectura y escritura suelen ser más bajas.

[15] *liquid-crystal display* o pantalla de cristal líquido

[16] Las impresoras *matriciales* o de *agujas* sólo se usan para aplicaciones muy específicas.

Las especificaciones de los portátiles (memoria, procesador) suelen ser similares, normalmente un poquito más reducidas. Los teléfonos móviles inteligentes o *smartphones* y las tabletas no suelen tener disco, sino una memoria flash no volátil, por ejemplo, de 8 GB, y una memoria RAM del orden de 1 GB.

2.7. Un pequeño glosario

Este glosario recoge algunos términos de uso común en la descripción de ordenadores y programas que no han sido definidos más arriba.

Adaptador de vídeo (también llamado tarjeta gráfica o controlador de vídeo): Dispositivo (tarjeta independiente, o integrada en la placa base) que permite conectar un monitor al ordenador. Hay muchos tipos de adaptadores de vídeo. Se tiene que considerar la resolución , es decir, el número de puntos, elementos de imagen (*píxeles*) que caben en una imagen, por ejemplo 1024×768 (horizontal $\times$ vertical), la profundidad *de color* (en bits: por ejemplo 24 bits permiten $2^{24} = 16\,777\,216$ colores diferentes) y otros parámetros como la frecuencia *de refresco* (que se mide en hercios o ciclos por segundo; véase "megahercio"). Actualmente no es extraño tener en ordenadores de sobremesa o portátiles resoluciones como la llamada *HD 1080* (1920 $\times$ 1080) o incluso más grandes.

ADSL (del inglés *asymmetric digital subscriber line*, línea de abonado digital asimétrica): Versión asimétrica de DSL (véase DSL). La asimetría se refiere a que la velocidad de transmisión de datos de la central hacia el abonado es superior a la velocidad de transmisión de datos del abonado hacia la central (por ejemplo 8 Mb/s hacia el abonado y 512 kB/s hacia la central).

Cache o *memoria caché*: Memoria RAM intermedia, de acceso más rápido por parte del procesador, donde se copia de cuando en cuando un bloque (también llamado "página") completo de posiciones consecutivas de la memoria RAM general para acelerar los accesos a posiciones en la misma zona. Por ejemplo, en un ordenador con 512 kilobytes (524.288 bytes) de *memoria caché*, tras acceder a la posición 2.000.000 es muy probable que el procesador quiera acceder a la posición 2.000.003. Si cuando se ha pedido la 2.000.000 se copian en la *memoria caché* las 524.288 posiciones que van de la 1.572.864 a la 2.097.151, el acceso a la posición 2.000.003 será más rápido.

DSL (del inglés *digital subscriber line*, línea de abonado digital), tecnología de conexión que permite aprovechar las líneas telefónicas y eléctricas

para hacer conexiones de alta velocidad (hasta unos 10 Mb/s). En el caso de las líneas eléctricas, la tecnología recibe también el nombre de PLC (*power line communications* o comunicaciones a través de líneas de fuerza), pero en España no se usa para proveer servicios de Internet doméstico.

fibra óptica: tecnología que transporta los datos usando una luz láser que se propaga a través de un cable muy fino de material transparente. En el momento de escribir estas líneas, los proveedores de Internet han empezado a ofrecer un servicio doméstico de conexión que permite conexiones del orden de centenares de Mb/s.

GHz: véase gigahertz.

gigahertz: un gigahertz son 1000 megahertzs (véase *megahercio* en este glosario).

GNU/Linux: un sistema operativo multitarea y multiusuario gratuito, del estilo del Unix que se podía encontrar en los llamados *miniordenadores* de los años 70 y 80, desarrollado de manera colaborativa por miles de voluntarios independientes y por empresas en todo el mundo y que es *software libre* (véase la entrada en este glosario): se puede copiar libremente si se cumplen ciertas condiciones. Se puede instalar GNU/Linux (que se presenta en muchas *distribuciones* diferentes como por ejemplo *Ubuntu*, *Mint*, *Fedora*, etc.) en un PC con procesador de la familia x86 (véase *Pentium*) o superior y en otros muchos tipos de ordenador.

Macintosh o *Mac*: nombre genérico (y comercial) de una familia de ordenadores construidos por Apple Computer y que son básicamente equivalentes a los PC. Estos ordenadores, lanzados al mercado en 1984, popularizaron la interfaz gráfica de usuario, toda una revolución para la época. Hace unos años había diferencias significativas entre los PC y los *Mac* de manera que no eran compatibles, es decir, que los programas de uno no funcionaban en el otro, tenían que adaptarse a las características particulares de cada uno. Estas diferencias se debían al hecho de que el procesador del *Mac* no era de la familia x86 (véase *Pentium*), sino de otra (antiguamente de la familia 68000 de Motorola, y después del llamado PowerPC). En la actualidad esta diferencia no existe y tanto unos como otros emplean procesadores de la familia x86. En el caso de los *Mac* desde el año 2006 incorporan procesadores Intel, así que se puede instalar Microsoft Windows o GNU/Linux con todos sus programas, aunque también podemos usar el sistema operativo propio de los Mac, llamado *MacOS*.

megahercio: Un megahercio (MHz) es un millón de hercios (Hz), es decir, un millón de ciclos por segundo. La velocidad de las unidades centrales de los ordenadores se miden en MHz o GHz, es decir, en millones o millares de millones de ciclos básicos de procesamiento de información —correspondientes a los *tics* o impulsos del reloj que sincroniza todos los dispositivos del ordenador— por segundo. La ejecución de una instrucción por parte del procesador suele consumir un número pequeño de ciclos, casi siempre más de uno. Los modelos actuales pueden ejecutar, en determinadas circunstancias, más de una instrucción a la vez, lo que hace que a veces se ejecute una instrucción por ciclo o incluso más de una. Una velocidad típica en la actualidad es 3 GHz, es decir, 3000 MHz. Una velocidad más alta implica una velocidad de ejecución más alta, siempre que no haya otras circunstancias limitantes (por ejemplo, falta de memoria). Otros componentes, como la memoria RAM, también funcionan con una velocidad determinada, independiente de la del procesador, que se mide también en MHz.

MHz: véase megahercio.

módem: abreviatura de modulador-desmodulador. En el caso del módem más común, el *módem telefónico*, se trata de un dispositivo (normalmente una placa interna, aunque también puede ser externo) que permite usar la línea telefónica (señales analógicas) para comunicaciones informáticas (digitales) entre dos ordenadores, estableciendo una llamada; antiguamente era la manera estándar de acceder a Internet desde casa. Uno de los parámetros más interesantes de un módem es la *velocidad* de transmisión de datos, que se mide en b/s (bits por segundo). Una velocidad clásica en módems domésticos era 33.600 b/s (más recientemente, 57.600 b/s; las líneas telefónicas actuales podrían admitir velocidades alrededor de los 100.000 b/s). Esto permitía mandar una carta de una página en unas décimas de segundo pero no sería suficiente para la mayor parte de los usos actuales de Internet.

La palabra *módem* se puede usar también para otros tipos de módems, normalmente más rápidos: los *módems de cable*, que permiten conectar el ordenador a Internet a través de los cables de empresas especializadas que ofrecen televisión, teléfono e Internet, los *módems ADSL* (véase *ADSL* en este glosario), los *módems de fibra óptica* (véase *fibra óptica* en este glosario), etc.

Software libre: (*free software*, también llamado *software de código fuente abierto* u *open-source software*) es el software que se distribuye con licencias que dan una serie de libertades a quien recibe el software: la libertad de usarlo para cualquier propósito sin restricción, la libertad de examinarlo para ver como funciona y modificarlo para adaptarlo a un nuevo uso, y la libertad de distribuir copias —originales o

modificadas— libremente a quienes se desee. Para poder modificar el software, no es suficiente con tener acceso a la versión ejecutable en el ordenador: tenemos que tener acceso al llamado *código fuente*, es decir, a la versión del software que escriben y modifican las personas que programan (de ahí el nombre *de código fuente abierto*) y que después se convierte automáticamente en la versión ejecutable. Ejemplos de software libre son: el sistema operativo *GNU/Linux*, el navegador *Firefox*, o el procesador de textos *LibreOffice*. No se debe confundir *software libre* con *software gratuito* (o *freeware*). Hay softwares gratuitos que no son libres porque no otorgan todas las libertades (por ejemplo, el lector de PDF *Adobe Acrobat*, o el programa de telefonía por Internet *Skype*: por ejemplo, a pesar de tener el software ejecutable, no tenemos acceso a su código fuente).

Pentium: nombre genérico de una familia actual de procesadores centrales de la compañía Intel, los más recientes de la serie "x86" de procesadores que empezó con el 8086 a principios de los años 80, pasando por el 80286, el (80)386 y el (80)486.[17] Los nuevos procesadores tenían juegos de instrucciones más complejos y eran capaces de ejecutar los programas que ejecutaban los anteriores (por ejemplo, un Pentium puede ejecutar cualquier programa escrito para un 386) pero introducían mejoras que permitían ordenadores más rápidos, con mayor capacidad de cálculo, capaces de procesar más datos en cada instrucción (8, 16, 32 —a partir del 386—, y actualmente 64 bits) y de gestionar más memoria. Los Pentium más recientes tienen más de un *núcleo* o sub-procesador, y pueden, por lo tanto, ejecutar instrucciones de programa en paralelo.

placa de sonido: En los ordenadores más antiguos, se tenía que comprar a parte una placa (o tarjeta) de sonido si se quería usar el ordenador para procesar, grabar, reproducir, y manipular sonidos digitalizados. En la actualidad todos los ordenadores llevan estas capacidades incorporadas.

tarjeta de red: En los ordenadores más antiguos, para conectar ordenadores y formar una red (normalmente local) para compartir recursos, había que dotar a cada ordenador de una placa o tarjeta de red. Hay varios estándares de conexión en red; los más usados son Ethernet (para conexiones con cables) y *Wi-Fi* (véase *Wi-Fi* en este glosario).

USB (del inglés *universal serial bus*, bus serie universal): Estándar o norma de conexión de dispositivos periféricos (impresoras, módems, reproductores digitales de música, cámaras digitales, unidades de memoria) que transmite los datos en serie (es decir, un bit detrás del otro) a

[17] El nombre *Pentium* se eligió porque Intel no podía registrar "586" como marca.

velocidades que en las versiones más modernas del estándar pueden llegar a los Gb/s, y que permite la conexión y desconexión de dispositivos de muchas clases "en caliente", es decir, sin tener que apagar el ordenador.

Wi-Fi (probablemente del inglés *wireless fidelity*, fidelidad inalámbrica): tecnología de conexión inalámbrica (vía radio), principalmente para formar redes locales, y que en la actualidad (estándar IEEE 802.11ac, enero de 2014) permite velocidades de transmisión de hasta 6.77 Gb/s.

2.8. Cuestiones y ejercicios

1. ¿Cuántos *kilobytes* hay en un *gigabyte*?

   a) 1.024

   b) 1.073.741.824

   c) 1.048.576

2. Si una memoria USB tiene 6 *gigabytes*, una página de texto (europeo occidental) típica tiene 50 líneas de 60 caracteres (contando los blancos) y cada carácter ocupa 1 *byte*, ¿cuántas páginas caben aproximadamente en la memoria?

   a) 200

   b) 20000

   c) 2000000

3. Una persona conectada a Internet por teléfono observa que las velocidades de transferencia que le indica su navegador (véase el capítulo 3) varían alrededor de los 300 *kilobytes* por segundo. Una de estas tres *no* puede ser la velocidad de su servicio de ADSL:

   a) 1 Mb/s

   b) 6 Mb/s

   c) 4 Mb/s

4. ¿Cuál de estas afirmaciones es incorrecta?

   a) Los módems convierten información digital en señales analógicas pero no a la inversa.

   b) Las velocidades típicas de conexión a Internet vía ADSL son de unos cuantos Mb/s.

   c) El servicio ADSL aprovecha las líneas de telefonía convencional para ofrecer conexión a Internet.

5. ¿Se podría grabar (guardar) en un CD-ROM toda la información contenida en un instante determinado en la memoria RAM de un ordenador viejo que tiene 512 MB?

   a) Sí.

   b) No, porque no cabe.

   c) No, porque un soporte es electrónico y el otro óptico.

6. ¿Cuál de estas afirmaciones es cierta?

   a) En cualquier dispositivo de almacenamiento (disco duro, CD-ROM, memoria USB) siempre hay un directorio principal o raíz.

   b) Un disco no puede contener más de dos niveles de jerarquía de carpetas.

   c) Una carpeta no puede contener sólo otra carpeta.

7. ¿Puede haber dos carpetas con el mismo nombre una dentro de la otra?

   a) No.

   b) Sí, si tienen fecha y hora diferentes.

   c) Sí.

8. ¿Cuántos valores posibles puede tomar un *byte*?

   a) 2

   b) 256

   c) 8

9. ¿Cuál de los tres medios de almacenamiento siguientes no es óptico?

   a) Un CD-ROM

   b) Un DVD

   c) Un disco fijo.

10. ¿Dónde reside un programa de ordenador mientras lo estamos ejecutante?

    a) En el disco duro.

    b) En la memoria RAM (al menos parcialmente).

    c) En el CD-ROM.

11. ¿Cuál de estas definiciones de fichero es más correcta?

a) Un conjunto de datos que se manipula como un todo, reside en algún medio de almacenamiento y tiene un nombre.

b) Una estructura que contiene los nombres de otros ficheros.

c) Una estructura de datos que representa el texto generado por un procesador de textos y que tiene un nombre asociado.

12. ¿Cuáles son las características de la memoria RAM de un ordenador de sobremesa?

a) es lenta, volátil y de acceso aleatorio.

b) es rápida, volátil y de acceso aleatorio.

c) es rápida, permanente y de acceso secuencial.

13. ¿Se puede hacer que varios ordenadores compartan un recurso conectado a uno de ellos como, por ejemplo, una impresora?

a) Sí, si los ordenadores están conectados formando una red local.

b) Sólo si la impresora está conectada a Internet.

c) Sí, instalándole un módem ADSL en la impresora.

14. Cada punto de una pantalla puede tener 256 colores: ¿cuántos *bytes* de memoria ocupa cada punto?

a) 1

b) 256

c) 8

15. Cuando el procesador central está ejecutando un programa, ¿dónde espera encontrar la siguiente instrucción?

a) En el CD-ROM.

b) En el disco duro.

c) En la memoria RAM.

16. ¿Es posible poner un fichero de texto en el directorio (carpeta) raíz?

a) Sí, como en cualquier directorio.

b) Sólo si es un fichero propio del sistema operativo.

c) No, primero se tiene que crear una carpeta (un directorio).

17. En la Universidad de Alicante hay alrededor de 35.000 alumnos. Si asignamos un número a cada alumno, ¿cuántos *bytes* hacen falta para guardar el número de cada alumno?

a) 15

b) 2

c) 3

18. Prácticamente todos los programas necesitan hacer operaciones básicas como por ejemplo abrir y cerrar archivos o gestionar el ratón y la pantalla. ¿Quiere decir esto que tanto un navegador como un procesador de textos como una memoria de traducción contienen instrucciones de programa para ejecutar estas operaciones básicas?

    a) No, sólo instrucciones para invocar los correspondientes programas del sistema operativo.

    b) Sí, porque forman parte del procesador central.

    c) Sí, porque, si no, no las podrían ejecutar.

19. ¿Dentro de una carpeta (directorio) podemos poner carpetas y documentos (ficheros) mezclados?

    a) No. Si una carpeta está dividida en subcarpetas, no puede contener documentos; los documentos tendrían que ir dentro de las subcarpetas

    b) Sólo en la carpeta raíz.

    c) Sí.

20. ¿Cuánta memoria ocupa una imagen de 1024×1024 puntos en la que cada punto puede tener 8 colores?

    a) 1 *megabyte*

    b) 384 *kilobytes*

    c) 8 *megabytes*

21. Algunos ordenadores portátiles están diseñados de forma que, cuando las baterías están a punto de agotarse (o el ordenador no se está usando), copian *toda* la memoria RAM en el disco duro y se apagan. Si volvemos a cargar las baterías y encendemos el ordenador, hacen la operación inversa. ¿Podemos esperar que la ejecución de los programas continúe en el mismo punto donde se encontraba cuando las baterías fallaron?

    a) No, porque la memoria RAM se borra cuando falta la alimentación eléctrica.

    b) No, porque sólo se ha guardado el sistema operativo.

    c) Sí, porque los programas en ejecución y sus datos estaban todos en la memoria RAM (si no estaban ya en el disco).

22. ¿Puede un fichero contener las instrucciones de un programa ejecutable?

    a) No.

    b) Sólo si está escrito en un lenguaje de programación de alto nivel, porque sólo así será un texto y podrá guardarse en un fichero.

    c) Sí.

23. Si las imágenes enviadas por una vieja cámara digital sin colores (blanco y negro) tienen 100×100 píxeles, ¿cuántas de estas imágenes podríamos almacenar en una memoria USB de 1 GB?

    a) Dependiendo de la codificación escogida para los caracteres de la imagen, entre 400 000 y 800 000.

    b) Unas 10 000.

    c) Unas 800 000.

24. ¿Qué característica es común a todos los tipos de software?

    a) Que empiezan a ejecutarse al conectar el ordenador.

    b) Que consisten en una lista de instrucciones ejecutables.

    c) Que se encargan de la gestión de todos los recursos del hardware del ordenador donde se ejecutan.

25. ¿Es posible que un fichero de texto y la carpeta en que está incluido tengan el mismo nombre?

    a) Sólo si el fichero ha sido creado por el sistema operativo.

    b) Sólo si se trata del directorio raíz.

    c) Sí, no importa el nombre de la carpeta.

26. ¿Cuántos bits necesitamos para codificar un número de teléfono de 9 cifras suponiendo que codificamos los dígitos uno a uno?

    a) 27

    b) 36

    c) 9

27. La tarjeta Compact Flash donde se almacenan las fotografías de una cámara digital tiene 2 GB. Si suponemos que hemos elegido una resolución y un formato de imagen que hace que cada fotografía necesite un espacio de 2.048 kB, ¿cuántas tarjetas de estas tenemos que comprar si queremos hacer 2.500 fotos a lo largo de un viaje?

    a) 1

b) 3

c) 4

28. El *sistema operativo* de un ordenador es...

    a) ... *hardware*.

    b) ... *software*.

    c) ... una manera de especificar el formato de los textos.

29. Si reducimos de 3.000 MHz a 1.500 MHz la frecuencia del reloj de un ordenador y todavía funciona...

    a) ... ejecutará los programas a la misma velocidad.

    b) ... ejecutará los programas más lentamente.

    c) ... tardará menos en ejecutar los programas.

30. Windows usa las *extensiones* de los nombres de ficheros para...

    a) ... asociarlos al programa que los abrirá cuando hagamos doble clic sobre el icono del fichero.

    b) ... ahorrar espacio cuando se guardan los ficheros.

    c) ... saber si están vacíos o contienen texto.

31. Un módem es un dispositivo que...

    a) ... convierte la información digital en señales analógicas.

    b) ... convierte señales analógicas en información digital.

    c) ... hace las dos cosas.

32. Indicad cuál de las afirmaciones siguientes es cierta:

    a) La memoria RAM almacena programas y datos mientras se ejecutan los programas.

    b) La memoria RAM es permanente y más rápida que la memoria secundaria.

    c) Las otras dos afirmaciones son falsas.

33. Los programas de aplicación ...

    a) ...siempre interactúan directamente con el hardware del ordenador.

    b) ...los pone en ejecución el sistema operativo.

    c) ...no pueden comunicarse con otras aplicaciones.

34. Indicad cuál de las afirmaciones siguientes es falsa:

a) Un *gigabyte* equivale a 1.024 *kilobytes*.

b) Un *byte* equivale a *8 bits*.

c) Un *kilobyte* equivale a 1024 *bytes*.

35. La información se almacena en la memoria del ordenador en forma binaria. ¿Qué quiere decir esto?

a) Que cada dato es una secuencia de *bytes* y cada byte puede adoptar 512 valores.

b) Que cada dato es una secuencia de números codificados en ASCII.

c) Que cada dato es una secuencia de *bits*, cada uno de los cuales sólo puede adoptar dos valores.

36. ¿Cuántos *bits* hacen falta para representar los 12 meses del año?

a) 4 *bits* y sobran 4 combinaciones.

b) 12 *bits*, uno por cada mes del año.

c) 6 *bits*, uno por cada dos meses.

37. Indicad cuál de las afirmaciones siguientes es falsa. Los programas de aplicación ...

a) ...suelen estar escritos para ser ejecutados sobre un sistema operativo concreto.

b) ...acceden a los recursos y dispositivos conectados al ordenador a través del sistema operativo.

c) ...nunca necesitan del sistema operativo una vez que han empezado a ejecutarse.

2.9. Soluciones

1. (c): Un gigabyte tiene 1.024 megabytes, y un megabyte, 1.024 kilobytes: $1.024 \times 1.024 = 1.048.576$.

2. (c): Una memoria USB de 6 GB (gigabytes) contiene aproximadamente 6.000.000.000 bytes. Un carácter, en las codificaciones usadas comúnmente en Europa occidental, ocupa un byte; por lo tanto, la página de 50×60 ocupa 3.000 bytes. Caben $6.000.000.000/3.000 = 2.000.000$ páginas en un disco.

3. (a): Una velocidad de 300 kilobytes por segundo equivale a unos $300 \times 8 = 2.400$ kilobits por segundo; alrededor 2,3 Mb/s.

4. (a): Los módems modulan (convierten señales digitales en analógicas) y desmodulan (convierten señales analógicas en digitales) para enviar y recibir datos a través de un determinado medio. Las líneas ADSL domésticas actuales admiten conexiones vía módem telefónico de unos cuántos megabits por segundo (véase la sección 2.7).

5. (a): Un CD-ROM puede almacenar como mínimo 650 MB.

6. (a)

7. (c)

8. (b): $2^8 = 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 256$.

9. (c): Los discos fijos son generalmente magnéticos.

10. (b): Al menos la porción del programa que se está ejecutando tiene que residir en la RAM.

11. (a)

12. (b)

13. (a)

14. (a): Cada punto puede tomar uno de 256 colores. Para poder almacenar el color hace falta un número de bits suficiente para hacer 256 combinaciones. Con 8 bits podemos hacer $2^8 = 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 256$ combinaciones. Por lo tanto, se necesitan 8 bits, es decir, un byte.

15. (c)

16. (a)

17. (b): El número de bits necesario para poder generar 35.000 combinaciones es el número de veces que hay que multiplicar $2 \times 2 \times \ldots$ justo hasta el punto en que se supera 35.000. Hay que hacerlo 16 veces; por lo tanto, necesitamos 16 bits. Como cada byte son 8 bits, son necesarios 2 bytes.

18. (a)

19. (c)

20. (b): Cada punto puede presentarse en 8 colores diferentes. Con 3 bits podemos almacenar $2 \times 2 \times 2 = 8$ colores. Por lo tanto, la imagen ocupa $1.024 \times 1024 \times 3 = 3.145.728$ bits, que son $3.145.728/8 = 393.216$ bytes, que son $393.216/1024 = 384$ kilobytes.

21. (c)

22. (c)

23. (c): Cada imagen ocupa $100 \times 100 = 10.000$ bits, que son $10.000/8 = 1.250$ bytes. 1 GB = $1.024 \times 1.024 \times 1.204 = 1.073.741.824$ bytes. $1.073.741.824/1.250 = 858.993$. Podemos almacenar más de 800.000 imágenes.

24. (b)

25. (c)

26. (b): Cada dígito decimal puede tomar, por separado, 10 valores diferentes (del cero al nueve). Tres bits por dígito decimal no son suficientes (permiten sólo 8 combinaciones); cuatro, sí. Así, cada dígito decimal ocupa 4 bits; si hay 9, necesitamos 36 bits.

27. (b): 2GB son $2 \times 1.024 \times 1.024 = 2.097.152$ kB. En la tarjeta Compact Flash caben $2.097.152/2.048 = 1.024$ imágenes. Con 3 tarjetas puedo almacenar $3 * 1024 = 3.072$ imágenes.

28. (b)

29. (b)

30. (a)

31. (c)

32. (a)

33. (b)

34. (a)

35. (c)

36. (a): Tenemos 12 valores diferentes (de enero a diciembre). Tres bits por mes no son suficientes (permiten sólo 8 combinaciones); cuatro sí (permiten 16 combinaciones).

37. (c)

Capítulo 3

Internet

Una de las herramientas informáticas básicas que se encuentran al alcance de las personas que se dedican profesionalmente a la traducción es Internet. Internet permite básicamente tres tipos de uso:

Como medio de comunicación: Internet permite la comunicación y el intercambio de archivos (véase el apartado 2.4) con clientes o proveedores, la participación en foros de profesionales, la realización de consultas, etc.

Como fuente de documentación: Además de contener textos de muchas clases que pueden servir de ejemplo o inspiración a la hora de hacer traducciones, se pueden encontrar enciclopedias, diccionarios, glosarios, memorias de traducción (véase el capítulo 10), y otras muchas fuentes de documentación.

Como repositorio de software de asistencia a la traducción: Muchos de los programas específicos de asistencia a la traducción están disponibles en Internet, como por ejemplo los sistemas de traducción automática (véase el capítulo 8) en línea o los programas de concordancias bilingües.[1] El acceso puede ser a través de un navegador, o a través otros programas que tengamos instalados localmente en nuestro ordenador.[2]

3.1. ¿Qué es Internet?

Se denomina *Internet* a un conjunto de ordenadores, distribuidos en todo el mundo e interconectados mediante un protocolo estándar (el *proto-*

[1] Programas de concordancias bilingües disponibles en Internet: Linguee (`http://www.linguee.es/`); Reverso Contexto (`http://context.reverso.net/`).

[2] Usando protocolos bien especificados, normalmente a través de API, *Application Program Interfaces* o *interfaces de programación de aplicaciones*.

colo de Internet o IP) de forma que los recursos presentes en unos ordenadores (normalmente, información) están disponibles para ser usados por los usuarios de otros ordenadores. Se dice que los ordenadores de Internet forman una *red*, en la cual los nodos o nudos son los ordenadores y los hilos, las conexiones. Las conexiones pueden ser de naturaleza muy diversa (líneas telefónicas, fibra óptica, enlaces de radio terrestres o por satélite, etc.), pero el protocolo de Internet está diseñado de forma que la naturaleza de la conexión no sea relevante para el usuario ni para los programas de aplicación que hacen uso de estas conexiones. Otros nombres que se usan en vez de *Internet* son *World Wide Web* o *WWW* ("telaraña de alcance mundial") o simplemente *web* ("telaraña"), femenino en español (*la web*).

3.2. Números IP

Cada nodo (cada ordenador) de la red Internet tiene un *número IP* único, el cual se compone de 4 bytes (4 enteros del 0 al 255) separados por puntos, como por ejemplo `192.168.5.5`. Los enteros iniciales se usan para designar grandes subredes, mientras que los finales se usan para designar redes más pequeñas, y dentro de éstas, ordenadores concretos (en esto recuerdan a los números de teléfono: dos abonados próximos normalmente comparten las cifras iniciales).

3.3. Nombres

Como recordar números IP no es fácil, normalmente se usan *nombres* o *direcciones* para referirse a las máquinas; algunos de los ordenadores de la red (llamados *servidores de nombres*) se encargan de traducir los nombres a números IP. Por ejemplo, un nombre podría ser `altea.dlsi.ua.es`, donde `altea` se refiere a una máquina concreta del Departamento de Lenguajes y Sistemas Informáticos (`dlsi`) de la Universitat d'Alacant (`ua`), que se encuentra en España (`es`); este orden es el inverso al de los números IP (en esto los nombres se asemejan a las direcciones postales: primero se da el más concreto y al final el país).

La tabla 3.1 da algunos ejemplos de indicativos de países. A veces, el último componente de un nombre no se corresponde con el indicativo de un país, sino que indica la naturaleza del lugar; antiguamente había que sobreentender que se trataba de un ordenador situado físicamente en los Estados Unidos de América, pero esto ya no es necesariamente así. Estos indicativos aparecen en la tabla 3.2. En otros países (`.uk`, `.nz`, `.za`) se usan indicativos similares (`.co`(mercial), `.ac`(académico), etc.) antes del indicativo de país (por ejemplo, `www.shef.ac.uk` es la Universidad de Sheffield).

INDICATIVO	PAÍS
.es	España
.fr	Francia
.pt	Portugal
.it	Italia
.uk	Reino Unido
.ru	Rusia
.za	Suráfrica
.ie	Irlanda
.tv	Tuvalu
.to	Tonga
.nu	Niue
.fm	Estados Federados de Micronesia

Tabla 3.1: Indicativos de Internet de algunos países. Fijaos que algunos indicativos (`.tv`, `.fm`, etc.) se usan para aplicaciones no estrictamente relacionadas con estos países.

INDICATIVO	TIPO
.gob	gubernamental
.mil	militar
.com	comercial
.org	organización sin ánimo de lucro
.edu	institución educativa
.info	webs informativas
.cat	cultura y lengua catalanas (patrocinado por la fundación puntCat)
.eus	cultura y lengua vascas (patrocinado por la fundación PuntuEus)
.museum	museos (patrocinado por MuseDoma)

Tabla 3.2: Algunos indicativos de Internet usados originalmente en los Estados Unidos de América y más recientemente en todo el mundo, algunos de ellos patrocinados por determinadas instituciones.

Para saber más sobre servidores de nombres

Podríamos hacer una analogía entre la relación entre los nombres y los números IP de los ordenadores de Internet y los nombres y los números de teléfono de la agenda de nuestro móvil. Cuando telefoneamos a una persona, normalmente lo hacemos buscando su nombre en la agenda, y pocas veces lo hacemos por el número, pero para hacer la llamada es necesario el número. Cuando accedemos a ordenadores de Internet, lo hacemos de manera similar: accedemos por el nombre y no por el número IP, el cual es indispensable para hacer la conexión. Pero, en contraste con la agenda de nuestro móvil, es impracticable tener todos los nombres y los números IP correspondientes a todos los ordenadores del mundo en nuestro ordenador. Por eso se usan *servidores de nombres*: ordenadores a los cuales nuestro ordenador se conecta por el número IP y a los cuales puede preguntar por el número IP correspondiendo a un nombre. Los *servidores de nombres* se organizan de forma que se distribuyen la información de manera jerárquica haciendo uso del *sistema de nombres de dominio* (en inglés, *domain name system* o DNS).

Por ejemplo, cuando queremos conectarnos a `cercador.dlsi.ua.es`, el servidor de nombres de nuestro proveedor de acceso a Internet ve que el nombre acaba en `.es` y pregunta al servidor de nombres que se encarga de este *dominio*; este servidor ve que el elemento anterior es `.ua` y pregunta al servidor de nombres de la Universitat d'Alacant (UA), y éste, a su vez, pregunta al servidor de nombres del Departamento de Lenguajes y Sistemas Informáticos, puesto que el elemento anterior es `.dlsi`. Este último, finalmente, entrega el número IP del ordenador llamado `cercador` al servidor de la UA, y éste al servidor del dominio geográfico `.es`, que lo entrega a nuestro proveedor de servicios de Internet y éste, a su vez, a nuestro ordenador para hacer la conexión. Por eso, cuando navegamos, la primera conexión tarda más: se está *resolviendo* el nombre que hemos tecleado. Una vez resuelto, nuestro ordenador se guarda la IP durando un tiempo para evitar preguntar de nuevo. Además, para reducir el tráfico en Internet los proveedores de servicios de Internet y los servidores de nombres consultados también se guardan temporalmente esta información de forma que no siempre se desencadena el proceso completo de consultas descrito.

3.4. Identificadores de recursos

Los servicios y los documentos concretos presentes en un ordenador (un servidor de Internet) que los hace disponibles se pueden designar mediante su *identificador uniforme de recursos* o, más comúnmente, *URI* (del inglés *uniform resource identifier*).[3] El URI es, por lo tanto, una expresión que identifica o localiza uniformemente un servicio o documento (un recurso) de cualquiera de los que se ofrecen en Internet.

Un URI tiene generalmente tres partes, aunque se dan algunas variaciones y una de ellas no es obligatoria:

esquema: indica la clase de recurso y cómo lo tiene que usar el ordenador solicitante (o *cliente*).

[3] La denominación más usual era URL, *uniform resource locator* o localizador uniforme de recursos, que todavía se usa profusamente, aunque no todos los URI son URL.

autoridad: identifica por su nombre o número IP el ordenador (*servidor*) donde está el recurso.

trayectoria: (opcional) da información sobre la localización del servicio o documento dentro del ordenador servidor (muchas veces similar a las *trayectorias* de los ficheros, p. 12).

Por ejemplo, el URI

```
http://www.canalcuina.tv/concurs/sms/index.html
```

se refiere a un documento de *hipertexto* —un documento de texto que contiene enlaces que permiten acceder directamente a otros hipertextos relacionados— compatible con el esquema `http` (*hypertext transfer protocol* o protocolo de transferencia de hipertextos) situado en el ordenador `www.canalcuina.tv` (de la empresa ficticia Canal Cocina, posiblemente perteneciente al mundo de la televisión[4].), y, dentro de este, en el directorio `concurso`, subdirectorio `sms`. El fichero que contiene el hipertexto se denomina `index.html`, donde las siglas HTML corresponden a *hypertext markup language*, nombre del lenguaje o sistema de marcas más usado para dar formato a los hipertextos (véase el apartado 4.1).

El esquema `https://` es similar al esquema `http://` pero incorpora, además, mecanismos para transmitir con seguridad información encriptada (cifrada). Muchos de los servidores de Internet encargados de manipular información privada usan este esquema.

Los URI no sólo sirven para enlazar hipertextos: el URI `mailto:anton@dlsi.ua.es` sirve para enviar correo electrónico (`mailto`) al usuario que tiene la dirección de correo electrónico `anton@dlsi.ua.es`. Otros esquemas son `rtsp://`, *real-time streaming protocolo*, para enlazar contenido como por ejemplo vídeo, audio, etc. en tiempo real, o `ftp://`, *file transfer protocol*, usado, cada vez menos, para descargar (transferir) ficheros para guardarlos en nuestro ordenador.

3.5. Navegadores

Los programas navegadores se conocen también por otros nombres: *browsers*, *exploradores*, etc. (véase también la pág. 8). Son programas que permiten acceder de manera sencilla a los documentos o servicios de Internet en ordenadores conectados a esta red; entre otras cosas, los navegadores interpretan los hipertextos escritos en HTML y los presentan a la persona usuaria en el formato que indican las marcas, de forma que los enlaces a otros hipertextos queden claramente destacados y sean *activos*, es decir,

[4] Aunque, como se muestra en la tabla 3.1, el indicativo designa un estado del Pacífico denominado Tuvalu.

que respondan a un clic del ratón *saltando* al hipertexto o recurso enlazado; además, los navegadores pueden *ejecutar* automáticamente otros programas de aplicación para poder abrir el recurso correspondiente si no es un hipertexto.

Los navegadores más usados son *Firefox* (un programa libre y de código fuente abierto desarrollado por centenares de colaboradores en todo el mundo), *Chrome* (el navegador de la compañía Google, el cual tiene una versión libre y de código fuente abierto denominada *Chromium*), *Microsoft Internet Explorer* (incorporado en el sistema operativo Windows), *Safari* (el cual forma parte del sistema operativo MacOS), y otros como *Opera*, etc.

3.6. Buscadores

Uno de los recursos de Internet más útiles son los *buscadores*. Se trata de páginas *web* que permiten buscar documentos de Internet; se tiene que teclear una o más palabras, además de otras *condiciones de búsqueda* opcionales, como por ejemplo que los documentos estén en una lengua determinada o en un servidor determinado, y entregan los URIs de los documentos que cumplen estas condiciones, enlaces a estos documentos y un pequeño resumen o recorte (denominado *snippet*) del contenido de las páginas deseadas.[5]

Ejemplos de búsquedas:

- `megamondrío`: documentos que contengan la palabra *megamondrío* y quizás también formas del mismo como por ejemplo el plural *megamondríos*, el compuesto *mega-mondrío*, o la forma sin acento *megamondrio*.
- `megamondrío síngulo`: documentos que contengan estas dos palabras o variantes.
- `megamondrío síngulo site:ua.es`: documentos que contengan estas dos palabras o variantes y que estén en documentos cuyo URI acabe en *ua.es*.
- `megamondrío síngulo filetype:pdf`: documentos PDF que contengan estas dos palabras.

Tiene que quedar claro que los buscadores realmente *no buscan* documentos en Internet sino que consultan *índices* que han ido construyendo a

[5] Algunos buscadores, como por ejemplo *Google*, modifican los enlaces que llevan a los resultados, de forma que no llevan directamente sino que pasan primero por el servidor del buscador, para conocer las preferencias de la gente y mejorar así la relevancia de los resultados, o incluso para establecer un perfil de cada persona usuaria. Esto hace que algunas personas se planteen el uso de buscadores que no hagan este *seguimiento* o *tracking*.

partir de los documentos visitados. Por lo tanto, puede haber documentos que los buscadores no encuentren porque nunca los han visitado. Por la misma razón, también puede pasar que los buscadores entregan resultados correspondientes a páginas que ya no existen.

Uno de los buscadores más populares a la hora de escribir estas líneas es *Google* (`http://www.google.com`); pero también hay otros como *Duckduckgo!*, *StartPage*, etc. La mayor parte de estos buscadores tienen interfaces de uso en muchas lenguas.

3.7. Correo electrónico

Uno de los servicios más usados de Internet es el correo electrónico (en inglés *electronic mail* o *e-mail*), que nos permite enviar mensajes (textos informatizados) a usuarios de otros ordenadores. Los mensajes pueden contener, además del texto mismo del mensaje, ficheros *anexos* (o *adjuntos*, en inglés *attachments*) como por ejemplo imágenes, documentos, mensajes reenviados, etc.

Las direcciones de correo electrónico tienen dos partes, separadas por el carácter "`@`", que se suele pronunciar *at* (en inglés, por parte de los informáticos más viejos) o *arroba*. La primera parte (o *parte local*) es frecuentemente el identificador de una persona y la segunda (la *parte de dominio*) suele ser el nombre de un ordenador (o de un grupo de ordenadores que comparten un mismo nombre). Por ejemplo, una dirección electrónica válida podría ser

`marty.mcfly@backtothefuture.info`

A veces, podemos usar nuestra dirección de correo electrónico para identificarnos a la hora de acceder a algunos de los servicios que se ofrecen en Internet, como por ejemplo los servicios de redes sociales (véase el apartado 3.9).

Una dirección electrónica puede también identificar una lista de usuarios (*alias*) o una lista de distribución (la cual envía una copia de cada mensaje que recibe a todos los inscritos en la lista). En ambos casos, si mandamos un mensaje, lo reciben todos los inscritos, de forma que se puede usar para establecer, por ejemplo, foros de discusión.[6]

Para leer, escribir o enviar los mensajes de correo electrónico, se usan *programas gestores de correo electrónico*, como por ejemplo Thunderbird, Outlook,

[6]Por ejemplo, la lista de distribución sobre traducción automática MT-List, mantenida por la EAMT (*European Association for Machine Translation*, asociación europea para la traducción automática), tiene la dirección `mt-list@eamt.org`; para formar parte de la lista hay que subscribirse en la URI `http://lists.eamt.org/mailman/listinfo/mt-list`. Si se manda un mensaje a `mt-list@eamt.org` lo reciben todos los subscritos. Otra lista de interés, Tradumàtica, sobre tecnologías de la traducción, permite suscripción a través de `https://listserv.rediris.es/cgi-bin/wa?A0=TRADUMATICA`.

etc. También es muy frecuente acceder al correo electrónico desde cualquier lugar usando un navegador, a través de un servicio llamado *webmail*; son comunes los *webmails* gratuitos (GMail, Yahoo Mail, Microsoft Outlook); cada alumna o alumno de la Universitat d'Alacant tiene, por serlo, una dirección de correo de la forma *xxxx*`@alu.ua.es`, accesible a través de la sección *webmail* de la página web de la Universidad.

3.8. Mensajería instantánea y chat

La mensajería instantánea y el chat (en inglés *chat*) permiten —como en el caso del correo electrónico, a través de programas especializados o *webs* accesibles con un navegador— una comunicación escrita muy rápida ("en tiempo real"), consistente en mensajes normalmente cortos —opcionalmente con anexos como por ejemplo fotografías, contactos—, de forma que el resultado es similar al de una conversación, pero por escrito, [7] cosa que permite un registro de comunicación muy informal que, de hecho, ha dado lugar a una lengua muy diferenciada tanto de la oral como de la escrita.

Con la generalización del uso de los móviles inteligentes o *smartphones*, han aparecido muchas aplicaciones de este tipo, como por ejemplo Telegram, Whatsapp, Line, etc., que usan como identificador el número de teléfono.

Las conversaciones pueden ser entre dos personas, o entre un *grupo* de personas, a veces reunidos en una *sala*, con intereses comunes. Las personas que participan en un *chat* de estos últimos, pueden a veces elegir un alias o mote y "entrar" a la sala, o estar siempre conectadas al grupo, y "conversar" por escrito públicamente. Desde el grupo o la sala se pueden establecer "conversaciones aparte" (en privado) cuando hace falta con alguna persona concreta.

Entre los servicios de mensajería instantánea comercial más populares están los asociados a redes sociales como por ejemplo Facebook o Tuenti, u otros asociados a otras aplicaciones como por ejemplo Google Hangouts o Skype.

3.9. Servicios de red social

En la actualidad, una de las aplicaciones más frecuentes de Internet son los *servicios de red social*, comúnmente denominados simplemente *redes sociales*. Son plataformas informáticas que usan Internet (a través de un navegador y frecuentemente también a través de programas de aplicación

[7] Aunque se está popularizando el uso de *notas de voz*, archivos de audio grabados y que se envían como anexos.

específicos, muy populares para teléfonos móviles) para construir redes de personas que comparten intereses u objetivos. Algunos ejemplos:

Facebook permite a cada persona publicar informaciones sobre su *estado*, incluyendo fotos, y enviarse mensajes; el estado puede ser visible para todo el mundo o sólo para personas *amigas*.

Google+ se puede ver como la respuesta de Google a Facebook; en Google+ el concepto básico es el del *círculo*.

Twitter es una red social que se basa en mensajes de menos de 140 caracteres que pueden llevar adjuntos fotos, enlaces, etc.

Instagram hace énfasis en la posibilidad de compartir fotografías y vídeos.

LinkedIn sirve para construir redes relacionadas con la actividad profesional.

Hay otros muchos de alcance mundial (Pinterest, Reddit, Tumblr, etc.) y algunos particulares de determinadas áreas geográficas (como por ejemplo VK en los países donde se habla ruso).

3.10. El acceso en Internet

3.10.1. Acceso doméstico

Para acceder a Internet desde casa, hace falta, por un lado, darse de alta con algún proveedor de servicios de Internet (ISP, *Internet service provider*) —algunos proveedores ofrecen, además del acceso a Internet, televisión digital y telefonía convencional— y de otro, tener un *módem* adecuado al tipo de conexión (módem ADSL, módem de cable, etc.). Normalmente los módems que nos venden los proveedores de servicios de Internet son al mismo tiempo módem y encaminador (en inglés *router*) para permitir que conectemos más de un dispositivo, normalmente a través de una conexión inalámbrica Wi-Fi, formando una red local (en inglés *local area network*, LAN). La figura 3.1 muestra un esquema del acceso doméstico a Internet desde varios dispositivos.

Nuestro proveedor de servicios de Internet asigna un número IP público a nuestro encaminador (*router*), de forma que forme parte de Internet y pueda facilitar el acceso a todos los dispositivos de nuestra red local (clientes) a todos los servicios y documentos disponibles en cualquier máquina (servidora) de Internet. A los dispositivos de nuestra red local el encaminador les asigna un número IP privado al que sólo tienen acceso los ordenadores que forman parte de esta red local; todos los dispositivos de esta red local se conectan a Internet usando el mismo número IP, el número IP público asignado a nuestro encaminador.

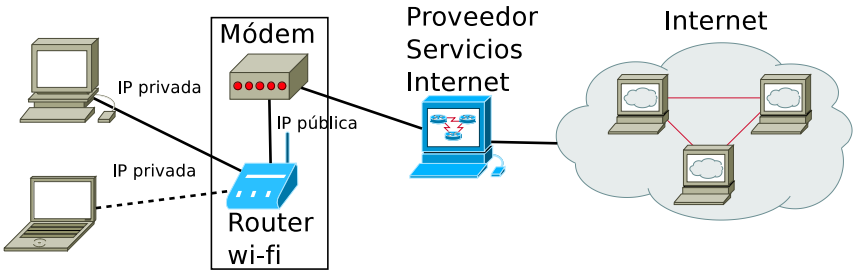


Figura 3.1: Esquema de acceso doméstico a Internet desde varios dispositivos conectados a un módem-encaminador.

En la mayoría de los casos, el número IP público que nuestro proveedor de acceso a Internet asigna a nuestro encaminador es temporal y va cambiando, por eso los ordenadores que tenemos en casa no pueden actuar como servidores.

En la actualidad, en los domicilios del País Valencià hay básicamente tres modalidades de acceso a Internet, todas por precios que oscilan alrededor de los 30–40 euros/mes:

ADSL: (véase el glosario de la sección 2.7) el módem hace la conexión a través de los hilos de telefonía convencional ya instalados en las casas; en la actualidad se consiguen velocidades de bajada de algunos Mb/s y de subida normalmente inferiores a 1 Mb/s.

Cable: el módem hace la conexión a través de un cable coaxial, tecnología que se usaba tradicionalmente para televisión, con velocidades y precios similares al ADSL.

Fibra: el módem hace la conexión a través de un cable de fibra óptica (láser); esta tecnología es la más reciente y permite velocidades de bajada de decenas o incluso centenares de Mb/s y velocidades de subida superiores al Mb/s.

Para saber más sobre los antiguos módems telefónicos

Hasta los primeros años del decenio del 2000, la mayor parte de los domicilios particulares y las pequeñas empresas se conectaban a Internet mediante la línea telefónica, pero usando una tecnología mucho más rudimentaria que requería hacer una llamada telefónica al número del proveedor de acceso a Internet, llamada que tenía que durar el tiempo de conexión, independientemente de la cantidad de datos que se transfirieran. La llamada se podía pagar por minutos o con planes que permitían conexiones ilimitadas en horario no comercial que se denominaban *tarifa plana*. El módem modu-

laba y desmodulaba señales similares a las que se envían cuando se hace una llamada de voz. Durante la conexión, la línea quedaba ocupada y no se podían hacer ni recibir llamadas. Las velocidades eran muy bajas, de decenas de kb/s.

3.10.2. Acceso móvil

Cada vez nos conectamos menos a Internet desde casa, y más desde nuestros dispositivos móviles como por ejemplo los *smartphones*. Si estamos en el radio de alcance de una red Wi-Fi a la cual tenemos acceso (como la de nuestra casa o la que nos ofrece la Universidad), nuestro dispositivo móvil se conectará en Internet a través de Wi-Fi. Si no, tendremos que usar los *datos móviles* que comercializa nuestro proveedor de telefonía móvil a través de su red celular. En la actualidad se está haciendo la transición de la tecnología llamada de tercera generación o 3G (que se muestra a veces también como una "H" en la pantalla), que permite velocidades de conexión de unos pocos Mb/s a la de cuarta generación o 4G, que permite velocidades mucho más rápidas (a veces, cuando la cobertura no es buena, nuestro móvil recurre a tecnologías más antiguas y más lentas, como por ejemplo EDGE[8] —que permite centenares de kb/s y se muestra como una "E" en la pantalla— o GPRS[9] —que permite decenas de kb/s y se muestra como una "G" en la pantalla). En la actualidad, podemos comprar paquetes de datos y pagar unos 5–7 euros por GB, o una cuota mensual de decenas de euros y tener datos ilimitados.

3.11. Cuestiones y ejercicios

1. La primera parte de un URI (identificador uniforme de recursos) especifica …

   a) …el esquema de acceso.

   b) …el nombre del servidor.

   c) …el directorio donde se encuentra el servicio.

2. Después del esquema de acceso, un URI (identificador uniforme de recursos) especifica

   a) la velocidad de transferencia.

   b) el nombre del servidor.

   c) el directorio donde se encuentra el servicio.

[8] *Enhanced Data Rates for GSM Evolution* "Tasas de datos mejoradas para la evolución del GSM"

[9] *General Packet Radio Service* "Servicio general de paquetes [de datos] por radio"

3. ¿Qué es "`http://www.tharaka.org.ke/nkoru`"?

   a) Un URI.

   b) Una dirección de correo electrónico.

   c) El nombre de un fichero local de nuestro ordenador.

4. Los números IP se componen de 4 números del 0 al 255 separados por puntos. ¿Cuántos bits son necesarios para almacenar un número IP?

   a) 16

   b) 32

   c) 4

5. Cuándo en un navegador no se indica el esquema de un URI, ¿qué esquema se sobreentiende?

   a) `http://`

   b) `mailto:`

   c) El esquema de Internet

6. ¿Qué se puede decir de los números IP de dos máquinas que se encuentran en la misma subred?

   a) No se puede decir nada: los números IP pueden no tener nada que ver.

   b) Que tienen en común los primeros bytes.

   c) Que tienen en común los últimos bytes.

7. Sakurako se conecta a Internet por vía telefónica con un ordenador que tiene un procesador de 1000 MHz, 128 megabytes de RAM y un módem de 57600 bits por segundo. Ahmed tiene un ordenador con un procesador de 500 MHz y 128 megabytes de RAM y ha contratado un módem de cable de 128 kilobits por segundo para conectarse a Internet. Sakurako quiere convencer a Ahmed de que ella se descarga los ficheros MP3 más rápido que él, pero Ahmed le dice que en iguales condiciones él tarda menos en bajarse los ficheros, a veces la mitad de tiempo. ¿Quién tiene razón?

   a) Ahmed

   b) Los dos se bajan los archivos en el mismo tiempo porque los dos ordenadores tienen la misma RAM.

   c) Sakurako

8. Si la máquina `fictici.deconya.ua.es` tiene el número IP 232.111.22.33, ¿cuál de los tres número IP siguientes es más probable que corresponda a la máquina `fals.deconya.ua.es`?

   a) 230.111.22.33

   b) 232.111.22.13

   c) 67.15.22.99

9. ¿Cómo se indica en Internet dónde está un recurso concreto?

   a) Mediante un URI.

   b) Mediante un hiperenlace.

   c) Mediante una etiqueta HTML.

10. ¿Cuál de los siguientes es un número IP válido?

    a) 64.128.64

    b) 255.256.111.1

    c) 111.255.111.111

11. En un número IP, ¿qué parte es igual para dos ordenadores conectados a la misma red local?

    a) La parte inicial.

    b) La parte final.

    c) Puede no coincidir nada porque los números IP se asignan aleatoriamente.

12. En nuestra casa hemos tenido teléfono toda la vida y ahora estamos pensando en conectarnos a Internet mediante ADSL. ¿Tenemos que hacer alguna instalación adicional en casa?

    a) No, pero nos quedaremos sin teléfono y sólo tendremos Internet.

    b) Sí, seguro que los técnicos vendrán a pasar cables por todas partes.

    c) No. Y disfrutaremos de teléfono e Internet a la vez.

13. Cuando realizamos una búsqueda en Internet, el buscador ...

    a) ...visita en ese momento los diferentes documentos de Internet y recopila aquellos que satisfacen el criterio de búsqueda.

    b) ...consulta un índice que ha construido previamente al visitar los diferentes documentos de Internet.

c) ...consulta un índice que ha construido previamente y también visita en ese momento los documentos de Internet por si hubiera alguno nuevo que no existía cuando creó el índice.

14. Si al buscar un recurso en Internet el buscador nos dice que no hay documentos que satisfagan el criterio de búsqueda ...

    a) ...podemos estar seguros de que no existe en todo Internet ningún documento que lo satisfaga.

    b) ...podría darse el caso de que un documento de reciente creación satisfaga el criterio de búsqueda pero no haya sido visitado por el buscador.

    c) Las otras dos respuestas son erróneas.

15. Dado el URI `http://edu.gob.es/educacion/universidades.html`, indica cuál de las siguientes afirmaciones es falsa:

    a) El recurso `universidades.html` está alojado en un ordenador cuyo nombre es `edu.gob.es`.

    b) La ruta de acceso al recurso es `educacion/universidades.html`.

    c) El ordenador donde se aloja el recurso se llama `http://edu.gob.es`.

3.12. Soluciones

1. (a)
2. (b)
3. (a)
4. (b): Cada número del 0 al 255 se puede almacenar en 8 bits ($2^8 = 256$) y hay cuatro: $8 \times 4 = 32$.
5. (a)
6. (b)
7. (a): 57.600 b/s son 57.600 / 1024 = 56,25 kb/s.
8. (b)
9. (a)
10. (c)
11. (a)
12. (c)

13. (b)
14. (b)
15. (c)

Capítulo 4

Textos y formatos

El tipo de fichero básico con el que trabajan los profesionales de la traducción suele ser un fichero con texto, es decir, un texto informatizado, también llamado *documento de texto*. Este fichero puede contener, además del texto mismo, información sobre la presentación (el formato de los párrafos y de las páginas, los tipos y los tamaños de letra que se usan con cada palabra, etc.) o sobre la organización del contenido (indicaciones de que una determinada parte del texto es el título de un capítulo, el título de una sección o una nota a pie de página, etc.).

Un texto informatizado puede tener diversos orígenes:

- Puede haber sido generado por otro programa de ordenador, por ejemplo a partir de los datos contenidos en alguna base de datos (véase el capítulo 5).
- Lo podemos haber recibido anexo a un mensaje de correo electrónico (véase el apartado 3.7) o por mensajería instantánea (véase el apartado 3.8).
- Lo podemos haber descargado (copiado) de algún servidor de Internet (veáse la pág. 33).
- Lo podemos haber generado, quizás a partir de otro texto, usando un *procesador de textos* (veáse el apartado 4.6).
- Lo puede haber generado un *sistema de reconocimiento del habla* a partir de la voz de la persona que lo ha dictado.
- Lo puede haber generado un *sistema de reconocimiento de textos escritos* a partir de un texto tipografiado o manuscrito.

4.1. Formatos de texto

Un *texto informatizado* es, como cualquier porción de datos informatizada, una *secuencia de bits*, es decir, de *unos* y *ceros*, del estilo de la siguiente:

```
0100000101001101001001...
```

Como ya hemos visto en el apartado 2.3, los *bits* se agrupan en grupos de ocho (*bytes*):

```
01000001 01001101 01001001...
```

Hay muchas maneras de organizar estos bytes para almacenar los textos; muchos de los problemas que aparecen cuando se tratan textos con el ordenador provienen de discrepancias en cuanto a la manera de hacerlo. En las siguientes secciones estudiaremos dos aspectos importantes que denominaremos *codificación* y *formato* propiamente dicho. La *codificación* es la asignación de una secuencia concreta, de uno o más *bytes*, a cada posible carácter de un texto. El *formato* de un documento es la parte no textual de este y sirve para codificar información estructural (sobre la organización del contenido del documento) o presentacional (sobre la apariencia que tendrá el documento cuando se presente).

4.2. Codificación de caracteres

La codificación de los caracteres de un texto consta de dos fases:

1. Se asigna a cada carácter un número entero positivo llamado *punto de código* o simplemente *código*; por ejemplo: "`a`" $\rightarrow$ 97; "`?`" $\rightarrow$ 63.

2. los códigos numéricos se convierten en bytes asignándoles una determinada secuencia de bits; por ejemplo: 97 $\rightarrow$ `01100001`; 63 $\rightarrow$ `00111111`.

4.2.1. ASCII

Como ya se ha comentado en la página 11, para almacenar textos se ha usado históricamente el estándar ASCII (*American Standard Code for Information Interchange*). Este estándar asigna un número del 0 al 127 a cada carácter del alfabeto latino usado en inglés y usa 7 bits para codificarlo,[1] de forma que permite almacenar un carácter por byte y todavía sobra un bit.[2] La tabla 4.1 muestra algunos ejemplos de códigos ASCII. Los códigos ASCII

[1] Con 7 bits se pueden hacer $2^7 = 128$ combinaciones. Por eso, los códigos asignados por ASCII van del 0 al 127.

[2] Inicialmente este bit se usaba como bit de control para detectar errores en la transmisión de los textos.

del 0 al 31 no corresponden a caracteres imprimibles sino a *caracteres de control* que tienen nombres especiales y se usan para un control rudimentario del formato y de la transmisión de los textos.

El estándar ASCII tiene la limitación de que no permite escribir caracteres propios de muchas lenguas europeas, como, por ejemplo, letras con signos diacríticos (*á*, *ò*, *ç*, *ñ*, *ü*, etc.) o letras especiales como *ß*.

4.2.2. Extensiones de ASCII

Con la llegada de los microordenadores[3] en los años ochenta se decidió ampliar el estándar ASCII, de 7 bits y por lo tanto con $2^7 = 128$ caracteres diferentes, a un código de 8 bits con $2^8 = 256$ caracteres diferentes. El octavo bit o "bit 7"[4]—el primero por la izquierda— es 1 para los nuevos caracteres (numerados del 128 al 255) y cero para los caracteres estándares de ASCII.

Hay varias extensiones de ASCII, cada una dirigida a un conjunto de lenguas concreto que usan (casi) el mismo alfabeto. En nuestra área geográfica se usa normalmente la codificación ISO-8859-1 o *Latin-1* (véase la tabla 4.2); esta codificación sirve para las lenguas siguientes: *afrikaans* (lengua germánica hablada en la República Sudafricana), alemán, inglés, vasco, catalán, danés, escocés, español, feroés, finés, francés, gallego, irlandés, islandés, italiano, neerlandés, noruego, portugués y sueco.[5] El sistema operativo Windows de Microsoft usa la codificación de 8 bits denominada CP-1252, también llamada *WinLatin-1*, que es más amplia que ISO-8859-1, puesto que usa algunos de los códigos 128–159 para caracteres (por ejemplo, usa el código 128 para el símbolo del euro).

Hay otras codificaciones en la familia ISO-8859. Por ejemplo, el albanés, el bosnio, el croata, el checo, el húngaro y el rumano usan una codificación llamada ISO-8859-2 o *Latin-2*; el letón, el lituano y el estonio usan la ISO-8859-4 o *Latin 4*; el ruso usa la ISO-8859-5 que contiene el alfabeto cirílico además del alfabeto latino básico. Por eso, es muy importante conocer qué esquema de codificación de caracteres se ha usado en un documento de texto determinado para poderlo leer correctamente; algunos formatos de texto incluyen esta información dentro del mismo documento.

El hecho de que haya varias maneras de usar los nuevos códigos hace que a veces los textos con caracteres especiales no queden bien cuando pasamos de un procesador de textos (o un editor de textos) a otro (los caracteres de ASCII se ven normalmente bien: los que fallan son los nuevos). Fijaos en que si en un documento ISO-8859-1 se escribe la frase *Què és això?*, donde los caracteres "è", "é" y "ò" tienen códigos por encima de 127 (233,

[3] Los primeros ordenadores que tenían un tamaño que permitía tener uno en casa.

[4] Recordad que en informática es habitual contar empezando por el cero.

[5] Hay una modificación llamada ISO-8859-15, que incluye, entre otros, el símbolo del euro y resuelve algunos problemas referentes al francés y al finés.

CÓDIGO BINARIO	CÓDIGO DECIMAL	CARÁCTER
0000000	0	NUL (carácter nulo)
...	...	...
0001001	9	TAB (tabulador)
0001010	10	NL (nueva línea)
...	...	...
0001101	13	CR (retorno de carro)
...	...	...
0100000	32	(un espacio en blanco)
0100001	33	!
0100010	34	"
...	...	...
0110000	48	0
0110001	49	1
...	...	...
0111000	56	8
0111001	57	9
...	...	...
1000000	64	@
1000001	65	A
1000010	66	B
...	...	...
1011010	90	Z
1011011	91	[
...	...	...
1100000	96	`
1100001	97	a
1100010	98	B
...	...	...
1111010	122	z
1111011	123	{
...	...	...
1111110	126	~

Tabla 4.1: Algunos ejemplos del código ASCII. Los códigos del 0 al 31 no corresponden a caracteres imprimibles sino a caracteres de control.

232 y 242 respectivamente), e intentamos leerlo como si fuera un documento ISO-8859-2, leeremos *¿Quč és aixň?*, porque dos de estos códigos (233 y 242) tienen otra interpretación en esta codificación ("č" y "ň" respectivamente). Por ello, no podemos mezclar en un mismo documento textos en lenguas que usan extensiones de ASCII diferentes.

4.2.3. Unicode

Los códigos de 8 bits como ISO-8859-1 (*Latin-1*) son adecuados para la mayor parte de las lenguas europeas, que se basan en el alfabeto latino con algunas modificaciones, pero hay lenguas en el mundo que tienen sistemas de escritura muy complejos con miles de símbolos diferentes, como por ejemplo el chino o el japonés. Para estas lenguas 256 combinaciones no son suficientes y se han propuesto varias soluciones. *Unicode*[6] (ISO 10646) es un nuevo estándar para codificar prácticamente los caracteres de todas las lenguas del mundo e incluso mezclar varios alfabetos en un mismo fichero.

Unicode usa 31 bits; es decir, permite $2^{31} = 2.147.483.648$ caracteres diferentes. La versión más comúnmente usada de Unicode (BMP, *Basic Multilingual Plane*) tiene 65.534 caracteres; esto comportaría el uso de 2 bytes (16 bits) en vez de uno ($2^{16} = 65.536$), lo que haría que un texto Unicode sencillo fuera el doble de grande que el texto ASCII correspondiente. Para ahorrar espacio, hay métodos de serialización de Unicode, como el UTF-8, que en el caso de las lenguas europeas con alfabeto latino ahorra espacio porque usa un único byte para los códigos ASCII (del 0 al 127, los más frecuentes), y más de un byte para los códigos siguientes (así, además, es compatible con el ASCII). En concreto, UTF-8 usa:

- para los códigos del 0 al 127, 1 byte (compatible con ASCII);
- para los códigos del 128 al 2047, 2 bytes;
- para los códigos del 2048 al 65535, 3 bytes, y así sucesivamente.

4.2.4. Limitaciones

Aunque ampliemos el ASCII a ISO-8859 o Unicode, todavía es muy limitado. Por ejemplo, si queremos que un texto tenga un cierto formato, sólo podremos usar caracteres de control como, por ejemplo, el espacio en blanco, el tabulador, el salto de línea, etc. Por ejemplo, no podremos cambiar fácilmente de tipo o de tamaño de letra, o indicar que una determinada parte del texto es el título de una sección o el texto de una nota a pie de página. En cualquier caso, las extensiones de ASCII (ISO-8859-X) y Unicode todavía se usan en aplicaciones como, por ejemplo, el correo electrónico,

[6] `http://www.unicode.org`

CÓDIGO BINARIO	CÓDIGO DECIMAL	CARÁCTER
10100000	160	(espacio no rompible)
10100001	161	¡
...	...	...
10110101	181	μ
10110110	182	¶
10110111	183	·
10111000	184	¸
...	...	...
11000000	192	À
11000001	193	Á
11000010	194	Â
11000011	195	Ã
11000100	196	Ä
11000101	197	Å
11000110	198	Æ
11000111	199	Ç
11001000	200	È
11001001	201	É
11001010	202	Ê
11001011	203	Ë
11001100	204	Ì
11001101	205	Í
11001110	206	Î
11001111	207	Ï
...	...	...
11100000	224	à
11100001	225	á
11100010	226	â
11100011	227	ã
11100100	228	ä
11100101	229	å
11100110	230	æ
...	...	...
11111111	255	ÿ

Tabla 4.2: Algunos ejemplos de ISO-8859-1 (*Latin-1*). Los códigos del 0 al 127 son como los de ASCII. Los códigos del 128 al 159 no están asignados.

o cuando queremos que un texto —cuyo contenido es mucho más importante que la apariencia— pueda ser leído por cualquier usuario sin importar el procesador de textos que use; los textos de este tipo se denominan a veces *textos planos* y se almacenan normalmente en ficheros con nombres que tienen la extensión `.txt`. Estos textos se pueden producir y leer con cualquier *editor de textos* (véase el apartado 4.6).

4.3. Formato propiamente dicho

Los documentos de texto son en general más ricos que simples secuencias de caracteres; los textos, además de caracteres, contienen información de *formato*. Por eso, es necesaria la asignación de *códigos* (que también se convertirán en bytes) para regular otras características del texto como:

- la apariencia *visual* que tendrá el documento cuando se presente (por ejemplo, "inicio cursivas", "final negritas", "letra de 16 puntos"), o la
- *estructura*, es decir, la organización del contenido del documento (por ejemplo, "título de sección", "lista numerada", "nota a pie de página", "fila de una tabla", etc.)

Para guardar esta información, se usan:

- Por un lado, codificaciones o formatos basados en texto (ISO-8859-X, Unicode, etc.). Tal es el caso del formato SGML (*standardized generalized markup language*), su versión simplificada (y mucho más extendida) XML (*extensible markup language*), el formato HTML (*hypertext markup language*; basado en SGML), el formato RTF (*rich text format*; propuesto por Microsoft y sin relación con SGML o XML), o el lenguaje para impresoras denominado Postscript. Todos estos formatos usan combinaciones especiales[7] de caracteres de texto para indicar estas características de estructuración o de presentación.[8]
- Por otro lado, codificaciones o formatos basados en códigos binarios no interpretables como caracteres. Tal es el caso de los formatos particulares de los procesadores de textos comerciales como por ejemplo Corel WordPerfect o Microsoft Word.[9]

[7] Combinaciones de caracteres poco frecuentes en textos usuales.

[8] Estos caracteres que indican el formato no son normalmente visibles para la persona usuaria mientras redacta o ve el documento, excepto si pide explícitamente que los quiere ver.

[9] Existe una cierta tendencia a considerar el formato de documento de Microsoft Word, con extensión `.doc`, como la manera estándar de enviar documentos de texto anejos a un mensaje electrónico, sin considerar el hecho que este formato es privado y está asociado al uso de un determinado producto no libre y de código fuente cerrado. El formato `.docx` también llamado Office Open XML u OOXML, está mejor documentado y estandarizado y puede ser procesado más satisfactoriamente con procesadores libres y de código fuente abierto.

Como ya se ha dicho, el uso de formatos de texto más avanzados no sólo sirve para determinar la presentación en la pantalla o cuando son imprimidos; como veremos más abajo, en el caso de SGML y XML, el formato sirve para *estructurar* el documento de texto en unidades directamente relacionadas con el contenido del documento, como, por ejemplo, secciones, títulos de sección, listas, párrafos, etc.; esta estructuración interna del documento puede ser usada después para hacer búsquedas de información con la ayuda de la estructura definida, como, por ejemplo, buscar una palabra concreta sólo en títulos de sección, o también para producir una presentación concreta del documento, como veremos más adelante. De hecho, recientemente, con la aparición de XML (veáse el apartado 4.4.2), se observa una tendencia hacia la adopción de formatos de documento estructurados, es decir, no relacionados únicamente con la presentación, sino también con la estructura propia del documento, formatos normalmente concebidos de forma que la presentación deseada se pueda producir a partir de la estructura usando ficheros (llamados *hojas de estilo*) con reglas de estilo bien definidas (véase la sección 4.7).

4.4. SGML y XML

4.4.1. SGML

SGML (*standardized generalized markup language*), el lenguaje estándar generalizado de marcas, tuvo un éxito relativo hasta medios de los noventa; pero la aparición hacia finales de esa década de una versión restringida y simplificada de SGML llamada XML (*extensible markup language*) impulsó notablemente la adopción de los formatos de estructuración de documentos, de tal manera que en la actualidad se usa XML muchísimo más que el SGML original; por eso, nos centraremos en este último formato. En cualquier caso, todavía hay formatos muy importantes que se basan en SGML, como el lenguaje de marcas para hipertextos HTML (véase el apartado 4.4.3), excepto el más reciente (HTML5, estandarizado el 2014) que ya no es una aplicación de SGML. Hay también versiones estándares de HTML conocidas como XHTML, que se basan directamente en XML (véase el apartado 4.4.2).[10]

4.4.2. XML

Marcas

Un documento XML es un documento de texto donde, además del texto propiamente dicho, podemos encontrar *etiquetas* o *marcas* (en inglés *tags*) que dan información sobre la naturaleza y la organización de cada uno de

[10]Nótese que HTML5 también tiene una versión *serializada en XML*, XHTML5.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE EMAIL SYSTEM "http://www.dlsi.ua.es/%7Efsanchez/tt/email.dtd">
<EMAIL>
  <DESTINATARIO>
    <NOMBRE>Mikel L. Forcada</NOMBRE>
    <DIRECCION>mlf@dlsi.ua.es</DIRECCION>
  </DESTINATARIO>
  <REMITENTE>
    <NOMBRE>Felipe Sánchez Martínez</NOMBRE>
    <DIRECCION>fsanchez@dlsi.ua.es</DIRECCION>
  </REMITENTE>
  <FECHA>9 de noviembre de 2015</FECHA>
  <ASUNTO>Capítulo 4 del libro de TT</ASUNTO>
  <TEXTO>
    <P>Mikel, estoy acabando de hacer modificaciones al capítulo
    dedicado a textos y formatos. Cuando acabe te aviso.</P>
    <P>Por favor, envíame los apuntes que preparamos
    sobre traducción automática que no los encuentro. ¡Gracias!</P>
  </TEXTO>
</EMAIL>
```

Figura 4.1: El texto de un mensaje de correo electrónico en XML.

los contenidos del documento; como ya se ha dicho, un documento XML es un documento *estructurado*. Por ejemplo, un documento XML correspondiente a un mensaje de correo electrónico podría tener la apariencia que se muestra en la figura 4.1. La primera línea declara que el documento es un documento XML de la versión 1.0 y que el juego de caracteres que usa es ISO-8859-1 (*Latin-1*; véase el apartado 4.2.2). Las etiquetas que aparecen entre paréntesis angulares indican las diversas partes del documento, denominadas *elementos*. Típicamente, se abren con <*nombre*> y se cierran con </*nombre*>. En el ejemplo, se puede ver que un mensaje de correo (<EMAIL>...</EMAIL>) tiene un destinatario, un remitente, una fecha, un título y un texto; es decir, los elementos pueden contener otros elementos; las marcas funcionan como paréntesis. Siguiendo con la jerarquía de inclusión de unos elementos en otros, tanto el destinatario como el remitente tienen nombre y dirección, y el texto se compone de párrafos (<P>...</P>).

Documentos XML bien formados

Estas son algunas de las características que hacen que un documento XML esté *bien formado*, es decir, sea un documento XML y no otra cosa:

- Cada etiqueta de inicio de elemento de la forma <*nombre*>, <*nombre atributo*="*valor*">, <*nombre atributo1*="*valor*" *atributo2*="*valor*">, etc. (con cero o más asignaciones de valores a atributos) tiene que estar emparejada con una etiqueta de final de elemento de la forma </*nombre*>, sin atributos pero con el mismo nombre.[11] Si el elemento está vacío, <*nombre*...></*nombre*>, también se puede escribir <*nombre*.../>.
- Un elemento puede contener cualquier número de elementos.
- Los elementos no se pueden solapar o cruzar: no es posible escribir, por ejemplo, `<a>texto<b>más texto</a>más texto todavía </b>`.
- El documento contiene un único elemento *raíz* que contiene todos los elementos del texto.
- El documento puede contener comentarios entre `<!--` y `-->` o instrucciones de procesamiento del tipo `<?`*nombre*...`?>` en cualquier lugar excepto dentro de las etiquetas.
- Los valores de los atributos tienen que ir entre comillas dobles (`"`*valor*`"`) o simples (`'`*valor*`'`).
- Un elemento no puede tener dos atributos con el mismo nombre.
- Los caracteres `<` y `&` no pueden aparecer en el texto de los elementos ni de los atributos. Esto es porque `<` indica el comienzo de una etiqueta y `&` el comienzo de una *entidad* como, por ejemplo, `&copy;` que se puede usar para representar el carácter ©: si se necesitan estos caracteres, se tienen que escribir las entidades `&lt;` y `&amp;`, respectivamente.

Como se puede ver, las reglas que definen un documento XML bien formado no dicen qué etiquetas son válidas y cuáles no, o qué atributos puede tener un determinado elemento, o qué elementos pueden ir dentro de un determinado elemento, en qué orden o en qué cantidad.

[11] En SGML se permite que algunos elementos se cierren *implícitamente*, sin necesidad de una etiqueta de final de elemento.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Este es el ejemplo de DTD de EMAIL -->
<!ELEMENT EMAIL (DESTINATARIO+, REMITENTE?, FECHA, ASUNTO, TEXTO)>
<!ELEMENT DESTINATARIO (NOMBRE?, DIRECCION)>
<!ELEMENT REMITENTE (NOMBRE?, DIRECCION)>
<!ELEMENT NOMBRE (#PCDATA)>
<!ELEMENT DIRECCION (#PCDATA)>
<!ELEMENT FECHA (#PCDATA)>
<!ELEMENT ASUNTO (#PCDATA)>
<!ELEMENT TEXTO (P)+>
<!ELEMENT P (#PCDATA)>
```

Figura 4.2: La DTD que define mensajes de correo electrónico como el de la figura 4.1.

Tipo de documentos

Para especificar, para un tipo determinado de documento XML, qué etiquetas son válidas, qué atributos puede tener cada elemento, o qué elementos pueden ir dentro de un determinado elemento, en qué orden o en qué cantidad, se puede usar una DTD (*document type definition* o definición del tipo de documento).[12]

La segunda línea del mensaje de correo de la figura 4.1 especifica el tipo del documento indicando por un lado la etiqueta raíz o principal del documento (`EMAIL`) y el URI (`SYSTEM`) donde se encuentra la DTD. Esta DTD se ve en la figura 4.2; examinamos ahora la DTD línea por línea para comprender cómo se usan las DTD para definir familias (tipos) de documentos en XML:

1. La primera línea declara que la DTD es una DTD de la versión 1.0 y que el juego de caracteres que se usa es el ISO-8859-1 (*Latin-1*):

   ```
   <?xml version="1.0" encoding="ISO-8859-1"?>
   ```

2. La segunda línea es un comentario. Los comentarios empiezan con `<!--` y acaban con `-->` y se pueden situar en cualquier parte de una DTD.

   ```
   <!-- Este es el ejemplo de DTD de EMAIL -->
   ```

3. Las líneas siguientes definen la estructura del documento definiendo sus *elementos*. La línea

[12] Las DTD no son la única manera de especificar familias de documentos XML; otra manera más potente son los llamados *esquemas XML* (en inglés *XML schema*).

```
<!ELEMENT EMAIL (DESTINATARIO+, REMITENTE?, FECHA, ASUNTO, TEXTO)>
```

define el elemento raíz o principal, EMAIL, y especifica que se compone (en el orden especificado) de uno o más DESTINATARIOs (el símbolo + indica que puede haber uno o más), de un REMITENTE opcional (indicado con ?), de una FECHA, de un ASUNTO y de un TEXTO.

4. Un DESTINATARIO del mensaje de correo tiene dos partes: el NOMBRE (opcional) y la dirección de correo (DIRECCION):

```
<!ELEMENT DESTINATARIO (NOMBRE?, DIRECCION)>
```

5. El remitente se define igual:

```
<!ELEMENT REMITENTE (NOMBRE?, DIRECCION)>
```

6. El NOMBRE, la DIRECCION la FECHA y el ASUNTO contienen texto sin marcas (indicado con #PCDATA):

```
<!ELEMENT NOMBRE (#PCDATA)>
<!ELEMENT DIRECCION (#PCDATA)>
<!ELEMENT FECHA (#PCDATA)>
<!ELEMENT ASUNTO (#PCDATA)>
```

7. El TEXTO se compone de uno o más (+) párrafos (P). Si quisiéramos que el texto estuviera compuesto por *cero* o más párrafos usaríamos "$*$" en vez de "+":

```
<!ELEMENT TEXTO (P)+>
```

8. Finalmente, los párrafos contienen texto:

```
<!ELEMENT P (#PCDATA)>
```

Una de las aplicaciones más importantes de las DTD es que sirven para la validación automática de los documentos: un programa *validador* lee la DTD y el documento XML y decide si este último es válido, es decir, si sigue la especificación dada en la DTD. Para que un documento sea válido respecto a una DTD cualquiera, primero tiene que estar bien formado, es decir, tiene que cumplir con las reglas básicas de escritura de documentos XML mencionadas más arriba.

A pesar de que una DTD sirve para la validación automática de documentos XML del tipo que la DTD define, el *significado* de las etiquetas (es decir, qué consecuencias tendrán cuando se procese el documento XML) lo tiene que establecer el programa o los programas que procesarán los documentos. Como ya se ha dicho, este significado puede estar asociado, por

ejemplo, a la manera (*estilo*, véase la p. 66) de presentar el documento cuando se imprime (por ejemplo, los destinatarios del mensaje de correo pueden ir en negrita), pero también podría servir para facilitar el procesamiento de la información (por ejemplo, buscar todos los mensajes que tienen un determinado destinatario, o, en libros codificados en XML, decidir qué partes tienen que ser traducidas automáticamente del español al inglés y cuáles no porque son citas literarias.[13] Incluso ficheros que normalmente no consideraríamos documentos, como por ejemplo las memorias de traducción (véase el capítulo 10) se estructuran de manera estándar usando un formato basado en XML llamado TMX.

Otra aplicación de las DTD es usarlas para facilitar la edición de documentos XML válidos: un editor de documentos XML puede consultar la DTD para sugerir a la persona usuaria el elemento o elementos correctos en el contexto actual, o para emitir un mensaje de error tan pronto como el documento pierda su validez.

4.4.3. (X)HTML

El formato XHTML (*extensible hypertext markup language* o *lenguaje extensible de marcas para hipertextos*) es uno de los tipos de documento que se pueden definir con XML y se corresponde con la versión XML del lenguaje HTML (*hypertext markup language*), este último basado en SGML (formato precursor de XML). Ambos lenguajes se usan para escribir los hipertextos de Internet (véase el capítulo 3) y son lo que interpretan los navegadores de Internet (véase el apartado 3.5).

Tanto en XHTML como en HTML, las marcas tienen un significado determinado. Por ejemplo, (X)HTML indica el comienzo de un segmento de texto destacado (enfatizado) con la marca "`<em>`" (4 caracteres ASCII) y el final con la marca "`</em>`" (5 caracteres). (X)HTML sirve para codificar hipertextos: los enlaces (hiperreferencias) a otros documentos (que a su vez pueden también ser hipertextos) empiezan con "`<a href="` *URI* `">`" —donde *URI* es el identificador del documento enlazado— y acaban con "`</a>`", etc. Los documentos (X)HTML empiezan idealmente con la marca "`<html>`" y acaban con la marca "`</html>`", y tienen, entre otros elementos, un título ("`<title>...</title>`") y un cuerpo ("`<body>...</body>`").

La principal diferencia entre HTML y XHTML es que como este último está basado en XML, el documento tiene que ser XML bien formado y, por lo tanto, no puede haber elementos que se abren pero no se cierran; esto sí que es válido en HTML cuando se trata de elementos vacíos como `img`

[13] Hay un estándar llamado TEI, del inglés *text encoding initiative*, "iniciativa de codificación de textos" (`http://www.tei-c.org`) que usa familias de DTD para definir diferentes tipos de obras (literarias y no literarias). De hecho, existen, por un lado, las antiguas DTD para SGML, y, por otro, las DTD TEI para XML.

o `meta`. Otra diferencia notable es que en XHTML los nombres de los elementos van siempre en minúscula, mientras que en HTML pueden ir en mayúsculas.

El documento XHTML que se muestra en la figura 4.3 se mostraría en un navegador aproximadamente como en la figura 4.4. Como puede verse, la primera línea, que empieza con "`<!DOCTYPE`" declara que el documento es un tipo de documento XHTML estándar según la versión 1.0 *estricta* de XHTML (hay varias versiones). En la tercera línea, la etiqueta "`<html>`" indica el comienzo del documento XHTML, y la etiqueta "`</html>`" del final indica el final del documento. Dentro del elemento `html` encontramos dos elementos: `head` (el *encabezamiento*) y body (el *cuerpo* del documento). Dentro del encabezamiento, un elemento `meta` que no tiene contenido (fijaos cómo se abre y se cierra al mismo tiempo) indica, a través de dos asignaciones del tipo *atributo*="*valor*", que el *juego de caracteres* que usa el documento es el ISO-8859-1, el más común en Europa occidental.[14] Dentro de `head` también encontramos el elemento `title`, que contiene un título que se presentará, cuando se abra el documento con un navegador, en la barra del navegador, *pero no como parte del texto del documento*. Dentro de `body` vemos encabezamientos de nivel 1 (`h1`), encabezamientos de nivel 2 (`h2`), párrafos (`p`), partes del texto destacadas (`em`), y enlaces (`a`). La tabla 4.3 describe algunas de las etiquetas más importantes que se usan en XHTML.

Cuando estamos mirando un documento HTML con un navegador, podemos ver las etiquetas HTML que lo formatean si seleccionamos la opción "ver fuente HTML" ("view HTML source") o similar que hay normalmente en el menú "ver" ("view").

4.4.4. Otros formatos basados en XML

Hoy en día se han popularizado los formatos de documentos basados en XML. Algunos de los formatos basados en XML que interesan a los traductores son: el formato TMX (*translation memory exchange*) que se usa para el intercambio de memorias de traducción (ficheros con extensión `.tmx`, véase el capítulo 10), el formato TBX (*termbase exchange*) para el intercambio de bases de datos terminológicas (ficheros con extensión `.tbx`, véase el capítulo 5) y el formato XLIFF (*XML localization interchange file format*). Este último es un formato creado para la estandarización del formato empleado por las diversas herramientas que se utilizan durante el proceso de *localización* de un producto (véase el capítulo 10).[15]

[14] Para escribir documentos en *checo* o en *coreano*, habría que cambiar parte del valor del atributo `content` porque la codificación ISO-8859-1 no permite escribir en estos idiomas.

[15] La *localización* se puede definir como el proceso de adaptación de un producto a los usos de una región específica del mundo.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type"
          content="text/html; charset=iso-8859-1"/>
    <title>Título del documento</title>
  </head>

  <body>
    <h1>Encabezado de nivel 1</h1>

    <h2>Encabezado de nivel 2</h2>

    <p>Este es el <em>primer</em> párrafo
    de este documento. El navegador decide cómo dividirlo
    en líneas para presentarlo. Idealmente, tendría que
    acabar con una marca de final de párrafo. </p>

    <h2>Otro encabezado de nivel 2</h2>

    <p>Este es el <em>último</em> párrafo
    de este documento XHTML. Los documentos XHTML pueden contener
    <a href="http://www.apertium.org">enlaces</a>
    a otros documentos (X)HTML, locales o remotos. </p>
  </body>
</html>
```

Figura 4.3: Un documento XHTML, tal como lo presentaría un editor de textos normal o usando la opción “view HTML source” (ver fuente HTML) del navegador.

ELEMENT	DESCRIPCIÓN	MÁS INFORMACIÓN
`<html>...</html>`	Contiene todo el documento	
`<head>...</head>`	Encabezamiento	
`<body>...</body>`	Cuerpo	
`<meta.../>`	Información sobre el documento	El elemento está vacío
`<title>...</title>`	Contiene el título del documento	
` `	Salto de línea forzado	El elemento está vacío.
`<h1>...</h1>`	Encabezamiento de nivel 1	
`<h2>...</h2>`	Encabezamiento de nivel 2	
...	...	...
`<h6>...</h6>`	Encabezamiento de nivel 6	
`<p>...</p>`	Párrafo	
`<ul>...</ul>`	Lista sin numerar	Contiene elementos `li`
`<ol>...</ol>`	Lista numerada	Contiene elementos `li`
`<li>...</li>`	Elemento de lista	Puede contener otra lista en su interior.
`<em>...</em>`	Énfasis	
`<strong>...</strong>`	Énfasis fuerte	
`<code>...</code>`	Ejemplo de código	
`<a...>...</a>`	"Ancla"	Si lleva un atributo `href="URI"`, el texto entre "`<a...>`" y "`</a>`" funciona como un enlace al documento que hay en el URI.
`<img.../>`	Imagen	El atributo `src="URI"` indica la dirección donde está la imagen. El atributo `alt="`*texto*`"` describe la imagen con palabras. El elemento está vacío.

Tabla 4.3: Algunos elementos básicos de XHTML, la versión XML de HTML.

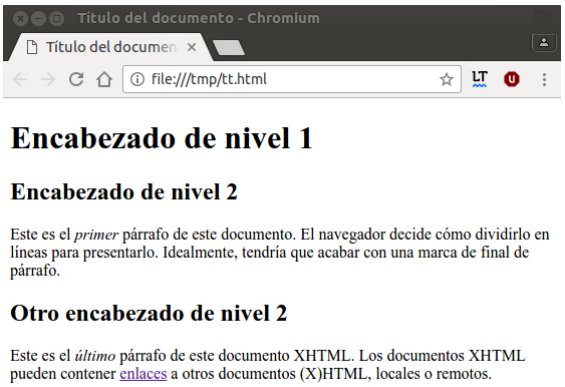


Figura 4.4: El documento XHTML de la figura 4.3, visto a través de un navegador de Internet.

Además de los formatos descritos en el párrafo anterior hay dos formatos usados por los procesadores de textos más modernos que están basados en XML. Estos formatos son OpenDocument y Office Open XML.

OpenDocument

OpenDocument es un formato de archivos abierto y estándar para almacenar, entre otros, textos (ficheros con extensión `.odt`) hojas de cálculo (ficheros con extensión `.ods`) y presentaciones (ficheros con extensión `.odp`). Este formato es el empleado por defecto por las aplicaciones ofimáticas LibreOffice y Openoffice.org y consiste en varios documentos XML —para el contenido, los estilos usados en el documento, etc.— comprimidos con ZIP.[16]

Office Open XML

Office Open XML, también conocido como OOXML u OpenXML, es otro formato estándar —impulsado por Microsoft— basado en XML que se utiliza para almacenar textos (ficheros con extensión `.docx`), hojas de cálculo (ficheros con extensión `.xlsx`) y presentaciones (ficheros con extensión `.pptx`). Igual que OpenDocument, un archivo OpenXML consiste en varios documentos XML comprimidos con ZIP.

[16] ZIP es un formato para el almacenamiento de ficheros comprimidos; los ficheros de este tipo suelen tener la extensión `.zip`.

```
{\rtf1\ansi\ansicpg1252
                              [...]

\par
{\b T\'edtol en negretes}\par
Text del par\'a0graf en lletra normal amb alguns incisos
{\i en cursives} i una marca de final de par\'a0graf al
final.\par
Els car\'a0cters que no pertanyen a l'ASCII est\'a0ndard
s'indiquen amb codis especials (en aquest cas s'ha usat
ANSI, amb {\i codepage} 1252, com es veu al principi del
document), com per exemple en el mot
{\i ling\'fc\'edstica}.\par

                              [...]
```

Figura 4.5: Parte de un documento de texto en format RTF.

4.5. Otros formatos

4.5.1. RTF

RTF (*rich text format*, es decir, *formato de texto rico*) fue un formato impulsado por la empresa Microsoft para facilitar el intercambio de documentos entre procesadores de textos manteniendo el formato, y que todavía se usa a veces. RTF también tiene etiquetas, que empiezan normalmente por una barra invertida (\); pero los ámbitos de acción de las etiquetas están delimitados por llaves ("{...}") en vez de por parejas de etiquetas; por ejemplo, un segmento en negritas se indica con "`{\b...}`", mientras que en HTML se usa "`<B>...</B>`". La figura 4.5 muestra parte de un documento RTF, en la que se ven algunas instrucciones del encabezamiento (empezando por "`{\rtf1`...") y donde también se observa la manera especial como se codifican algunos caracteres.

4.5.2. PDF

PDF (del inglés *portable document format*, formato portable de documento) es otro formato desarrollado para capturar completamente las características presentacionales de los documentos. En PDF, el documento se muestra exactamente con la misma apariencia independientemente del ordenador, sistema operativo o aplicación que se use para verlo. Los documentos PDF pueden almacenar, además del texto, tipo de letra, gráficos,

sonidos, etc. Este formato fue impulsado en los años noventa por la empresa Adobe, que ofrece en la actualidad un programa gratuito[17] denominado Adobe Acrobat Reader DC— para visualizar los documentos;[18] para crearlos podemos usar programas especializados o cualquier procesador de textos que permita *exportar* (realmente *imprimir*) nuestro documento a PDF.

4.6. Procesadores de textos

Un *procesador de textos* es un programa que permite crear y modificar documentos de texto informatizados. También se pueden usar *editores*: la diferencia entre un procesador de textos y un *editor* es que este último programa es un procesador de textos planos (sin información de formato, etc.) que normalmente se usa para preparar textos en algún lenguaje artificial (por ejemplo, programas escritos en algún lenguaje de programación) que servirán de entrada para otro programa, o textos muy sencillos donde el formato no es crucial, como un mensaje electrónico sencillo.

El procesamiento de textos también se denomina *tratamiento de textos* (paralelamente al francés, *traitement textes*). En inglés, el énfasis es sobre las palabras: *word processing*.

Por supuesto, esta sección no pretende instruir en el uso de ningún procesador de textos concreto, sino que quiere describir brevemente algunas características comunes a los procesadores de textos que se usan en la actualidad. De hecho, el uso de los procesadores de texto se aprende mucho mejor en el laboratorio; además, en vista del hecho de que los procesadores de textos cambian constantemente, quizás es mejor no aprender a usar un procesador concreto sino a buscar en cada procesador las herramientas que necesitamos. Esto es posible porque la mayor parte de los procesadores van provistos de manuales o de sistemas de ayuda en línea; algunos tienen incluso "asistentes" que observan lo que hace la persona usuaria y le sugieren —con más o menos fortuna— posibles acciones en cada momento.

En cuanto a la *apariencia* del programa, la mayor parte de los procesadores de texto se manifiestan básicamente como una o varias ventanas, cada una de las cuales muestra una sección de alguno de los documentos de texto informatizados que estamos creando y modificando (los documentos que tenemos *abiertos*). La tendencia actual favorece que el texto se muestro tan parecido como sea posible a la versión impresa que se producirá, en cuanto a formato, tipo de letra, etc. (en inglés, este concepto de fidelidad visual se resume con la palabra *wysiwyg*, hecho con las siglas de "what

[17] pero no libre ni de código fuente abierto

[18] Hay alternativas libres y de código fuente abierto como por ejemplo Sumatra PDF, Evince, Okular, etc. Incluso los mismos navegadores vienen ya con visores de PDF.

you see is what you get", es decir, "lo que veis es lo que obtendréis"); la sección 4.7.1 describe algunos problemas derivados de esta tendencia.

En cuanto a la *operación*, los procesadores de texto asumen que la mayor parte de los caracteres que tecleamos se tienen que insertar detrás del carácter que actualmente se encuentra destacado con una marca llamada *cursor* de texto (puede ser diferente del cursor o apuntador que indica la posición virtual del ratón en la pantalla), o bien lo tienen que sobrescribir. Sin embargo, se reservan determinadas teclas (algunas sencillas, y otras en combinación con las teclas especiales "Alt" o "Control") para hacer operaciones, algunas muy básicas como, por ejemplo, mover el cursor de texto o borrar caracteres y otras más complejas, como, por ejemplo, pegar un bloque de texto que habíamos borrado previamente o guardar el texto completo en el disco.[19] Pero muchas de estas operaciones, conjuntamente con otras que no se usan tan a menudo, también están accesibles mediante *menús*; los nombres de estos menús suelen estar situados típicamente en la parte de arriba de la ventana: si se hace clic con el ratón, se despliegan y nos muestran las opciones que contienen, que podemos elegir con el ratón.

Sobre la búsqueda de palabras. Algunos procesadores de textos permiten buscar usando las llamadas *expresiones regulares*, que permiten, mediante caracteres especiales llamados *comodines* (inglés *wildcards*), buscar todas las palabras y todas las porciones de texto que siguen un patrón determinado. Por ejemplo, una búsqueda con la expresión regular `pres*a` encontraría las palabras *prea*, *presa*, *pressa*, *presssa*, etc., o la expresión regular `<[^>]+>` que encontraría todas las etiquetas del estilo de XML, puesto que empiezan por <, tienen uno o más (+) caracteres que *no* (^) son >, y acaban con >. Para saber más sobre expresiones regulares podéis consultar la página de la Wikipedia `https://es.wikipedia.org/wiki/Expresi%C3%B3n_regular`.

4.7. Contenido, estructura y presentación de los documentos

4.7.1. El problema *wysiwyg*

La mayoría de los procesadores de textos actuales son *wysiwyg* en el sentido explicado más arriba: el texto que se edita se presenta gráficamente en la ventana prácticamente igual a como se verá en el papel cuando lo

[19] Estas teclas y combinaciones de teclas que permiten un acceso rápido a operaciones rutinarias se suelen denominar en inglés *hotkeys*; por ejemplo, en Windows, la combinación control–X recorta el texto seleccionado, la combinación control–V inserta un texto previamente recortado, etc.

enviemos a la impresora; esto ha facilitado enormemente el acceso de todo el mundo a los procesadores de textos. Pero el esquema *wysiwyg*, completamente generalizado desde mitad de los ochenta, tiene también, como veremos, sus inconvenientes. La persona escritora tiende a centrarse en los atributos *visuales* del texto (tipo y tamaños de letra, márgenes, etc.), puesto que confía en que una buena *presentación* transmitirá a las personas lectoras la estructura *lógica* que la persona escritora tiene en la cabeza. Con el documento, por lo tanto, sólo se guardará esta información de presentación, prácticamente sin ninguna indicación de la estructura lógica de los contenidos. Imaginaos las siguientes situaciones problemáticas:

> Vladimir ha decidido que los títulos de sección del informe anual que le han encargado estarán en Helvetica de 14 puntos, negrita y los de subsección en Arial de 12 puntos, negrita cursiva. A su directora no le gustan así y se los ha hecho cambiar a Lucida Sans de 14, negrita y Lucida de 12, negrita sin cursivas. Como el informe tiene que estar acabado para mañana por la mañana, Vladimir se queda en la oficina hasta las 11 de la noche, cambiando uno por uno los tipos de letra de los títulos de secciones y subsecciones. El día siguiente, por la mañana, Marina, la directora, le pasa un documento con una sección más que se tiene que insertar entre la 4 y la 5. Vladimir no puede ir a almorzar: tiene que cambiar los números de secciones y subsecciones a partir de la 5 y repasar si se tiene que cambiar alguna referencia que se haga desde una parte del texto a una sección por su número.

> Nos han encargado traducir un texto informatizado. En la lengua origen es costumbre poner en *cursivas* tanto las palabras extranjeras ("*Sprachgefühl*") como los términos cuando se definen por primera vez ("Un *byte* es..."), se sangra la primera línea de todos los párrafos, y los números de sección traen un punto al final ("1.1. Introducción"), pero en la lengua de llegada los términos nuevos van en negritas ("Un **byte** es ..."), se sangra la primera línea de todos los párrafos excepto la del primer párrafo de una sección, y los números de sección no llevan punto al final ("1.1 Introducción"). El texto ha sido traducido manteniendo las convenciones de la lengua origen: para hacerlo adecuado a la lengua de llegada, nos toca, por un lado, ir mirando uno por uno los segmentos de texto en cursivas, decidir si son definiciones, y cambiarlos a negritas si hace falta; por otro lado, nos toca ir quitando el puntito final de todos los números de sección.

En estos dos casos, si la persona que escribió los textos sólo codificó información relativa a la presentación visual no podremos evitar hacer los

trabajos tediosos descritos. Se podría decir que si quien escribe se deja llevar por la filosofía "what you see is what you get" acaba con "what you see is *all* you get", es decir, sólo tiene lo que ve. Pero la mayoría de los procesadores de textos *wysiwyg* actuales permiten un cierto nivel de codificación de la *estructura*, a través de los llamados *estilos*:[20] hay *estilos de párrafo* (párrafo del cuerpo de texto, encabezamientos de varios niveles, etc.) y estilos *de carácter* (definiciones, énfasis, énfasis fuerte, texto de ordenador, etc.). A cada estilo se le asignan unas determinadas características de presentación: por ejemplo, los encabezamientos de nivel 2 van en Helvetica de 14 puntos negrita y numerados automáticamente con el número de la sección de nivel 1, un punto y el número de la sección de nivel 2; las definiciones van en negrita y el énfasis en cursiva, etc. El procesador de textos aplica automáticamente las mismas características de presentación *a todos los segmentos del documento* que tienen el mismo estilo. Esto resolvería las situaciones problemáticas explicadas más arriba.

En el primer problema, si se hubieran usado los estilos como se indica, la numeración y el estilo de las secciones se determinaría automáticamente y sólo habría que indicar (sólo una vez) qué tipo de letra corresponde a los títulos de sección; además se renumerarían automáticamente todas las secciones. Si las referencias de unas secciones a otras se hubieron hecho usando referencias cruzadas simbólicas (muchos procesadores de textos las permiten), también se actualizarían automáticamente.

En el segundo problema, si el documento hubiera contenido información sobre qué términos son definiciones y cuáles son palabras extranjeras, sólo habría que cambiar el estilo de las definiciones y todas quedarían en negritas. Por otro lado sólo habría que indicar que no es necesario el último punto en los números de sección y todos pasarían automáticamente al formato deseado.

Estos ejemplos ilustran la conveniencia de que los autores de los documentos se centren más en la estructuración lógica del contenido del documento que escriben. Después, sólo hay que indicar al procesador cuál tiene que ser la presentación de cada elemento de esta estructura lógica y obtendremos la presentación deseada.

4.7.2. Hojas de estilo

En XML y HTML, esta separación entre la estructura del contenido y la presentación de un documento se ejecuta a través de especificaciones llamadas *hojas de estilo*. Uno de los tipos más sencillos de hojas de estilo son las llamadas hojas de estilo en cascada[21] (CSS, *cascaded style sheets*) que

[20] Esta es la denominación usada por *Word* y por los procesadores libres y de código abierto *Openoffice.org* y *LibreOffice*.

[21] Más información en `http://www.w3c.org/Style/CSS/`.

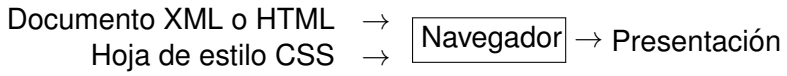


Figura 4.6: Presentación de documentos XML y HTML con hojas de estilo CSS.

se usan sobre todo con navegadores y HTML, aunque también se pueden usar para presentar XML directamente en los navegadores.

Las hojas de estilo CSS asignan características de presentación a cada elemento del documento. Por ejemplo, la orden CSS

```
h2 {   display : block ;
       font-size : large ;
       font-family : sans-serif ;
       text-align : left ;
       margin-top: 0.2cm ;
       margin-bottom : 0.2cm ; }
```

indica que todos los encabezamientos de segundo nivel (`h2`) de (X)HTML se visualizan (`display`) como bloques de texto separado (`block`), con un tamaño de letra (`font-size`) grande (`large`) de la familia *sans serif*, alineado (`text-align`) a la izquierda, y con márgenes superior e inferior de 0,2 cm.

Las hojas de estilo se pueden usar también para visualizar documentos XML directamente en los navegadores más recientes. Por ejemplo, podemos hacer que la presentación visual del mensaje de correo electrónico de la figura 4.1 tenga un encabezamiento con el texto "*Mensaje de correo*" centrado, grande y en negritas con esta orden CSS (con comentarios entre `/*` y `*/`):

```
EMAIL:before {                    /*Antes del EMAIL*/
   content : "Mensaje de correo" ; /*El texto deseado*/
   display : block ;               /*como un bloque*/
   font-weight : bold ;            /*en negrita*/
   text-align : center ;           /*centrado */
   font-size : x-large ;           /*y con letra extragrande*/
}
```

En el caso de las hojas de estilo CSS, el esquema de uso es el que se indica en la figura 4.6: el navegador lee el documento HTML o XML, aplica los estilos de la hoja CSS, y genera una presentación. La hoja de estilo CSS puede estar en el mismo fichero que el documento HTML o XML, o en un fichero externo.

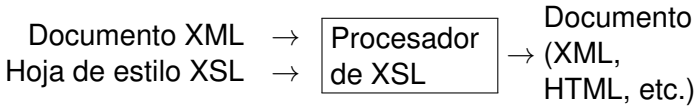


Figura 4.7: Transformación de documentos XML con hojas de estilo XSL.

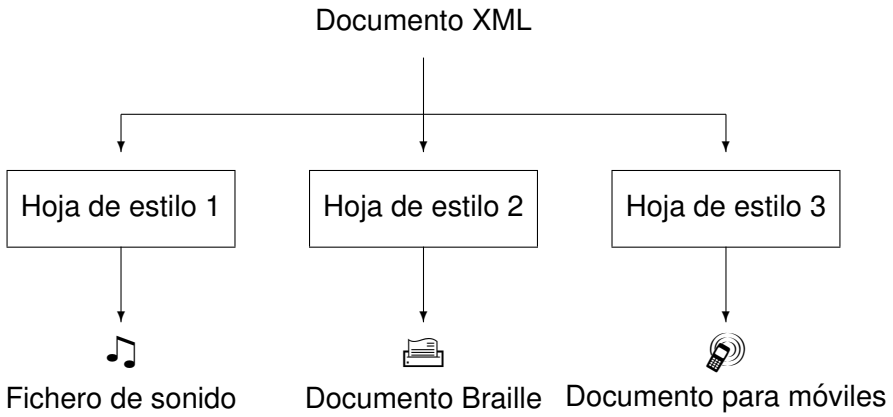


Figura 4.8: Obtención de tres presentaciones diferentes de un único documento XML mediante hojas de estilo.

Para la presentación de documentos XML, existe un lenguaje de programación de hojas de estilo mucho más potente que CSS llamado XSL[22] (*extended stylesheet language*) que permite *transformar* un documento XML (con etiquetas que indican la estructura del contenido) en otro documento XML, HTML o de cualquiera otro formato (Postscript, PDF, RTF, etc.), por ejemplo, para presentarlo visualmente (véase la figura 4.7). Los navegadores más recientes ya son capaces de aplicar hojas de estilo XSL a páginas *web* escritas en XML y presentarlas cómo si estuvieron escritas originalmente en HTML.

Este tipo de presentación visual no es la única transformación posible que podemos obtener con hojas de estilo: como se muestra en la figura 4.8, para un mismo documento XML podemos generar un conjunto de *vistas* de su contenido en diferentes medios (*media*) sólo usando la hoja de estilo adecuada.

[22]Más información en `http://www.w3c.org/Style/XSL/`.

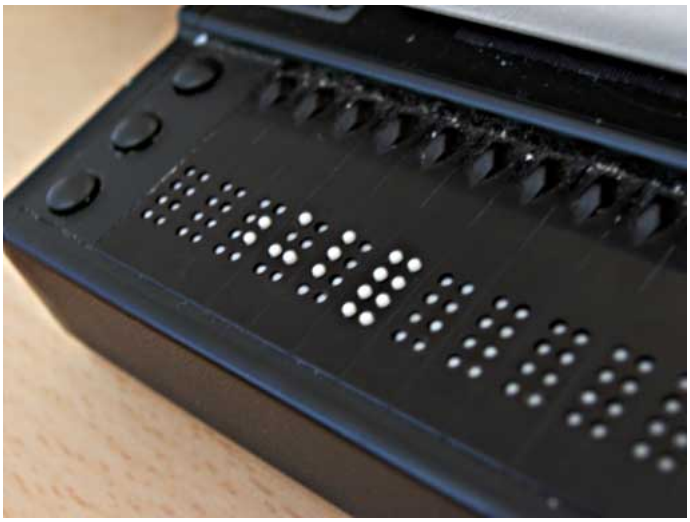


Figura 4.9: Tablero braille (imagen tomada de la entrada *Refreshable Braille Display* de la Wikipedia en inglés)

4.7.3. Accesibilidad

En casi toda la discusión anterior hemos hablado de presentación refiriéndonos siempre a un medio visual, de forma que hemos excluido, por ejemplo, a las personas que tienen discapacidades o limitaciones relacionadas con el sentido de la vista (pueden no ver nada o ver muy mal, o sufrir ataques epilépticos cuando ven una imagen que cambia rápidamente de color). Cuando presentamos un documento visualmente, intentamos que la representación visual de las diversas partes del documento comuniquen la estructura lógica del contenido a la persona lectora, pero, ¿cómo presentamos la estructura lógica de un documento a una persona ciega? Mecanismos como los tipos o tamaños de letra o la forma o el alineamiento visual de los párrafos, listas o tablas no le sirven; esta persona quizás quiere acceder a los documentos mediante un tablero Braille (una especie de pantalla táctil donde se forman los signos del alfabeto de los invidentes, véase la figura 4.8) o mediante un sistema de síntesis de voz que lea la página en voz alta.

Si quien ha escrito el documento sólo ha codificado la estructura lógica que tenía en su cabeza mediante indicadores visuales de formato (negritas o cursivas para el énfasis, párrafos de una línea en letra más gorda para títulos, etc.) será difícil transformar este formato para otra presentación.

Pero no sólo las personas con discapacidades visuales pueden tener problemas; quien lee un documento lo puede estar haciendo a través de una pantalla de texto (no gráfica) o pequeña (como la de un teléfono móvil),

o a través de una conexión muy lenta a la red, o puede estar en una situación en la cual sus ojos estén ocupados (por ejemplo, cuando conduce un vehículo). La presentación de documentos en estos medios tiene problemas similares.

Si el énfasis en el documento ha sido almacenado como énfasis y no con letra negrita o cursiva, si los títulos de sección están indicados como tales y no porque son párrafos de una única línea en negritas gordas, será mucho más fácil transformarlo para presentarlo en un medio no visual o en una pantalla reducida o limitada, por ejemplo, usando *hojas de estilo* especialmente concebidas para la presentación *aural* (sonora), *táctil* (Braille), etc.

La separación de la estructuración del contenido, por un lado, y de los mecanismos de presentación del documento, por otro, facilita la *accesibilidad* al documento a través de varios medios a personas discapacitadas o en situaciones especiales.

Para saber más sobre tecnologías auxiliares para la generación de textos

Reconocimiento automático del habla. El *reconocimiento automático del habla* (RAH) se puede definir como la producción de textos informatizados —en *tiempo real*, es decir, tan instantáneamente como sea posible— a partir de la voz humana (véase Samuelson-Brown 1996). El RAH de propósito general está todavía muy lejos de ser perfecto, y, de hecho, es todavía un campo de investigación activo, aunque recientemente se ha incorporado bastante satisfactoriamente a dispositivos como por ejemplo teléfonos móviles (por ejemplo, los dispositivos con sistema operativo Android permiten hacer búsquedas por voz). En cambio, el RAH para un propósito específico (por ejemplo, la consulta telefónica de horarios de trenes o de las condiciones del tráfico) está mucho más avanzado. La mayor parte de la inversión de la comunidad internacional en RAH es, por razones obvias, sobre el inglés.

El RAH genera texto a partir de la voz recogida a través de un micrófono utilizando un dispositivo de captura para digitalizarla y después un sistema de reconocimiento automático de la voz (*automatic speech recognition*) para detectar fonemas, sílabas o palabras completas (depende del sistema concreto) y traducirlas posteriormente a un texto informatizado. Hay sistemas de reconocimiento *independientes del hablante* y sistemas *dependientes del hablante* (los últimos normalmente tienen que ser *entrenados* por la persona antes de su uso). El RAH es especialmente difícil por la gran variabilidad acústica que presentan los fonemas:

- según el contexto articulatorio (por ejemplo, no es igual el sonido del fonema palatal catalán representado por el dígrafo *ig* en "paseig curt" —sordo— que en "passeig allargat" —sonoro—);
- según el hablante (cada persona tiene unos órganos fonadores de forma diferente —acústicamente diferentes— y procesos de producción del habla diferentes —por ejemplo, hay quién habla más despacio y quien habla muy de prisa—);

- según el dialecto del hablante (por ejemplo, los valencianos hacemos africadas las *j* que en catalán central son palatales fricativas sonoras).
- según el estado emocional del hablante, etc.

Es un hecho muy establecido que, para superar estas dificultades, los humanos hacemos un uso muy intensivo de los conocimientos lingüísticos que tenemos sobre el idioma que estamos escuchando y del contexto comunicativo: así, si oímos decir "*percà nom* passes l'*antre xoc* de *craus*?" a un amigo cuando vemos que no puede abrir el coche, entendemos perfectamente que nos quiere decir *¿Por qué no me pasas el otro juego de llaves?*, o si oímos decir en voz alta "mu han dim moltis baltes" es muy probable que entendamos claramente "me lo han dicho muchas veces" a pesar de los cambios fonéticos, puesto que inconscientemente buscamos la interpretación correcta más cercana a lo que hemos oído (en el contexto concreto en que se diga la frase). Considerad este doblete inglés clásico sobre el tema: *people can easily recognize speech* no es muy diferente de *people can easily wreck a nice beach*; otro doblete lo forman las expresiones *sax and violins on TV* y la más verosímil *sex and violence on TV*. Los resultados del RAH son especialmente dependientes de las particularidades lingüísticas de la lengua involucrada y el éxito depende de la existencia de un buen *modelo de lengua* —rápido y conciso, es decir, computacionalmente eficiente— que simule la parte no contextual de la comprensión humana y permita obtener el texto más probable en un idioma determinado a partir del texto en bruto producido por el sistema de RAH. La mayor parte de los sistemas usan vocabularios grandes y modelos estadísticos.

Reconocimiento automático de textos escritos. El *reconocimiento automático de textos escritos* (RATE) se puede definir como la producción de textos informatizados a partir de textos manuscritos o tipografiados. En el caso de textos tipografiados la tarea es mucho más sencilla; en el caso de manuscritos, la complejidad es comparable a la del reconocimiento del habla.

El RATE genera un texto informatizado a partir de un documento impreso, usando un escáner (en inglés *scanner*) y un programa de reconocimiento óptico (también se conoce como *automático*) de caracteres (OCR, *optical character recognition*). Primeramente, el documento impreso es leído (escaneado) usando el escáner, y se genera un fichero que contiene la imagen digital (por ejemplo, una parrilla muy fina de cuadrados blancos y negros). Después, el programa de OCR lee la página, descubre donde están los párrafos, las líneas y, finalmente, los caracteres concretos, y los transforma en un texto informatizado (normalmente bastante imperfecto, especialmente si es manuscrito). Como en el caso del reconocimiento del habla, es crucial el uso de información sobre el idioma concreto (diccionarios, estadística sobre las secuencias de letras) para corregir los errores del OCR. Por ejemplo, si un programa de lectura automática de textos produce por error el texto "4ixò 6s uua mcrda", huelga decir que lo leemos sin demasiados problemas, a pesar de los errores en todas las palabras; esto es gracias a nuestros conocimientos sobre las secuencias de letras comunes en catalán.

4.8. Cuestiones y ejercicios

1. Para validar un documento XML necesitamos ...

   a) ... otro documento XML, este último con las marcas sin contenido.

   b) ... una hoja de estilo CSS.

c) ... una definición de tipo de documento (DTD).

2. ¿Cómo se indica en una DTD que el elemento `teixit` contiene opcionalmente los elementos `grandaria` y `color` en este orden?

   a) `<!MARK teixit grandaria, color #OPTIONAL>`

   b) `<!MARK teixit (grandaria?,color?)>`

   c) `<!ELEMENT teixit (grandaria?,color?)>`

3. Un documento XML es *válido* ...

   a) ... si sólo usa los nombres de elementos definidos en la DTD; el resto de las directrices de la DTD sólo sirven para hacer documentos *bien formados*.

   b) ... si sólo usa las marcas válidas de los documentos HTML.

   c) ... si sigue las reglas de la DTD cuando incluye un elemento dentro de otro y, además, no incluye ningún elemento no definido en la DTD.

4. Un texto informatizado se caracteriza principalmente ...

   a) ... por su formato, por un lado, y por el juego de caracteres con que está codificado, por otro.

   b) ... por la versión del sistema operativo y el procesador de textos con que ha sido escrito.

   c) ... por la hoja de estilo que indica los aspectos estéticos de su presentación.

5. ¿Qué hace que el siguiente fragmento de XML esté *mal formado*?

   ```
   <tit int=hi>Zjuknim agarnow</tit>
   ```

   a) Entre `tit` y > no puede haber nada.

   b) La etiqueta `tit` no es válida en XML; tendría que ser `title`.

   c) Si hay algún atributo, el valor tiene que ir entre comillas.

6. Si en una DTD encontramos las reglas

   ```
   <!ELEMENT taula (capçalera?,fila+)>
   <!ELEMENT fila (casella*)>
   <!ELEMENT casella (#PCDATA|taula)*>
   ```

   ¿cuál de las tres situaciones siguientes es válida de acuerdo con esta DTD?

a) `<taula></taula>`

b) `<taula><fila><casella>zz<taula><fila></fila></taula>zz </casella><fila></taula>`

c) `<taula><fila><casella>zz</casella><casella>ww</casella> </fila></taula>`

7. ¿Qué indica el fragmento `encoding="...&quot;` en la primera línea (`<?xml...?>`) de un documento XML?

   a) La versión de XML.

   b) Dónde está la DTD necesaria para validarlo.

   c) Cuál es el juego de caracteres que usa el documento XML.

8. Cuántos bytes ocupa el segmento de XML siguiente:

   `<qq>ww</qq>`

   a) 11 como mínimo, dependiendo de la codificación.

   b) 11, independientemente de la codificación.

   c) 4 exactamente.

9. Cuando las marcas de formato sólo especifican el *contenido* de un documento (identificando las partes y la estructura de cada una), ¿cómo se asigna una *presentación* determinada al documento?

   a) Con una o más hojas de estilo.

   b) Con una codificación de caracteres (p.e., Unicode o ISO-8859-1).

   c) No se puede asignar presentación.

10. ¿Qué se conserva de ASCII en los sistemas de codificación de caracteres más avanzados como Unicode UTF-8, ISO-8859-1 (*Latin-1*), etc.?

    a) Los caracteres y sus números de código.

    b) Los caracteres, pero con números de código diferentes.

    c) No queda nada. Se ha reorganizado toda la codificación.

11. Estamos en Eslovaquia, donde se usa la codificación de caracteres ISO-8859-2 (*Latin-2*). Desde Alacant, nos envían un documento de texto plano, escrito con la codificación ISO-8859-1 (*Latin-1*) y lo abrimos como si fuera ISO-8859-2 (*Latin-2*). ¿Qué pasa?

    a) No vemos bien ninguna letra: todo son símbolos extraños e ininteligibles.

b) Vemos bien todas las letras excepto las acentuadas, las que llevan diéresis, la *ñ* o la *ç*: en su lugar aparecen otros símbolos o letras típicas de las lenguas de Europa del Este.

c) Vemos bien todas las letras excepto las acentuadas, las que llevan diéresis, la *ñ* o la *ç*: en su lugar aparecen las versiones sin acento, la *n* o la *c*.

12. ¿Qué es RTF?

    a) Un esquema avanzado de codificación de caracteres.

    b) Un formato abierto de intercambio de memorias de traducción.

    c) Un formato abierto para intercambiar documentos de texto entre procesadores de textos.

13. Un documento HTML tiene un enlace con el texto "Más información" y con URI de destino `http://www.detalls-e.com/mes.html`. ¿Cómo es este enlace en HTML?

    a) `<a href="http://www.detalls-e.com/mes.html">Más información</a`

    b) `<a href="Més informació">http://www.detalls-e.com/mes.html</a>`

    c) `<a htxt="Més informació" href="http://www.detalls-e.com/mes.html">`

14. ¿Dónde va el título de un documento HTML (el que se muestra en *la barra* del navegador)?

    a) En un elemento `title` dentro de `head`.

    b) En un elemento `title` dentro de `body`.

    c) En un elemento `h1` dentro de `head`.

15. Si los caracteres de un texto están codificados usando el juego de caracteres ISO-8859-1 (*Latin-1*), ¿qué códigos tienen las letras de la `A` a la `Z`?

    a) Depende del formato del texto (HTML, etc.).

    b) Los mismos que en la codificación ASCII.

    c) Los que tenían en la codificación ASCII más 128.

16. ¿Cuántos bytes ocupa como mínimo el siguiente fichero HTML?

```
<html><body><p>Texto.</p></body></html>
```

    a) 11

    b) 39

c) 76

17. En la codificación de caracteres ISO-8859-1 (*Latin-1*), todos los caracteres acentuados del español o del catalán tienen códigos ...

    a) ... entre 0 y 127.

    b) ... entre 128 y 255.

    c) ... mayores que 256.

18. Un texto codificado en ISO-8859-1 (*Latin-1*) tiene 1000 caracteres justos (contando los espacios en blanco y los saltos de línea). ¿Cuántos bytes ocupa?

    a) 1000 exactamente, 1 por carácter.

    b) 2000 exactamente, 2 por carácter.

    c) entre 1000 y 2000, entre 1 byte y 2 bytes por carácter.

19. En XML, si se abre un elemento con la marca `<frase>`, ¿con qué marca se cierra?

    a) Con `</frase>`.

    b) Con `<frase>`.

    c) Automáticamente cuando se abre cualquier otro elemento.

20. Si en un documento XML encontramos la situación

    ```
    <rec><id>Zork</id><addr>Zmeggs</addr></rec>
    ```

    y una DTD define el elemento `rec` con la regla

    ```
    <!ELEMENT rec (id,up?,addr*)>
    ```

    ¿Puede ser que el documento sea válido según la DTD?

    a) Depende de cómo sea de estricto el programa validador.

    b) No, porque esta situación no es válida.

    c) Sí, si el resto del documento es válido.

21. ¿Qué vemos si abrimos un texto HTML con un editor de textos sencillo como el *Bloc de notas* o *Notepad* de Windows?

    a) El texto HTML pero sin las marcas entre "<" y "/>".

    b) El texto HTML tal como está hecho por dentro, con las marcas entre "<" y ">" y todo.

    c) Una pantalla en blanco.

22. ¿En que se diferencian dos extensiones de ASCII diferentes?

    a) En los caracteres asignados a los 256 códigos.

    b) En los caracteres asignados a los códigos del 0 al 127.

    c) En los caracteres asignados a los códigos del 128 al 255.

23. Si en una DTD encontramos la regla

    ```
    <!ELEMENT cv (nom,any?,ob+)>
    ```

    ¿cuál de las tres situaciones siguientes no es válida de acuerdo con esta DTD?

    a) `<cv><nom>Pere</nom><any>1992</any></cv>`

    b) `<cv><nom>Pere</nom><ob>Escrits</ob></cv>`

    c) `<cv><nom>Pere</nom><any>1992</any><ob>Crits</ob><ob>Plors</ob>`

24. En un documento HTML queremos que la frase *este documento* sea un enlace al documento que tiene el URI `http://www.uc.za/t.html`: ¿cuál de las siguientes porciones de HTML es la correcta?

    a) `<a url="http://www.uc.za/t.html">este documento</a>`

    b) `<a href="http://www.uc.za/t.html">este documento</a>`

    c) `<link url="http://www.uc.za/t.html">este documento</link>`

25. El fragmento de documento HTML "`<strong><em>link</strong></em>`" tiene un error. ¿Cuál es la causa?

    a) El nombre de las marcas no es válido, porque no indica ninguna información sobre el contenido.

    b) El orden de las marcas de apertura y cierre no es correcto.

    c) No se ha indicado el valor del atributo `href` del elemento `em`.

26. La longitud media de una palabra en gondavés es de 5,5 caracteres y la edición electrónica de *Gundhawól Vlâj* ("La Voz de Gondàvia"), tiene unas 100.000 palabras diarias como media. Si el gondavés se escribe usando la codificación ISO-8859-1 (*Latin-1*), ¿cuántos ejemplares del diario se pueden guardar en un CD-ROM?

    a) Más de dos años.

    b) Un ejemplar sólo.

    c) Un mes aproximadamente.

27. ¿Cómo se indica en una DTD que el elemento `<fitxa>` contiene obligatoriamente los campos `<nom>` y `<tel>` y, opcionalmente, el campo `<email>`?

a) `<!ELEMENT fitxa (nom,tel,email?)>`

b) `<!ELEMENT fitxa (nom,tel,email+)>`

c) `<!ATTLIST fitxa nom CDATA #required tel CDATA #required email CDATA>`

28. En XML, ¿que quiere decir `<mang/>`?

    a) No quiere decir nada, porque no hay ningún elemento que se llame `mang`.

    b) Lo mismo que `<mang></mang>`.

    c) No quiere decir nada, porque tendría que ser `</mang>`.

29. ¿Cuál de estos tres elementos XHTML (o HTML) no puede ir dentro del elemento `body`?

    a) `meta`.

    b) `img`.

    c) `h1`.

30. Tenemos un documento XML que es válido de acuerdo con una determinada DTD. Borramos un elemento completo (por ejemplo `<element>text</element>`), que no es el elemento raíz del documento. ¿Cuál de estas tres situaciones no es posible?

    a) Que el documento XML resultante no sea válido respecto de la DTD.

    b) Que el documento XML sea válido respecto de la DTD.

    c) Que el documento XML no sea un documento XML bien formado.

31. En HTML, ¿podemos poner un enlace `<a href="...">...</a>` dentro de un elemento de lista `<li>...</li>`?

    a) Sí.

    b) No, porque el documento estaría mal formado.

    c) No, porque el documento no sería válido.

32. Un fichero de texto escrito en inglés contiene sólo caracteres ASCII. Lo abrimos con un editor y lo guardamos en formato Unicode UTF-8. Ahora ocupa ...

    a) ... el doble de espacio.

    b) ... exactamente el mismo espacio.

    c) ... la mitad de espacio.

33. Señala el fragmento de HTML que generará el texto más grande:

    a) `<h1>Título</h1>`

    b) `<h2>Título</h2>`

    c) `<h3>Título</h3>`

34. ¿Cómo se llama el juego de caracteres estándar universal, el que asigna un número de código diferente y único a cada uno de los caracteres de cada una de las lenguas del mundo?

    a) Unicode.

    b) ISO-8859-1 (*Latin-1*).

    c) XML.

35. ¿Cuándo es preferible utilizar el juego de caracteres Unicode en lugar del ISO-8859-1 (*Latin-1*)?

    a) Cuando vamos a mezclar texto en diferentes idiomas.

    b) Cuando un texto en español tiene muchos acentos.

    c) Cuando el texto sólo se usará en una situación de asimilación.

36. Podemos almacenar correctamente el texto en español "*La España de charanga y pandereta, cerrado y sacristía, devota de Frascuelo y de María, de espíritu burlón y de alma quita*" usando el juego de caracteres ASCII?

    a) Sí, sin problemas.

    b) Sí, si antes lo hemos convertido a UTF-8.

    c) No.

37. ¿Cuánto ocupa un fichero de texto que contiene 2000 caracteres pertenecientes al alfabeto español?

    a) Si se ha codificado en ISO-8859-1 (*Latin-1*), entre 2000 y 4000 bytes.

    b) Si se ha codificado en ASCII, 2000 bytes.

    c) Si se ha codificado en UTF-8, entre 2000 y 4000 bytes.

38. Si al visualizar el documento de texto `noticia.txt` en el navegador aparecen palabras como `camiÃ³n` e `InformÃ¡tica`, ¿cuál puede ser la causa?

    a) Un error por parte de la persona que ha escrito el documento de texto.

    b) Que el navegador está usando para leer el documento un juego de caracteres diferente del que se usó al escribirlo.

    c) Que el documento no es un documento HTML.

4.9. Soluciones

1. (c)
2. (c)
3. (c)
4. (a)
5. (c)
6. (c)
7. (c)
8. (a)
9. (a)
10. (a)
11. (b)
12. (c)
13. (a)
14. (a)
15. (b)
16. (b)
17. (b)
18. (a)
19. (a)
20. (c)
21. (b)
22. (c)
23. (a)
24. (b)
25. (b)
26. (a)

27. (a)
28. (b)
29. (a)
30. (c)
31. (a)
32. (b)
33. (a)
34. (a)
35. (a)
36. (c)
37. (c)
38. (b)

Capítulo 5

Bases de datos

Los gestores de terminología son unos de los programas más usados por los traductores para la traducción humana asistida por una máquina (MAHT; véase el apartado 6.4). Como que se trata de un caso especial de aquello que se denomina en informática *bases de datos*, en este capítulo introduciremos primero este concepto y después estudiaremos la aplicación a la gestión terminológica a través de las bases de datos terminológicas.

5.1. ¿Qué es una base de datos?

Como se explica en la pág. 12, un *fichero* es un conjunto de datos que se guardan en un medio de almacenamiento secundario, que se manipulan como un todo y que se identifican por un nombre. Muchos de los ficheros que usan las personas que se dedican a la traducción son ficheros (o *documentos*) de texto en varios formatos (como los descritos en el epígrafe 4.1), pero también los hay que se corresponden con el significado de la palabra *fichero* fuera de la informática: contienen *fichas*, todas con un formato más o menos constante; por ejemplo, todas las fichas de un fichero bibliográfico contienen información sobre los autores, el título, el año de publicación, etc.

En informática, los ficheros de este tipo se suelen denominar normalmente *bases de datos*; las fichas se denominan *registros* y cada elemento de información de la ficha se denomina *campo*.

Más generalmente, una base de datos se puede ver (en el llamado modelo *plano* o de *tablas*) como un conjunto de tablas en las cuales las columnas o *campos* guardan valores del mismo tipo y donde los elementos de una fila o *registro* están relacionados entre sí. Así, una tabla es un conjunto de registros (fichas) cada uno de los cuales tiene la misma estructura de campos (datos), almacenadas en un fichero informático.

Así, los registros de una base de datos bibliográfica contienen información en campos: uno para los autores, otro para el título, etc. Por otro lado,

los registros de las bases de datos usadas para la gestión de un videoclub contienen campos para almacenar la información referente a los socios (como por ejemplo, el nombre, el teléfono o el domicilio), a las películas (el título, la persona que la ha dirigido o el número de copias disponibles) y a los préstamos (los datos de inicio y plazo del préstamo o el precio de alquiler). Los campos pueden ser de varios *tipos*, según la naturaleza de los datos que se guarden (cadenas de caracteres, valores numéricos enteros o con decimales, fechas del calendario, etc.)

5.2. Operaciones con bases de datos

Las operaciones más comunes que se realizan sobre una base de datos también son similares a aquellas que podemos hacer con un fichero de fichas de cartulina, pero la gestión es más sencilla y son posibles muchos más usos:

- *Creación de la estructura de la base de datos*: definir cada tabla, definiendo la estructura de las fichas y el tipo de datos de cada campo.
- *Altas o adiciones:* añadir un nuevo registro a la base de datos, haciendo que sus campos tomen los valores correspondientes.
- *Bajas o supresiones:* eliminar uno o más registros.
- *Modificaciones:* cambiar el valor de uno o más campos de uno o más registros de la base de datos.
- *Búsquedas o consultas:* buscar uno o más registros que cumplen un determinado criterio de búsqueda. La organización de la información en forma de base de datos simplifica enormemente las consultas, puesto que los ordenadores son mucho más rápidos y seguros a la hora de, por ejemplo, comparar el contenido de un determinado campo de todas las fichas con un cierto valor o patrón (por ejemplo, los autores que empiezan por *Ant*) y listar el contenido de otro campo (por ejemplo, el título) para cada ficha coincidente.

  La combinación de varias estrategias de búsqueda permite la *explotación* de los datos almacenados en una base de datos. Por ejemplo, se puede generar, a partir de un fichero bibliográfico, las referencias bibliográficas citadas en un texto ordenadas alfabéticamente y en el formato requerido por una determinada revista, o generar, a partir de una base de datos de clientes, una carta de recordatorio para los morosos, con detalles sobre sus deudas.

El programa que permite hacer, entre otros, estas operaciones de manera sencilla o incluso automática (aspecto muy importante cuando la base de

datos contiene miles de registros) es un programa *gestor de bases de datos*.[1] Normalmente, los usuarios reales no ejecutan un programa gestor de bases de datos universal o genérico, sino que usan programas o *aplicaciones* que simplifican la creación, el mantenimiento y el uso de la base de datos para un perfil de usuario concreto; también pueden hacer uso de programas que internamente incluyen un gestor de bases de datos o que lo usan. En particular, es posible organizar las bases de datos de forma que estén instaladas en uno o más servidores y se puedan consultar y explotar desde otro ordenador a través de Internet.

5.2.1. Búsquedas

Cuando queremos buscar una determinada información en un fichero de fichas de cartulina (por ejemplo, qué autores han usado la palabra *árbol* en el título de sus obras), y este fichero no está ordenado de acuerdo con ningún criterio conveniente, nos vemos obligados a mirar todas las fichas una a una. Pero es común que los ficheros estén ordenados según uno de sus campos: por ejemplo, un fichero bibliográfico puede estar ordenado por el apellido del primer autor, o por la materia. Si la consulta o la búsqueda que queremos hacer se refiere al campo por el cual se ha establecido la ordenación, ésta es mucho más sencilla que si se refiere a otro campo, y se puede completar sin mirar todas las fichas; por ejemplo, haciendo una *búsqueda dicotómica*.

En una búsqueda dicotómica miramos la ficha que hay en medio del fichero; si nos hemos pasado, repetimos la operación con la primera mitad del fichero, y si nos hemos quedado cortos, lo hacemos con la segunda mitad. Se puede demostrar que la búsqueda dicotómica consulta como mucho n fichas si el fichero tiene entre 2^{n-1} y 2^n fichas, porque tras cada consulta se reduce a la mitad el tamaño del fichero que hay que explorar. Por ejemplo, si el fichero tiene 1234 fichas, hay suficiente con $n = 11$ consultas porque $2^{10} = 1024$ y $2^{11} = 2048$.

Por supuesto, es posible calcular el número máximo de consultas de manera más pedestre, dividiendo el número de fichas por 2 y poniéndonos en el caso peor. En el ejemplo de las 1234 fichas:

- Después de la 1ª consulta, si la ficha central no es la que buscamos, nos quedan dos mitades: una de 616 fichas, y otra de 617. Imaginad que vamos al peor caso: 617 fichas.
- Después de la 2ª consulta, si la ficha central no es la que buscamos, nos quedan dos mitades de 308 fichas.
- 3ª consulta: o la encontramos, o tenemos que buscar en 154 fichas.

[1] A veces se denomina por metonimia *base de datos* al programa gestor.

- 4ª consulta: o la encontramos, o tenemos que buscar en 77 fichas.
- 5ª consulta: o la encontramos, o tenemos que buscar en 38 fichas.
- 6ª consulta: o la encontramos, o tenemos que buscar en 19 fichas.
- 7ª consulta: o la encontramos, o tenemos que buscar en 9 fichas.
- 8ª consulta: o la encontramos, o tenemos que buscar en 4 fichas.
- 9ª consulta: o la encontramos, o tenemos que buscar en 2 fichas.
- 10ª consulta: o la encontramos, o tenemos que buscar en 1 ficha.
- 11ª consulta: o es la que buscamos, o no existe.

Total, 11 consultas como mucho.

Otro ejemplo: la tabla 5.1 ilustra gráficamente el proceso de búsqueda dicotómica sobre una lista de apellidos ordenados alfabéticamente. Para cada paso de la búsqueda, se muestra una tabla donde el elemento consultado en cada momento está en negritas y la parte de la lista descartada está sombreada. Imaginad, en primer lugar, que queremos buscar el elemento "Garrido". Inicialmente, miramos el elemento de en medio (el 13) de la lista entera (la lista que incluye los elementos del 1 al 25) y encontramos "Larrañaga"; como nos hemos pasado, nos quedamos con la mitad baja de la lista (que incluye los elementos del 1 al 12) olvidándonos de la otra mitad. Ahora repetimos el proceso y miramos el elemento de en medio (el 6) de la nueva sublista y encontramos el elemento "Esteve"; como es menor alfabéticamente que el elemento que estamos buscando, nos quedamos con la mitad alta (que incluye los elementos del 7 al 12) de la sublista actual. Volvemos a repetir el proceso, esta vez considerando sólo la sublista de elementos entre el 7 y el 12; tenemos suerte y al mirar el elemento central (el 9) nos damos cuenta de que coincide con el elemento buscado: la búsqueda acaba pues con éxito. Si la búsqueda hubiera sido del elemento "González", los pasos anteriores habrían sido los mismos, pero además habríamos mirado el elemento 11, el elemento 10 y, finalmente, habríamos concluido que el elemento no se encuentra en la lista. En este caso, el número de consultas habría sido 5 y, por tanto, se cumple la afirmación anterior: la búsqueda dicotómica consulta como mucho n fichas si el fichero tiene entre 2^{n-1} y 2^n fichas; aquí el número de elementos total (25) está entre $2^4 = 16$ y $2^5 = 32$, y el número de elementos consultados ha sido $n = 5$.

La búsqueda dicotómica es sólo una de las muchas técnicas que se pueden usar para acelerar las consultas que se refieren a un campo ordenado; el programa gestor de bases de datos puede usar otra técnica de búsqueda dependiendo de su diseño o del tipo de campo.

Pero un fichero de fichas de cartulina sólo se puede ordenar siguiendo un único criterio. Si queremos facilitar las consultas asociadas a más de un

(a) Estado antes de empezar la búsqueda.

1 Beléndez	2 Canals	3 Carrasco	4 Escolano	5 Espí
6 Esteve	7 Forcada	8 Garcia	9 Garrido	10 Gómez
11 Guardiola	12 Iturraspe	13 Larrañaga	14 Marco	15 Micó
16 Montserrat	17 Muñoz	18 Nogueroles	19 Odriozola	20 Oncina
21 Ortiz	22 Pastor	23 Pérez	24 Sempere	25 Zubizarreta

(b) Primer paso: miramos el elemento 13 y nos quedamos con la mitad baja de la lista.

1 Beléndez	2 Canals	3 Carrasco	4 Escolano	5 Espí
6 Esteve	7 Forcada	8 Garcia	9 Garrido	10 Gómez
11 Guardiola	12 Iturraspe	**13 Larrañaga**	14 Marco	15 Micó
16 Montserrat	17 Muñoz	18 Nogueroles	19 Odriozola	20 Oncina
21 Ortiz	22 Pastor	23 Pérez	24 Sempere	25 Zubizarreta

(c) Segundo paso: miramos el elemento 6 y nos quedamos con la mitad alta de la sublista anterior.

1 Beléndez	2 Canals	3 Carrasco	4 Escolano	5 Espí
6 Esteve	7 Forcada	8 Garcia	9 Garrido	10 Gómez
11 Guardiola	12 Iturraspe	13 Larrañaga	14 Marco	15 Micó
16 Montserrat	17 Muñoz	18 Nogueroles	19 Odriozola	20 Oncina
21 Ortiz	22 Pastor	23 Pérez	24 Sempere	25 Zubizarreta

(d) Tercero y último paso: miramos el elemento 9 y encontramos el elemento buscado.

1 Beléndez	2 Canals	3 Carrasco	4 Escolano	5 Espí
6 Esteve	7 Forcada	8 Garcia	**9 Garrido**	10 Gómez
11 Guardiola	12 Iturraspe	13 Larrañaga	14 Marco	15 Micó
16 Montserrat	17 Muñoz	18 Nogueroles	19 Odriozola	20 Oncina
21 Ortiz	22 Pastor	23 Pérez	24 Sempere	25 Zubizarreta

Tabla 5.1: Ejemplo de búsqueda dicotómica sobre una lista de apellidos ordenada alfabéticamente. El elemento a buscar es “Garrido”. Los elementos sombreados han sido descartado durante la búsqueda, es decir, sabemos que el elemento buscado no está entre ellos.

campo (por ejemplo, autores y materias) nos veremos obligados a mantener dos copias del fichero entero, cada copia ordenada por un criterio; con un sistema de bases de datos no hay que hacer esta duplicación: sólo hay que marcar los campos por los cuales buscaremos más frecuentemente (los cuales a veces se suelen denominar *índices*), y el programa gestor de bases de datos *indexará* la base de datos para permitir búsquedas rápidas por estos campos.

Para saber más sobre la indexación de bases de datos

Si las consultas más frecuentes de una base de datos se refieren a un campo —o a una combinación de campos, como por ejemplo el día, el mes y el año que forman una fecha— que toma valores que se pueden ordenar, los registros se pueden ordenar por este campo, igual que el fichero de fichas de cartulina. Pero una de las ventajas más claras de las bases de datos es que permiten que los registros estén ordenados por más de un campo, sin tener que duplicar la base de datos a pesar de que la duplicación de una base de datos pequeña puede no parecer en principio problemática, las cosas cambian si consideramos una base de datos con miles de registros y con un gran número de campos, todos llenos de información. Es evidente, por tanto, que hay que establecer un sistema alternativo para poder hacer búsquedas rápidas por más de un campo. Esto se consigue mediante un procedimiento denominado *indexación*: básicamente, se asigna un número a cada ficha y se construye una tabla o *índice* ordenado (otra base de datos) que contiene registros con dos campos: uno, el campo por el cual se quiere ordenar, y el otro, la posición en la base de datos del registro que contiene este valor del campo (en cierto sentido, este índice no es demasiado diferente del índice alfabético que hay al final de algunos libros: buscamos la palabra alfabéticamente y nos dice en qué páginas se encuentra).

Se puede construir un índice para cada uno de los campos asociados a las consultas más frecuentes y así se evita recorrer toda la base de datos cada vez que se hace una consulta: se busca el valor del campo en el registro correspondiente y, cuando se encuentra, se usa la posición del registro para acceder directamente. Una base de datos con estas propiedades está *indexada*. Los índices se tienen que rehacer parcialmente cuando se hacen altas, bajas o modificaciones de registros para que las consultas continúen siendo eficientes. Cuando creamos una nueva base de datos y definimos la estructura de campos que tendrá cada uno de sus registros, podemos designar qué campos corresponden a los índices; el gestor de la base de datos creará automáticamente los índices correspondientes.

Considerad la base de datos con 10 registros de la tabla siguiente:

NÚM	VASCO	SERBO-CROATA	CATALÁN
1	bat	jedin	un
2	bi	dva	dos
3	hiru	tri	tres
4	lau	četiri	quatre
5	bost	pet	cinc
6	sei	šest	sis
7	zazpi	sedam	set
8	zortzi	osam	vuit
9	bederatzi	devet	nou
10	hamar	deset	deu

Como podéis ver, la base de datos contiene 10 registros con 4 campos; los registros están ordenados por el campo "NÚM", que indica la posición de cada registro. Si queremos hacer consultas rápidas por los campos "VASCO" y "SERBO-CROATA" sin tener que visitar (en el peor caso) todos los registros, el gestor de bases de datos tiene que definir un índice para cada uno de estos campos como los de las tablas siguientes, que muestran los índices correspondientes a los campos "SERBO-CROATA" (izquierda) y "VASCO" (derecha) de la base de datos de la tabla anterior:

SERBO-CROATA	NÚM
četiri	4
deset	10
devet	9
dva	2
jedin	1
osam	8
pet	5
sedam	7
šest	6
tri	3

VASCO	NÚM
bat	1
bederatzi	9
bi	2
bost	5
hamar	10
hiru	3
lau	4
sei	6
zazpi	7
zortzi	8

Podéis comprobar como cada ficha del índice contiene sólo el campo indexado y una referencia a su posición en la base de datos. Si pensáis en el supuesto de que la base de datos de la tabla anterior tuviera, digamos, 20 campos más (con los equivalentes en 20 lenguas más), os podéis hacer una idea del ahorro que se consigue respecto a la duplicación entera.

5.3. Bases de datos léxicas o terminológicas

Uno de los programas más comúnmente usados por los profesionales de la traducción son los gestores de bases de datos léxicas (normalmente llamados gestores de bases de datos *terminológicas*, aunque se pueden usar para otras muchas aplicaciones además de las estrictamente terminológicas). Los profesionales de la traducción gestionan, con estos gestores, bases de datos léxicas que les ayudan a traducir consistentemente los términos y, quizás, las locuciones y frases de una determinada área de conocimiento (terminología).

Los registros o las fichas de una base de datos léxica multilingüe es-

tablecen correspondencias entre los términos usados en varias lenguas en un campo determinado del conocimiento. En estas bases de datos, *cada ficha representa un concepto* y contiene un campo índice para almacenar el término correspondiente en cada lengua. Esta organización (una ficha por concepto) es coherente con la definición de terminología como la disciplina cuyo objeto es el estudio sistemático del etiquetado o designación de *conceptos* particulares de una o más áreas temáticas o de uno o más ámbitos de la actividad humana con el propósito de documentar y promover el uso correcto;[2] además, es especialmente adecuada cuando la base de datos terminológica es multilingüe.

Las bases de datos léxicas o terminológicas pueden contener muchos tipos de campos:

- El término en cada una de las lenguas (campos que normalmente se usan de índice para hacer las búsquedas más eficientes).
- El sentido (entre los posibles sentidos del término) al cual se refiere la ficha o registro actual.
- El autor de la ficha (cuando más de una persona gestiona la base de datos).
- La fecha de creación y de modificación de la ficha.
- La definición del término en una o más lenguas. La mayor parte de los gestores permiten que en los términos usados en la definición de un determinado término se marquen *remisiones*, es decir, enlaces activos a las fichas donde se definen.
- El campo temático de la ficha.
- Otros términos relacionados.
- Información sobre la morfología o la flexión del término en cada una de las lenguas.
- Variantes ortográficas o geográficas; sinónimos; antónimos; etimología, etc.

Una base de datos de este tipo la puede consultar una persona mientras está haciendo una traducción manualmente o puede estar incluida dentro de un programa de traducción automática o asistida por ordenador. Por ejemplo, muchos programas de traducción asistida por ordenador (véase el capítulo 10) incluyen bases de datos terminológicas de este tipo y permiten que la persona usuaria las mantenga y las consulte, bien usando un programa independiente, o bien desde el procesador de textos que prefiera.

[2] `https://es.wikipedia.org/wiki/Terminologa`

También hay bases de datos terminológicas que se pueden consultar en linea:

- El Institut d'Estudis Catalans mantiene el TERMCAT (`http://www.termcat.cat`). La intención de TERMCAT es principalmente normativa: se asocia a cada concepto el término preferido en catalán (y también los términos usuales en español, inglés, francés, etc.).
- IATE (*Inter-Active Terminology for Europe*, `http://iate.europa.eu/`) es la base de datos terminológica de referencia de la Unión Europea y proporciona términos para cada concepto en las 24 lenguas oficiales de la Unión Europea y en latín. La base de datos, en formato TBX (véase el apartado 5.3.1) se puede descargar totalmente o parcialmente para usarla fuera de línea.

5.3.1. El intercambio de bases de datos terminológicas

La creación y el mantenimiento de una buena base de datos terminológica requiere de un gran esfuerzo y muchas horas de trabajo. Esto, la convierte en un recurso valioso y a menudo los traductores se las intercambian. Pero la información almacenada sobre cada término puede ser muy diferente de una base de datos a otra y, por tanto, usar una base de datos terminológica en diferentes sistemas a la vez no es una tarea fácil. Afortunadamente, en los últimos años se ha desarrollado un conjunto de formatos estándares para facilitar este intercambio; uno de los más conocidos se TBX[3], (por *TermBase eXchange*, "intercambio de bases de datos terminológicas"), aunque hay otras como OLIF (*Open Lexicon Interchange Format*). Con el tiempo, los programas que incluyen una base de datos terminológica van incorporando la capacidad de leer y escribir documentos en formato TBX.

Para saber más sobre el formato TBX

El formato TBX sigue las especificaciones XML (véase el apartado 4.4); es decir, los documentos TBX son un tipo de documento XML definido por una DTD concreta. Además, también sigue las directrices del estándar ISO 12620, que define un conjunto de campos, y sus posibles valores, para la información terminológica.

He aquí un ejemplo de documento TBX:

```
<?xml version='1.0'?>
<!DOCTYPE martif SYSTEM  "./TBXcoreStructureDTD-v-1-0.DTD">
<martif type='TBX' xml:lang='en' >
 <martifHeader>
  <fileDesc>
   <sourceDesc>
    <p>from an Oracle corporation termBase</p>
```

[3]El URI para más información es `http://www.lisa.org/tbx/`.

```
    </sourceDesc>
   </fileDesc>
   <encodingDesc>
    <p type='DCSName'>TBXdefaultXCS-v-1-0.XML</p>
   </encodingDesc>
  </martifHeader>
  <text>
   <body>
     <termEntry id='eid-Oracle-67'>
       <descrip type='subjectField'>
         manufacturing
       </descrip>
       <descrip type='definition'>
         A value between 0 and 1 used in ...
       </descrip>
       <langSet xml:lang='en'>
        <tig>
         <term tid='tid-Oracle-67-en1'>
           alpha smoothing factor
         </term>
         <termNote type='termType'>
           fullForm
         </termNote>
        </tig>
       </langSet>
       <langSet xml:lang='hu'>
         <tig>
          <term tid='tid-Oracle-67-hu1'>
            Alfa sim&#x00ED;t&#x00E1;si t&#x00E9;nyez&#x0151;
          </term>
         </tig>
       </langSet>
     </termEntry>
   </body>
  </text>
</martif>
```

La información de cada término (en el ejemplo solo uno) se incluye dentro del elemento `termEntry`, que contiene descripciones (`descrip`) sobre el dominio de uso y la definición del término, además de una sección `langSet` para cada idioma (aquí, inglés y húngaro) donde se especifica el término (`term`) y su información (`termNote`). Antes del cuerpo (`body`), el elemento raíz `martif` contiene una cabecera (`martifHeader`) con información sobre el origen de la base de datos. El término húngaro es "Alfa simítási tényező"; en el documento de ejemplo, los caracteres especiales se indican con su código Unicode (por ejemplo, `&#x0151;` es el carácter "ő")

5.4. Cuestiones y ejercicios

1. Tenemos una base de datos con fichas de 100 alumnos ordenada por el NIF. Si buscamos una alumna por el NIF, ¿cuántas consultas hace, como máximo, el programa gestor de la base de datos hasta llegar a

la ficha deseada?

a) 100.

b) La mitad, 50.

c) 7, porque $2^6 = 64 < 100 < 2^7 = 128$.

2. Queremos tener una base de datos ordenada simultáneamente por dos campos para optimizar las búsquedas. ¿Es esto posible? ¿Cómo?

a) No es posible.

b) Sí, pero es necesario duplicar todas las fichas en memoria.

c) Sí. Se tienen que construir dos índices, uno para cada campo.

3. ¿Es necesario tener dos copias de todas las fichas de una base de datos para poderla tener ordenada por más de un campo?

a) Depende; si los campos son numéricos no es necesario.

b) No hay ningún otro remedio: hay que duplicar la base de datos.

c) No; se puede crear más de un *índice*.

4. Cuando usamos un programa gestor de bases de datos terminológicas para buscar un término en una base de datos...

a) ... algunos gestores nos permiten buscarlo sin conocer la forma exacta.

b) ... tenemos que conocer como se escribe exactamente el término para poder encontrarlo.

c) ...siempre es necesario conocer su categoría léxica e indicársela al gestor

5. Si buscamos en una base de datos de 240 fichas de alumnos una ficha (un registro) por el número de teléfono, un campo por el cual no está ordenada, ¿cuántas consultas tendrá que hacer, como máximo, el programa gestor de la base de datos hasta llegar a la ficha deseada?

a) 240

b) 120

c) 9

6. Cuando queremos tener los registros (las fichas) de una base de datos ordenada simultáneamente por dos campos para optimizar las búsquedas construimos dos índices. Los índices contienen...

a) Entradas compuestas por el valor del campo índice y el número de la ficha (del registro).

b) Sólo los números de las fichas ordenados según el valor del campo.

c) Una copia de las fichas ordenadas por el campo índice correspondiente.

7. ¿Qué tienen en común todos los campos de una ficha terminológica?

   a) Se refieren al mismo concepto.

   b) Se refieren al mismo término.

   c) Se encuentran en el mismo índice.

8. ¿En qué se diferencia un campo clave o campo índice del resto de los campos de una ficha?...

   a) Se guarda en un tipo de memoria RAM más rápida llamada *caché*.

   b) La manera de almacenar el campo es diferente (los campos índice o clave se almacenan de manera comprimida y los otros no).

   c) Las búsquedas de fichas por este campo son mucho más rápidas que las que se hacen por campos que no son clave o índice.

9. Cuando se duplica el número de fichas de una tabla determinada de una base de datos, ¿qué sucede con el número de consultas que realiza una búsqueda dicotómica?

   a) Se duplica.

   b) Se queda exactamente como está.

   c) Se incrementa de media, en 1.

10. En una base de datos léxica o terminológica con 2.000 fichas, cuando pedimos al programa gestor que busque un determinado término en una determinada lengua, ¿cuántas consultas tiene que hacer en el peor caso para entregarnos la ficha?

    a) Tiene que hacer, necesariamente, 2.000 consultas porque hay 2.000 fichas.

    b) No tiene que hacer ninguna consulta, va directamente a la ficha.

    c) Depende. Si está indexada por el término en aquella lengua, hará como mucho 11 consultas, y si no, como mucho 2.000 consultas.

11. En una base de datos terminológica, cada concepto ...

    a) ...se corresponde con un campo de un registro general.

b) ...se corresponde con múltiples registros, uno por cada uno de los términos usados en cada lengua para representar este concepto.

c) ...se corresponde con un registro en el cual se encuentran los términos usados en cada lengua para representar este concepto.

12. Hemos cronometrado el tiempo que tarda un programa gestor de bases de datos en encontrar la ficha que tiene un determinado valor por un campo determinado, cuando aumenta el número de fichas. Los resultados son:

NÚMERO DE FICHAS	TIEMPO
1.000.000	4,9 s
2.000.000	5,1 s
3.000.000	5,3 s
4.000.000	5,4 s
6.000.000	5,5 s

¿Qué podemos decir de la base de datos?

a) Que no está ordenada por el campo por el cual estamos buscando.

b) Que usa XML para obtener una velocidad aceptable.

c) Que está ordenada por el campo por el cual estamos buscando.

13. El programa gestor de una base de datos léxica hacía en el peor caso 480 consultas antes de encontrar la ficha correspondiente a un término inglés concreto, antes de ordenarla por el término en inglés. Después de ordenarla hace

a) 240, la mitad.

b) muchas menos consultas: 9.

c) 480 consultas igualmente.

5.5. Soluciones

1. (c)
2. (c)
3. (c)
4. (a)
5. (a)

6. (a)
7. (a)
8. (c)
9. (c)
10. (c)
11. (c)
12. (c)
13. (b)

Capítulo 6

La traducción automática y sus aplicaciones

Una de las aplicaciones más importantes de la informática a la traducción es la *traducción automática* (TA). Pero antes de considerar la automatización de la traducción y sus aplicaciones sería bueno que reflexionáramos un poco acerca del significado exacto de la palabra *traducción*. Más adelante, en la sección 9.3, discutiremos sobre la relación entre traducción humana y traducción automática.

6.1. ¿Qué es la traducción?

Para empezar, se tiene que tener en cuenta que la palabra *traducción* es ambigua[1] porque se puede referir al *proceso* de traducir o al *producto* (resultado) de este proceso.

Sager (1993)[2] empieza su definición diciendo que, como proceso, se puede denominar traducción a "un rango de actividades humanas deliberadas, que se hacen como resultado de instrucciones recibidas de un tercero, y que consisten en la producción de textos en una lengua meta (LM), basada, entre otras cosas, en la modificación de un texto en una lengua origen (LO) para hacerlo adecuado a un propósito nuevo", pero todavía no describe la naturaleza de la modificación.

Como producto, una *traducción* se puede identificar como tal porque es un documento (en LM) derivado de otro documento en otro idioma (LO), y que mantiene una cierta similitud de contenido con éste.

Se pueden decir todavía más cosas sobre la traducción:

- las traducciones suelen estar escritas en un sublenguaje particular (registro, especialidad, etc.) de la comunidad lingüística de la LM, basa-

[1] Como otros muchos sustantivos acabados en *-ción*.

[2] Los conceptos de esta sección están tomados casi íntegramente de esta obra.

do en un sublenguaje paralelo de la LO;

- los documentos y las traducciones correspondientes se pueden clasificar en tipos[3] y esta tipología afecta a la traducción;
- la traducción se ve afectada por elementos extralingüísticos porque, normalmente, los documentos son entidades que unen la expresión lingüística con la no lingüística;
- las traducciones tienen un *receptor* o *lector*; una traducción, como acto comunicativo, debe considerar, además de la intención de la traducción, las expectativas de los lectores, que resultan de su trasfondo cultural y de sus necesidades comunicativas, y que influyen en la recepción del texto traducido;
- la traducción siempre tiene una motivación: la superación de barreras comunicativas; por esto, se ha creado una profesión.

Se puede profundizar un poco más en la definición de traducción que hemos considerado más arriba, revisando definiciones existentes (algunas tomadas de Sager 1993):

- Nida (1966, p. 19): "La traducción consiste en producir en la LM el equivalente natural más cercano del mensaje en la LO, primero en cuanto al significado y después en cuanto al estilo." Sager dice que, más que *natural* (en sentido absoluto) habría que decir *adecuado* (a la tarea concreta). Esta definición introduce dos de las tres dimensiones básicas de un documento escrito (original o traducido): el *contenido* (significado) y la *forma* (estilo), pero olvida el *propósito*.
- Flamand (1983): "traducir es representar con precisión (fidelidad al autor) un mensaje en LO en una forma auténtica y correcta de la LM, adaptada al contenido y al receptor (fidelidad al lector)". El problema de esta definición es la indefinición del concepto de *fidelidad*.
- Jakobson (1966): "Traducción es la interpretación de signos verbales por medio de otra lengua". Esta definición evita el concepto de *equivalencia* e introduce el de *interpretación* como conjunto de procesos cognitivos que tienen lugar en la mente del traductor.
- En el *Diccionari de la Llengua Catalana*[4] se define *traducción* como la "reproducción del contenido de un texto o de un enunciado oral, formulado en una lengua, en formas propias de otra lengua" (y *traducir*

[3] Por ejemplo, *carta comercial*, *edicto municipal*, *comentario editorial*, *manual técnico informático* o compilación *de poemas*.

[4] Editorial Enciclopèdia Catalana, 7ª ed., 1987.

como "escribir o decir en una lengua aquello que ha sido escrito o dicho en otra"). La definición incluye, por lo tanto, el tratamiento y la producción de mensajes no textuales (orales).

- Alcaraz Varó y Martinez Linares (1997) definen traducción como "expresión de un enunciado en la lengua de llegada [lengua meta] que sea equivalente al de la lengua de partida [lengua origen]"; queda por definir la noción de *equivalencia*, que los mismos autores definen así: "la posesión del mismo valor por parte de los enunciados de la lengua de partida y de la de llegada"; la equivalencia puede ser *semántica*, *estilística* y *textual*.

Para acabar este apartado, conviene mencionar algunos procesos a los que no llamaremos *traducción* en el contexto de este libro:

- la adaptación de textos antiguos a la forma moderna de un idioma;
- la traducción de palabras y frases cuando se enseña un idioma nuevo;
- la interpretación (de mensajes hablados);
- la codificación (en Morse, etc.).

6.2. Traducción automática

La traducción automática (TA) se puede definir como el proceso (o el producto) de traducir un texto informatizado[5] en una lengua origen a un texto informatizado en una lengua meta mediante el uso de un programa de ordenador. Normalmente se reserva la denominación *traducción automática* para la completamente automática; cuando se produce intervención humana se habla de *traducción asistida por ordenador* o de *traducción semiautomática*. El resultado de la traducción automática es normalmente un producto bastante diferente del de la traducción profesional, y en la mayoría de los casos no se puede usar en su lugar tal y como está; por eso, este capítulo está dedicado a analizar las diversas modalidades de interacción entre personas y máquinas en traducción.

Es necesario hacer una aclaración sobre el tratamiento de los textos informatizados. Cuando los programas de traducción automática y semiautomática tienen que tratar documentos estructurados (como los discutidos en los epígrafes 4.4 y 4.5.1) tienen que ser capaces de identificar las partes de los documentos que corresponden a los textos que se tienen que traducir, separándolas de las etiquetas de formato. Normalmente, los programas tienen un módulo inicial que podríamos denominar *desformateador* y

[5] Llamaremos *texto informatizado* a un fichero que contiene un texto codificado en un formato conocido (véase el capítulo 4), y que puede ser editado con un editor o con un procesador de textos adecuado.

un módulo final que podríamos denominar *reformateador* y que restituye las etiquetas de formato de manera que el formato y la estructura del documento se conserven tanto como sea posible. En general, estas operaciones se pueden considerar básicamente independientes del proceso de traducción —como haremos en este libro—, pero hay programas más avanzados que son incluso capaces de usar la información de las etiquetas de formato como contexto para elegir una traducción cuando hay más de una alternativa.

Las referencias que se han hecho en el epígrafe 6.1 a propósito o motivación de la traducción y a la tipología de los documentos que tienen que ser traducidos son también muy importantes a la hora de analizar la traducción automática.

Sobre el nombre en otras lenguas. En inglés, la traducción automática se denomina *machine translation* y se abrevia MT, paralelamente al alemán, que usa la denominación *maschinelle Übersetzung*; en estas dos lenguas se expresa la noción de automatismo mediante la referencia a una *máquina*. En cambio, en francés se habla, como en catalán o en español, de *traduction automatique*. Otras lenguas, como por ejemplo el neerlandés, usan una palabra compuesta con la palabra *ordenador*: *Computervertaling*.

Para saber más sobre la historia de la traducción automática

La mayor parte de lo discutido en este apartado se ha extraído de los trabajos de John Hutchins, especialmente de Hutchins (1995) y Hutchins (2001). El Dr. John Hutchins es considerado el historiador de la traducción automática, y hasta hace poco mantenía activamente el archivo `www.mt-archive.info`, en donde se pueden encontrar reproducciones facsímiles de muchos artículos de los inicios de esta disciplina.

Los pioneros, hasta 1954: La traducción mediante máquinas es una ambición humana desde hace siglos que no se hizo realidad hasta el siglo XX. No hacía mucho que se había creado el primer ordenador, cuando ya se empezó a pensar en la posibilidad de usarlos para traducir lenguajes humanos.

A pesar de que en el decenio que va de 1930 a 1940 hubo algunos trabajos precursores, es a principios de los años cincuenta cuando empieza realmente la investigación en TA en muchas universidades de todo el mundo, especialmente en los Estados Unidos de América. Los recursos de hardware, software y lenguajes de programación eran demasiado reducidos y la primera aproximación fue la traducción palabra por palabra basada en diccionario con algunas reglas sencillas de reordenamiento (a veces erróneamente llamada *traducción directa*) y similar a los sistemas de traducción indirecta por transferencia morfológica avanzada (véase el apartado 8.3.1). Esta carencia de recursos hizo que los primeros objetivos fueran muy modestos y, así, los primeros investigadores se concentraron en el desarrollo de lenguajes controlados (véase el apartado 6.5.3) y en la ayuda humana en tareas de preedición y postedición (véase el apartado 6.5); estaba bastante claro que los sistemas reales no podrían producir más que traducciones de muy baja calidad. En 1952 se celebró en los Estados Unidos el

primer congreso sobre TA donde se definieron las líneas fundamentales a seguir.

El decenio del optimismo, 1954–1966: La primera demostración pública de un sistema de TA fue desarrollada por IBM y la Universidad Georgetown en el año 1954. Se tradujo al inglés un conjunto de 49 frases en ruso usando un diccionario de nada más que 250 palabras y 6 reglas gramaticales; estas lenguas fueron elegidas por razones geopolíticas para los primeros sistemas de TA. A pesar de que los resultados no eran demasiados buenos, el público y la industria creyeron que en unos años se podrían conseguir traducciones automáticas de calidad de documentos científicos y técnicos. Esta idea cogió fuerza gracias a que empezaron a aparecer mejoras significativas en el hardware, los primeros lenguajes de programación y muchas mejoras en la lingüística formal (especialmente en el área de la sintaxis). El entusiasmo hizo que se financiaran un gran número de proyectos entre la mitad de la década de los 50 y la mitad de la década de los 60, proyectos dentro de los que nació la mayor parte de las técnicas actuales, como la traducción indirecta por transferencia o la traducción por interlingua (véase el capítulo 8).

El objetivo era el desarrollo de sistemas perfectos. Había que reducir al mínimo la intervención humana en el proceso de TA, hasta lograr la independencia total y una calidad comparable a la de los humanos. Prácticamente nadie consideró cómo se podría sacar provecho a un sistema imperfecto (con excepciones contadas; Masterman (1967) estudió la utilidad de la traducción *palabra por palabra* como *pidgin*, es decir, como lengua de contacto, en comparación con una traducción *nativa*): ¿Para qué pensar en ello si pronto se dispondría de sistemas perfectos? Los traductores se sintieron amenazados. Sin embargo, algunas voces se pronunciaron en contra del perfeccionismo dominante y defendieron una aproximación al problema a más a largo plazo y la construcción de sistemas que hicieran un uso efectivo de la interacción persona-máquina.

Un decenio después, los avances eran escasos y el futuro próximo no parecía poder mejorar la situación. Muchos investigadores empezaban a encontrar barreras de todo tipo, especialmente semánticas, que parecían demasiado difíciles de superar y que exigían métodos más complejos. La Academia Nacional de las Ciencias de los Estados Unidos publicó en 1966 el informe ALPAC (Automatic Language Processing Advisory Committee) en el cual se recomendaba que los numerosos recursos que se dedicaban a la investigación en TA se utilizaran para tareas menos ambiciosas y más básicas relacionadas con el procesamiento del lenguaje natural y con el desarrollo de herramientas de apoyo para los traductores, como por ejemplo diccionarios automáticos. La conclusión era que sólo después de conocer las raíces del problema, podría estudiarse la construcción de un sistema de TA real. El informe aseguraba que la TA era más lenta y menos exacta que la realizada por humanos, además de ser el doble de cara, y que no había ningún indicio de la obtención en el futuro más o menos inmediato de un sistema de TA útil. El informe hizo que se redujera significativamente el número de personas que se dedicaban a la TA y que los laboratorios empezaran a trabajar en lo que se conoció como lingüística computacional.

Desde el informe ALPAC (1966) hasta los ochenta: El informe acabó casi virtualmente con la investigación en TA en los Estados Unidos (también tuvo un impacto negativo en los proyectos desarrollados en el resto del mundo) y durante muchos años la TA fue vista como un auténtico fracaso. Aún así, algunos grupos continuaron trabajando en Canadá y en Europa y aparecieron los primeros sistemas que funcionaban; en 1970 el sistema Systran empezó a ser usado por la USAF (United States Air Force) y en 1976 por la Comisión de la Comunidad Europea. También en 1976 aparece Metéo, desarrollado por la Universidad de Montréal, que traduce al francés

informes meteorológicos. Por esta época, además, los sistemas de TA empiezan a ser demandados por empresas y administraciones y no sólo para traducir textos científicos y técnicos.

Desde el informe ALPAC el campo sufrió una redefinición progresiva hacia una concepción de la TA como un proceso en el cual los traductores humanos juegan un papel básico, y empiezan a desarrollarse herramientas de traducción pensando en esta intervención.

Las principales corrientes dentro de la TA desde los años 70 son, por lo tanto: herramientas de apoyo a la traducción para traductores, sistemas de TA con intervención humana e investigación teórica hacia un sistema completamente automático de traducción.

Principios de los ochenta: En la década de 1980 aparecen nuevos sistemas de TA en todo el mundo con expectativas más reales y el interés en la TA resurge. Son especialmente importantes los resultados obtenidos en varias empresas como Xerox, donde se elimina casi por completo la postedición (véase la pag. 122) gracias al control de la lengua origen (véase el apartado 6.5.3); esto permite la traducción sencilla de los manuales técnicos en inglés de la compañía a un gran número de idiomas (francés, alemán, italiano, español, portugués y lenguas escandinavas).

Durante esta década los esfuerzos se centran en la traducción indirecta, con representaciones intermedias o sin ellas (como la interlingua; véase el apartado 8.4), mediante análisis morfológicos y sintácticos y, a veces, conocimiento semántico. Los proyectos más notables son GETA-Ariane (Grenoble), SUSY (Saarbrücken), Mu (Kyoto), DLT (Utrecht), Rosetta (Eindhoven), el proyecto de la Universidad Carnegie-Mellon (Pittsburgh) y dos proyectos internacionales: Eurotra, financiado por la Comunidad Europea, y el proyecto japonés CICC, con participantes en China, Indonesia y Tailandia.

Eurotra es uno de los proyectos de traducción más conocidos del decenio de 1980. Su objetivo era la construcción de un sistema de transferencia multilingüe que permitiera la traducción entre todas las lenguas de la Comunidad Europea. A pesar de que se esperaba que la traducción resultante fuera de gran calidad, todavía se preveía una gran cantidad de postedición. El proyecto, que se abandonó en el año 1992, no fue capaz de entregar un sistema de TA funcional, pero estimuló la investigación sobre tecnologías lingüísticas en toda Europa.

En estos años se consolida la idea de que los sistemas de TA no son para traductores; un traductor necesita herramientas que le faciliten el trabajo: diccionarios, bases de datos terminológicas (véase el capítulo 5), sistemas de comunicación, memorias de traducción (véase el capítulo 10), etc. De hecho, actualmente, la postedición no se encarga siempre a traductores (muchos de los cuales no consideran esto como parte de su trabajo), sino a personas que se presume que tiene formación específica en postedición.

Todo el mundo acepta ya en este decenio la importancia de los lenguajes controlados y los sublenguajes en la TA, como ya habían defendido los precursores de la TA durante el decenio de los cincuenta.

El sistema comercial más sofisticado de los 80 es Metal (1988), financiado por Siemens y que traduce del alemán al inglés. Se trata básicamente de un sistema por transferencia, indicado para la traducción de documentos relacionados con el procesamiento de datos y las telecomunicaciones.

A finales de la década de 1980 empieza la aplicación de técnicas de inteligencia artificial al procesamiento del lenguaje humano (sistemas expertos y sistemas basados en conocimiento diseñados para entender los textos).

Principios de los noventa: Todos los sistemas de TA de los ochenta, tanto los de transferencia como los de interlingua, funcionan básicamente a partir de reglas lingüísticas. Pero en la década de 1990 aparecen nuevas estrategias conocidas como métodos basados en corpus. Los métodos basados en corpus se pueden dividir en dos grupos: estadísticos y basados en ejemplos.

Los métodos estadísticos ya fueron considerados en los años sesenta, pero pronto fueron descartados porque los resultados obtenidos no eran aceptables. Pero el descubrimiento de nuevas técnicas hizo posible proyectos como Candide, de IBM. Candide usa métodos estadísticos para el análisis y la generación, y ninguna regla lingüística. Los trabajos en IBM utilizaron el corpus de textos en inglés y francés resultantes de las sesiones del Parlamento de Canadá. El método consiste en alinear en primer lugar las frases, los grupos de palabras y las palabras en los dos textos y calcular después la probabilidad de que una palabra del texto origen se corresponda con una o más palabras del texto meta con las cuales ha sido alineada.

Los métodos basados en ejemplos (véase el capítulo 10) se aprovechan también de la existencia de grandes corpus de textos traducidos (por eso también se les llama basados en memoria). La idea fundamental es que el proceso de traducción se puede hacer a menudo consultando traducciones anteriores e identificando frases o grupos de palabras en el corpus ya traducido. Para poder llevar a cabo la traducción es necesario que los textos del corpus hayan sido alineados previamente (mediante métodos estadísticos o métodos basados en reglas).

A pesar de que la gran innovación de los noventa fueron los métodos descritos antes, la investigación y el desarrollo de los sistemas clásicos también continuó: por ejemplo, el proyecto Eurolang, basado en el sistema de transferencia Metal, permitía traducir del inglés al francés, alemán, italiano y español, y viceversa. En los últimos 15 años, uno de los campos con más investigaciones ha sido el de traducción del habla, una idea que evidentemente ha estado presente desde hace décadas, pero que sólo ahora se puede materializar parcialmente. El objetivo no es obtener un sistema de traducción perfecta, sino un sistema adecuado para aplicaciones con lenguajes, dominios y usuarios restringidos. Los principales son los desarrollados en ATR, CMU y el proyecto Verbmobil.

Una característica importante de los inicios de la década de 1990 es la aparición de las primeras aplicaciones prácticas para traductores: herramientas de apoyo a la traducción, diccionarios y bases de datos terminológicas, procesadores de texto multilingües, acceso a glosarios y terminologías electrónicas, herramientas de comunicación (escáners, OCRs, Internet; véanse los capítulos 3 y 4) o herramientas para entornos restringidos. La combinación de algunas de estas herramientas en un software concreto es lo que se conoce como *estaciones de trabajo para traductores*; por ejemplo, el Translation Manager de IBM, recientemente liberado como software libre/de código fuente abierto con el nombre OpenTM2, `http://www.opentm2.org`, o el Translator Workbench de Trados, ahora llamado SDL Trados Studio. La mayor parte de estas estaciones de trabajo están disponibles para ordenadores personales.

Desde finales de los noventa hasta la actualidad: La TA y las herramientas de apoyo a la traducción son cada vez más usadas por las grandes empresas y por las administraciones, principalmente para la traducción de documentación técnica.

A lo largo de los últimos años, con la generalización del uso de Internet, se han desarrollado servicios de traducción disponibles en linea, como por ejemplo, Google Translate (`http://translate.google.com`) o Bing Translator (`http://translator.bing.com`), de uso muy común por parte del público en general para la *asimilación* (véase el apartado 6.3.1) de contenidos web escritos en otras lenguas e incluso para la traducción de carteles y textos fotografiados con la cámara del teléfono móvil.

Desde sus inicios, casi toda la investigación y casi todos los sistemas comerciales de TA se han centrado en los principales idiomas internacionales: inglés, francés, español, japonés, ruso, etc. Todavía queda mucho por hacer con las otras lenguas del mundo; con excepciones como el proyecto Apertium (`http://www.apertium.org`), que ofrece traducción automática para lenguas menos centrales, como por ejemplo el gallego, el occitano y el bretón.

En el momento de escribir estas líneas (diciembre de 2015), la mayor parte de los sistemas de traducción automática se basan en una evolución de la *traducción automática estadística* iniciada en IBM durante la década de 1990; se suele hablar de *traducción automática estadística basada en segmentos*, en inglés *phrase-based statistical machine translation*. Esta hegemonía se debe, en gran parte, a la disponibilidad de software libre/de código fuente abierto para *entrenar* y aplicar sistemas de traducción automática, como por ejemplo Moses (`http://statmt.org/moses`). Incluso hay empresas como KantanMT (`http://kantanmt.com`) que construyen sistemas a medida para sus clientes usando simplemente un navegador.

En los últimos años se está investigando una nueva modalidad de traducción basada en el llamado *aprendizaje profundo*, en inglés *deep learning*, que usa métodos de un campo de la inteligencia artificial denominado *redes neuronales*, y los resultados empiezan a ser, en pruebas de laboratorio, comparables a los mejores disponibles.

6.3. Utilidad de la traducción automática

La traducción automática produce resultados que normalmente no pueden sustituir directamente los producidos por profesionales de la traducción (véase el capítulo 9). Por ejemplo, en muchos casos es difícil conseguir que el ordenador sepa elegir la interpretación correcta entre las posibles interpretaciones de un enunciado ambiguo como

Los soldados dispararon a los niños. Los vi caer.

dado que esto requiere el uso de cantidades enormes de conocimiento enciclopédico sobre el funcionamiento del "mundo real". En este caso, el sistema tiene que saber: que los disparos hieren gravemente o matan a las personas que los reciben y que la condición de herido grave o muerto es incompatible con mantenerse derecho, y que, por todo esto, la interpretación más probable es que cayeron los niños, no los soldados.

Muchas de las aplicaciones de la traducción automática se pueden dividir en dos grandes grupos: la *asimilación* de información (cuando una persona usa la traducción automática para obtener información a partir de un documento escrito en otra lengua) y la *diseminación* —también llamada *difusión*— de información (cuando una persona usa la traducción automática para producir documentos que tienen que ser distribuidos a más de un usuario). La traducción automática, a pesar de ser muy diferente de las traducciones hechas por profesionales competentes, puede ser una herramienta muy útil en estos dos grupos de aplicaciones.

6.3.1. Asimilación

En situaciones de *asimilación* de información no parece necesaria una traducción gramaticalmente correcta y similar al texto que produciría una persona nativa, sino más bien una traducción rápida y razonablemente inteligible. Se debe tener en cuenta que hay características de los textos nativos que pueden no ser necesarias para la comprensión. Por ejemplo, un texto puede ser inteligible aunque no concuerden los adjetivos con los nombres o incluso aunque se hayan eliminado los artículos (*A amigo mio le gustan chicas mayores*), o el orden de las palabras no sea gramatical (*La guerra evitar no podremos*).[6]

Una de las primeras aplicaciones de la traducción automática en los EE.UU. fue el llamado *screening* o exploración de documentos para decidir cuáles eran relevantes y merecían una atención más detallada: se quería tener acceso a la información tecnológica presente en documentos de la Unión Soviética. Los usos civiles del *screening* han superado actualmente el uso tradicional, el militar. En el caso del *screening*, incluso una traducción incompleta además de incorrecta (por ejemplo, sólo de las palabras terminológicas) puede ser de gran utilidad. Otros ejemplos de uso de la traducción automática para la *asimilación* son:

- La traducción automática del correo electrónico entre las personas de un grupo de trabajo internacional con la finalidad de agilizar las comunicaciones.
- La traducción inmediata de documentos durante la *navegación* por Internet (de hecho, hay programas diseñados especialmente para esta finalidad, como *Google Translate* o *Bing Translator*).
- La traducción automática de *conversaciones electrónicas* interactivas (usando el teclado y la pantalla de ordenadores conectados entre sí; *chat*) entre personas que hablan dos idiomas diferentes. Las limitaciones de la traducción automática se pueden compensar con preguntas o diciendo las cosas de otro modo hasta que los dos interlocutores se entiendan (es decir, mediante una *negociación*).
- La traducción de despachos de prensa en otros idiomas.

Es importante indicar que en casi todas las situaciones de asimilación el papel del traductor profesional es inexistente, dado que el trabajo es de naturaleza muy diferente, y el uso de un traductor profesional sería muy caro y muy lento.

Por rudimentario que sea un sistema de traducción automática, puede ser muy útil en tareas de asimilación. Una de las aproximaciones más

[6] De hecho, se podrían diseñar los sistemas de traducción automática para que no se *preocuparan* por estos asuntos menores.

simples a la TA es la llamada *traducción palabra por palabra*, en la que el programa identifica cada palabra, la busca en un diccionario bilingüe y la sustituye por una traducción aproximada (véase también la pág.154. A modo de ejemplo, considerad el siguiente texto en tok pisin[7] (el texto está tomado de Lyovin 1997):

> *Long taim bifo, wanpela ailan, draipela pik i save stap ya, na em i save kaikai ol man. Em i save kaikai ol man nau; wanpela taim, wanpela taim nau ol man go tokim bikpela man bilong ol, bos bilong ol, ol i go tokim em nau, em i tok: "Orait yumi mas painim nupela ailan".*

Si cogemos un diccionario y traducimos el texto palabra por palabra, tomando la primera traducción posible en cada caso —puede haber más de una—, se obtiene el texto siguiente:[8]

> *En tiempo pasado, un isla, enorme cerdo - soler vivir mencionado y él - soler comer más-de-un hombre. Él - soler comer más-de-un hombre entonces; un tiempo, un tiempo entonces, más-de-un hombre ir hablar gran hombre en más-de-un, ninguno en más-de-un, más-de-un - ir hablar él entonces, él - decir: "Muy-bien, vosotros-y-yo haber-de encontrar nueva isla".*

Y ahora, ¿verdad que se entiende un poquito más? Una traducción más idiomática podría ser:

> *Hace mucho tiempo, en una cierta isla, vivía un gran cerdo y se solía comer a la gente. Se solía comer a la gente, y una vez, la gente fue y dijo a su gran hombre, a su jefe, fueron y hablaron con él. Él dijo: "Muy bien, tenemos que encontrar una nueva isla".*

El orden de las palabras no es muy diferente en tok pisin y en español y esto hace que la traducción palabra por palabra sea basta leíble. En cambio, si el texto original está en vasco, las cosas no son tan sencillas. El texto, prácticamente ininteligible para quien no sepa vasco:

> *Baazkaria bukatu ondoren Koldo egunkarira joan zen eta Teoren foto bat hartu zuen. Gero, egunkariaren ale zaharrak irakurri zituen, boxeo txapelketako berriak aztertzeko. Boxealarien izenak apuntatu zituen.*

[7] Lengua de contacto que se habla en Papúa Nueva Guinea y que tiene 50.000 hablantes que la hablan como primera lengua y más de dos millones de hablantes que la hablan como segunda lengua.

[8] Es posible que ya os hayáis dado cuenta de que el tok pisin tiene mucho vocabulario tomado del inglés, como lengua de contacto que es.

se puede traducir palabra por palabra cómo:

> *La-comida acabada después Koldo al-diario ido era y de-Teo foto una tomado lo había. Después, del-diario número los-viejos leído los-había, boxeo del-campeonato las-noticias para-a-examinar. De los-boxeadores los nombres apuntado los-había.*

que es mucho más difícil de leer que el resultado de traducir el texto en tok pisin palabra por palabra. Una traducción idiomática posible es:

> *Después de comer Koldo fue al diario y tomó una foto de Teo. Después, leyó [los] números viejos del diario para examinar las noticias del campeonato de boxeo. Apuntó los nombres de los boxeadores.*

Fijaos que incluso en este caso tan desfavorable el texto traducido palabra por palabra da bastantes pistas sobre el significado del texto original.

En los últimos años, especialmente desde que se ha generalizado el acceso público en Internet, se observa una tendencia a incorporar sistemas de traducción automática como uno de los componentes de sistemas más grandes de comunicación. Esta aplicación de la TA para la asimilación se puede ver *en chats* bilingües, o en los sistemas que traducen las páginas *web* según vamos visitándolas siguiendo enlaces; en estos sistemas, la TA no se invoca explícitamente, sino implícitamente cuando usamos el servicio.

6.3.2. Diseminación

En situaciones de *diseminación* de la información hay que revisar el borrador de traducción producido por el traductor automático y hacer las modificaciones oportunas para convertirla en una traducción adecuada al propósito de las traducciones. Para minimizar las modificaciones a hacer en la traducción automática, puede ser útil restringir la lengua de origen (no permitir todas las realizaciones posibles, ni todo el léxico, ni todos los registros) a un lenguaje que pueda ser traducido automáticamente con el mínimo posible de problemas, es decir, con el mínimo esfuerzo de postedición, o al menos, con un esfuerzo aceptable para un revisor.[9] Esto es especialmente importante cuando se trata de traducir manuales técnicos a varios idiomas. Las restricciones se pueden expresar bajo la forma de mensajes interactivos dirigidos a la persona que prepara el documento original.[10]

La traducción automática para la diseminación es especialmente eficiente cuando sólo se traducen textos pertenecientes a una parte muy reducida y muy regulada del idioma en cuestión (un *sublenguaje*). Un ejemplo es Météo, el sistema que desde 1982 hasta 2001 producía en Canadá informes meteorológicos simultáneos en francés e inglés.

[9] Es decir, cuando la revisión no es más costosa que rehacer toda la traducción a mano.

[10] Véase el apartado 6.5.3, donde se discute un concepto muy relacionado, el de *lenguaje controlado*.

6.4. Traducción semiautomática

Muchas situaciones de traducción automática se pueden clasificar como situaciones de traducción asistida por ordenador (en inglés *computer-aided translation*; CAT), también llamada a veces *traducción semiautomática*. El término *computer-aided translation* se usa normalmente para referirse al entorno de software que permite la traducción profesional con el apoyo de bases de datos léxicas (véase el epígrafe 5.3), de las sugerencias de traducción provenientes de memorias de traducción (véase el capítulo 10), e incluso, de la traducción automática.

Para precisar mejor qué queremos decir con esto de "asistida por ordenador", es necesario considerar las nociones de traducción humana asistida por una máquina (en inglés *machine-aided human translation*; MAHT), y traducción automática asistida por un humano (en inglés *human-aided machine translation*; HAMT), que establecen las dos situaciones básicas de interacción entre una persona y un ordenador a la hora de hacer la traducción. Los párrafos siguientes dan algunos ejemplos.

MAHT: El usuario (un traductor competente o un profesional independiente) utiliza diccionarios bilingües, tesauros o *thesaurus*, conjugadores y declinadores, correctores ortográficos, sintácticos y de estilo, y formularios o modelos de documentos, como ayuda mientras produce una traducción de manera manual usando un procesador de textos. Otras herramientas —de uso común entre varios traductores, y accesibles normalmente como recursos remotos— pueden ser las bases de datos terminológicas y las bases de datos léxicas multilingües (véase el epígrafe 5.3), o las memorias de traducción (véase el capítulo 10).

HAMT: Un programa de traducción automática pregunta al usuario cuando tiene más de una posible traducción para una palabra o para una frase. Esta y otras situaciones de *negociación* del texto original con el usuario del sistema implican una interacción que también puede ayudar a preparar un texto más correcto, es decir, a *preeditarlo* (véase el apartado 6.5) para que pueda ser traducido automáticamente. En otras ocasiones, el programa puede analizar la estructura profunda de la frase y presentar las posibles interpretaciones al autor, para que resuelva alguna posible ambigüedad. En estos sistemas interactivos hay que tener en cuenta dos factores: el primero, que un sistema que pregunta demasiado no es cómodo de usar (no es *ergonómico*); y el segundo, que puede suceder que el usuario sea monolingüe, circunstancia que cambia mucho la naturaleza de la interacción entre el programa y el usuario. Los usuarios de este tipo de sistemas se podrían clasificar en tres grandes grupos: traductores ocasionales, traductores profesionales individuales y traductores profesionales que trabajan

para empresas de traducción.

6.5. Automatización del proceso de traducción

A la hora de abordar la automatización del proceso de traducción hay que hacer un análisis de los costes de traducción para estimar el ahorro en recursos (como por ejemplo tiempos y dinero) que se producirá con la introducción de la traducción automática. El capítulo 9 se centra en la evaluación de los sistemas de traducción automática y el análisis de costes de traducción; en este apartado discutiremos las diferentes tareas y opciones para automatizar el proceso de traducción.

6.5.1. Postedición

La *postedición* es la modificación *mínima* de una traducción generada por ordenador para *hacerla adecuada a un* ***propósito*** *bien definido*: el texto meta producido por el sistema se refina o revisa (*postedita*) para que sea gramaticalmente correcto o esté escrito de acuerdo con un registro determinado.

A la hora de posteditar tenemos que evitar hacer cambios *preferenciales* (esta solución adecuada "me gusta más" que esta otra que también es adecuada). Los cambios estilísticos se tienen que hacer estrictamente cuando, de no hacerse, la traducción resultante no cumpliría con el propósito para el cual fue encargada. Las modificaciones pueden ser: *borrados* de una palabra que sobra, *sustituciones* de una palabra por otra, o *inserciones* de una palabra que falta. Tienen que ser las *mínimas* necesarias: si hay más de una edición posible, hay que elegir la que se haga con el mínimo de modificaciones necesarias.

Hay que tener en cuenta que la persona posteditora, además de conocer la lengua meta y ser capaz de convertir el texto en bruto a una forma genuina en esta lengua (es decir, además de ser profesional de la traducción), debe ser una verdadera especialista en postedición, conocedora del sistema de traducción automática y de sus errores más típicos. Así, la tarea de postedición es mucho más eficiente, puesto que al conocer el origen y la causa de los errores se hace más fácil y rápida la corrección.

En una primera aproximación, la postedición será conveniente cuando

$$\textbf{coste}\left(\begin{array}{c}\text{traducción automática}\\ + \\ \text{postedición}\end{array}\right) < \textbf{coste}(\text{traducción profesional}).$$

Hay que comprobar que la fórmula anterior se cumple, aunque sea a largo plazo, antes de elegir una estrategia de traducción para la diseminación basada en la postedición.[11]

[11] Para un análisis de costes más detallado, véase el apartado 9.2.1.

6.5.2. Preedición

La *preedición* consiste en preparar o adaptar (*preeditar*) el texto origen para facilitar su traducción y mejorar el comportamiento del sistema de traducción automática, reduciendo la necesidad de postedición de la traducción en bruto. Esto se consigue, por ejemplo, eliminando la ambigüedad del texto,[12] evitando el uso de la voz pasiva, reduciendo el uso de oraciones subordinadas o usando frases cortas y completas sintáctica y semánticamente.[13] La preedición del texto origen se puede hacer también para marcar partes del texto que no tienen que ser traducidas, como por ejemplo una cita, o que tienen que ser tratadas de manera especial por no ser frases completas, como un título.

La preedición suele ser tanto más conveniente cuanto mayor es el número de lenguas a las que se traduce el texto preeditado porque un cambio en el texto origen puede ahorrar tantas postediciones como lenguas de llegada tengamos.

En resumen, hay tres modalidades básicas de interacción entre las personas y los programas de traducción automática:

- la preedición (preparación del texto *antes* de la traducción automática),
- la postedición (corrección del texto *después* de la traducción automática) y
- la interacción de la persona con el sistema de traducción automática durante el proceso de traducción.

6.5.3. Lenguajes controlados

Cuando la traducción automática se usa para la diseminación de documentos técnicos de temática homogénea, puede ser interesante hacer que los documentos originales estén escritos usando un léxico estándar sin ambigüedades semánticas y siguiendo unas reglas sintácticas y de estilo bien determinadas, es decir, en un *lenguaje controlado* (Wojcik y Hoard 1996; Arnold et al. 1994; O'Brien 2003) diseñado de forma que el resultado de la traducción automática pueda ser usado directamente para publicarlo con el mínimo posible de postedición.

Un *lenguaje controlado* es "un subconjunto del lenguaje natural definido con precisión, por un lado restringido en cuanto al léxico, la gramática y el

[12] Por ejemplo, en inglés técnico, la palabra *replace* presenta una *ambigüedad léxica* (véase el apartado 7.2.1), puesto que puede querer decir *exchange* (reemplazar) o *put back* (volver a colocar).

[13] Kohl (2008) ofrece indicaciones para escribir textos en inglés para una audiencia global, de forma que los textos sean más fáciles de entender para los no nativos y más fáciles de traducir manualmente y automáticamente.

estilo, y por otro, posiblemente extendido con terminología y construcciones gramaticales específicas de un dominio" (Huijsen 1998).

Un lenguaje controlado tiene a menudo asociado un conjunto de programas de apoyo que ayudan a evaluar y escribir documentos que cumplan las restricciones. Quien escribe en un lenguaje controlado normalmente usa un editor de textos inteligente que realiza las siguientes tareas:

- Comprobar el cumplimiento de las restricciones:
  - terminológicas (como el caso de la palabra *replace* mencionado más arriba); para eso, puede ser útil acceder a una base de datos terminológica (como las mencionadas en el capítulo 5);
  - sintácticas, por ejemplo, haciendo el análisis sintáctico de las oraciones y detectando las ambigüedades estructurales (véase el apartado 7.2.2), y
  - de estilo, por ejemplo, especificando cuál debe ser el formato de las fechas o de las horas.
- Emitir un mensaje de error, lo más informativo posible, cuando se detecte una violación de las especificaciones del lenguaje.
- Proponer a la persona usuaria formas alternativas válidas al texto erróneo.

Como puede apreciarse, los desarrollos técnicos hechos alrededor del diseño de un lenguaje controlado se relacionan con muchos conceptos que se tratan en este libro.

Un ejemplo histórico de lenguaje controlado que fue utilizado para mejorar los resultados de la traducción automática —concretamente, los obtenidos con un sistema también histórico llamado Weidner MicroCat— es PACE (*Perkins Approved Clear English*), el lenguaje controlado usado durante los años ochenta y parte de los noventa por la empresa de ingeniería Perkins Engines para facilitar la traducción automática de los manuales que describen las características y el mantenimiento de estos motores (Newton 1992; Douglas y Hurst 1996). Uno de los principios de PACE es "una palabra, un significado", es decir, se establecen restricciones léxicas claras a través de un diccionario, cosa que simplifica el diseño de los diccionarios del sistema de traducción automática. Además del léxico, PACE también especifica la sintaxis (Arnold et al. 1994, secció 8.3). Otros ejemplos de lenguajes controlados son el *ScaniaSwedish* usado por la firma de camiones y autobuses Scania (Almqvist y Sågvall Hein 1996), o el *Caterpillar Technical English* de la compañía de maquinaria de excavación Caterpillar.

También hay lenguajes controlados que no están específicamente diseñados para la traducción automática, como el inglés simplificado (*Simplified English*) del AECMA (Asociación Europea de Industrias Aeroespaciales), que se caracteriza por "una sintaxis sencilla, un número limitado de

palabras, un número limitado de significados bien definidos por palabra (normalmente uno), y un número limitado de categorías léxicas[14] por palabra (normalmente una)", con "el objetivo de producir textos breves y no ambiguos"(AECMA 2007).

Algunas de las ventajas del uso de lenguajes controlados (Schwitten 2007) se pueden resumir como sigue:

- los textos son más sencillos e inteligibles;
- el mantenimiento de los documentos es más fácil;
- se simplifica el tratamiento computacional de los documentos, en particular la traducción automática.

En cuanto a las desventajas, podemos decir que:

- el diseño de un lenguaje controlado no es nada trivial: hay que estudiar en profundidad corpus de textos pertenecientes al dominio y tomar decisiones difíciles;
- el poder de expresión de un lenguaje controlado es siempre más restringido;
- la escritura de textos en un lenguaje controlado es más lenta;
- es necesaria una inversión adicional de tiempo en el aprendizaje del lenguaje controlado por parte de los autores.

Las dos últimas desventajas se pueden reducir si se dota a los autores de herramientas informáticas, como por ejemplo un editor de textos inteligente que les ayude a escribir en el lenguaje controlado.

Por último, hay que dejar claro que el uso de un lenguaje controlado es una alternativa a la preedición de los textos, pero que no elimina por completo la necesidad de postedición o, al menos, de revisión de las traducciones en bruto.

6.6. Cuestiones y ejercicios

1. (*) Elegid un idioma cualquiera que conozcáis bien, L. Seguramente L tenga palabras polisémicas que en otra lengua L' tienen más de una traducción, según el sentido que se tome. Elegid tres palabras

[14] Las *categorías léxicas* (o simplemente *categorías*) son conjuntos de palabras que tienen la misma función sintáctica; hay categorías *mayores*, *léxicas* o *de clase abierta* (substantivo, adjetivo, verbo, etc.) que crecen cuando se añade léxico nuevo a la lengua y categorías *menores*, *gramaticales* o *de clase cerrada* (artículos, conjunciones, etc.), que no crecen y contienen palabras con función gramatical. La sintaxis se define normalmente, no en términos de palabras, sino en términos de categorías léxicas.

de L que tengan este problema y describid cómo las trataríais en un lenguaje controlado basado en L. Las reglas que formuléis para los autores que escriban en el lenguaje controlado tienen que estar escritas en L y no tienen que contener referencias a otras lenguas.

2. En los sistemas de traducción automática, la preedición...

   a) ... reduce la cantidad de postedición.

   b) ... es una alternativa a la postedición, que elimina completamente esta última fase.

   c) ... imposibilita el uso del sistema para tareas de diseminación de información.

3. Indica en cuál de estas situaciones de traducción automática son menos cruciales la gramaticalidad o naturalidad lingüística de la traducción.

   a) Joan usa Web Translator mientras navega por las páginas de Internet de la Universität Mainz para saber qué asignatura da el profesor Karl-Hans Lehninger y cuáles son sus intereses investigadores.

   b) Joan usa Web Translator para hacer una versión en alemán de su página Web.

   c) El personal de IBM traduce patentes europeas para detectar posibles avances en corrección de errores de comunicaciones digitales.

4. Imaginad que podemos elegir entre dos sistemas de traducción automática diferente t_A y t_B para traducir manuales de televisores del inglés al francés, y que se tiene que diseñar un inglés controlado para minimizar la postedición. Las reglas del inglés controlado, ¿pueden depender del sistema de TA elegido?

   a) No, porque los lenguajes controlados se tienen que diseñar independientemente de los sistemas de TA.

   b) Sí, porque en cada caso se tienen que evitar problemas diferentes.

   c) No, porque la lengua meta de los dos sistemas es la misma.

5. Indica cuál de estas situaciones de traducción automática es de *asimilación* de información:

   a) Narciso usa el programa traductor del inglés al español Spanish Assistant para leer los documentos electrónicos que encuentra en Internet sobre la influencia del euskera sobre el gascón.

b) Joan usa Web Translator para hacer una versión en alemán de su página Web antes de publicarla en Internet.

c) La empresa Into the Wind traduce automáticamente su catálogo de dos tipos de cometas a varias lenguas.

6. Muchas veces, la preedición la hace el autor cuando interacciona con el programa de traducción automática. ¿Es posible diseñar un sistema de preedición interactiva para autores monolingües?

   a) Sí.

   b) No. Para preeditar correctamente hay que conocer el idioma de destino.

   c) Sólo para ciertos idiomas con estructura gramatical sencilla como el inglés.

7. ¿Cuál de las siguientes *no* es una ventaja de los lenguajes controlados?

   a) Se evita la necesidad de que una persona interaccione con el programa de traducción automática para resolver ambigüedades durante la traducción.

   b) Los textos meta resultantes son mucho más cortos.

   c) Los textos origen se hacen más inteligibles.

8. ¿Por qué es necesaria la preedición en los sistemas de traducción automática?

   a) Para evitar construcciones o frases difíciles de traducir.

   b) Para que el formato quede más agradable a la vista.

   c) Es una alternativa a la postedición.

9. Imaginad que un traductor profesional cobra 0,05 euros por palabra de texto traducido y que un corrector de textos cobra 0,10 euros por palabra de texto corregido. Imaginad que tenemos un sistema de traducción automática que nos cuesta 0,03 euros por palabra traducida y que produce un 10 % de palabras incorrectas en las traducciones. ¿Conviene adoptarlo y contratar al corrector o es mejor contratar al traductor profesional? (si no sabéis calcularlo en general, haced los cálculos con un texto de, por ejemplo, 1000 palabras).

10. La traducción automática instantánea de páginas *web* durante la navegación es un caso de traducción automática...

    a) ... con preedición.

    b) ... para la diseminación.

    c) ... para la asimilación.

11. El "control" de los lenguajes controlados...

    a) ... se refiere tanto a la terminología como a la sintaxis.

    b) ... solo puede referirse a la sintaxis.

    c) ... sólo puede referirse a la terminología.

12. Cuando se usan para la traducción, los lenguajes controlados restringen directamente...

    a) ... la lengua meta.

    b) ... la lengua origen.

    c) ... tanto la lengua origen como la lengua meta.

13. Si un anglohablante usa el traductor automático–inglés de `babelfish.altavista.com` para leer en línea el diario brasileño *O Globo*, está usando la traducción automática para un propósito...

    a) ...de asimilación de información.

    b) ...de diseminación.

    c) ...para el que no está pensada.

14. ¿Cuál es la alternativa estándar a la preedición en un entorno de producción masiva de documentación multilingüe?

    a) El uso de un lenguaje controlado

    b) El uso de un sistema de interlingua.

    c) La postedición sistemática

15. Fran consulta a través de Internet la base de datos terminológica IATE (véase el apartado 5.3) cuando traduce dosieres antiglobalización del inglés al neerlandés. ¿En cuál de las tres situaciones siguientes se encuentra?

    a) Traducción automática asistida por la persona

    b) Traducción humana asistida por la máquina

    c) Usa un lenguaje controlado

16. Si enviamos un documento HTML a un servidor de traducción automática y después posteditamos el resultado para que sea una traducción aceptable del original antes de publicarla, estamos usando la traducción automática...

    a) ... con memoria de traducción.

    b) ... para la diseminación.

c) … para la asimilación.

17. La adopción de un lenguaje controlado en una situación de traducción de documentos de una lengua a muchas lenguas para la diseminación es, en el proceso completo, una alternativa a la…

    a) …postedición repetitiva de los documentos meta.

    b) …la preedición repetitiva de los documentos origen.

    c) …la traducción de fragmentos ya traducidos anteriormente.

18. Un sistema que sugiere mejoras en el estilo de un documento se puede considerar como …

    a) … HAMT.

    b) … MAHT.

    c) … un sistema de traducción automática ergonómico.

19. Una inventora monolingüe consulta documentos web traducidos a su lengua para descubrir si su nuevo invento ha sido patentado antes. Si la traducción se hace mediante un sistema automático, ¿qué uso está haciendo?

    a) Asimilación; más concretamente para aquello que se llama *screening*.

    b) Diseminación.

    c) Postedición, ya que el idioma del documento cambia para que pueda ser entendido.

20. El programa de la Generalitat Valenciana SALT 4.0 traduce textos del español a la variedad valenciana del catalán y pregunta esporádicamente a la persona usuaria qué equivalente es más adecuado para algunas palabras ambiguas difíciles. Esta es una situación de…

    a) … postedición.

    b) … traducción automática asistida por la persona.

    c) … traducción humana asistida por el ordenador.

21. Queremos posteditar un texto traducido automáticamente mirando tan poco como sea posible el texto original. ¿Nos ayuda conocer cuáles son las palabras homógrafas (véase la p. 122) más comunes de la lengua origen?

    a) No, porque las palabras homógrafas del texto origen no afectan al texto meta en bruto.

    b) No, porque solo estamos mirando el texto meta.

c) Sí, porque son una fuente muy importante de errores especialmente difíciles de corregir si no se conoce qué ha pasado.

22. Una persona está escribiendo un documento en lengua origen que después será traducido automáticamente a más de una lengua meta y el sistema que usa para escribir le avisa cuando teclea una palabra que dará problemas de traducción —y le sugiere alternativas— o cuando escribe una estructura que será difícil de traducir. Esta es una situación...

a) ...de preedición.

b) ...de aplicación de un lenguaje controlado.

c) ...de postedición.

23. ¿Cuál de las siguientes situaciones es absurda en traducción automática?

a) La postedición en una aplicación de asimilación.

b) La postedición en una aplicación de diseminación.

c) El uso de un lenguaje controlado en una aplicación de diseminación.

24. Solo una de estas tres afirmaciones es correcta. ¿Cuál?

a) Los lenguajes controlados definen reglas de postedición.

b) El uso de un lenguaje controlado elimina completamente la necesidad de postedición.

c) Cuando se aplican las reglas de un lenguaje controlado, el texto resultante es gramaticalmente aceptable, pero se evitan construcciones y palabras que dan problemas.

25. Un sistema de traducción automática hipotético del ruso al español produce texto que es básicamente correcto excepto por el hecho de que no genera ni artículos determinados (*el*, *la*, *los*, *las*) ni indeterminados (*un*, *una*, *unos*, *unas*). ¿Qué diríais de este sistema?

a) Que es especialmente adecuado para la asimilación, pero no tanto para la diseminación porque los artículos son más del 10 % del texto.

b) Que es especialmente adecuado para la diseminación, ya que los artículos son palabras muy poco frecuentes en el texto y por lo tanto, no será necesaria mucha postedición.

c) Que no es útil ni para la asimilación ni para la diseminación.

26. ¿Se puede posteditar sin mirar el texto original?

a) Sí.

b) En general, no. Quien postedita produce una traducción. Por lo tanto, tiene que estar seguro de que el resultado es traducción del texto original.

c) Si el texto es técnico, se puede hacer sin mirar. En otro caso, hay que mirar siempre el texto original.

27. El uso de un lenguaje controlado hace que ...

    a) ... la escritura sea más rápida.

    b) ... el estilo del documento resultante sea más homogéneo.

    c) ... el poder de expresión del idioma sea más grande.

28. Cuando posteditamos un texto encontramos una palabra que el traductor automático no ha sabido traducir y nos la ofrece en la lengua origen. Sin embargo, este error no ha afectado a la traducción del resto de la oración. ¿Qué debemos que hacer?

    a) Preeditar el texto original completo sustituyendo la palabra por un sinónimo que sí reconozca el traductor automático y volver a traducir todo el texto.

    b) Probar a traducir todo el texto con otro traductor automático.

    c) Corregirla y seguir posteditando.

29. Si en una fábrica de frigoríficos se usan sistemas de traducción automática para traducir a otras muchas lenguas los manuales de los numerosos modelos que se fabrican (y que son muy similares entre sí), la solución más eficiente para evitar errores de traducción es...

    a) ... regular la manera en la que los autores escriben los manuales.

    b) ... posteditar todas las traducciones.

    c) ... preeditar los manuales antes de traducirlos.

30. A la hora de preeditar un texto para traducirlo automáticamente conviene ...

    a) ... usar frases cortas.

    b) ... usar la forma pasiva.

    c) ... usar oraciones subordinadas.

31. La postedición de la traducción realizada por un traductor automático es siempre necesaria ...

    a) ... para usarla con finalidades de diseminación.

    b) ... para usarla con finalidades de asimilación.

    c) ... cuando se ha realizado también preedición.

6.7. Soluciones

1. (*) Por ejemplo, si L es el español, palabras como *escondite* pueden referirse a un lugar donde esconderse (1) o a un juego (2) (en L'=catalán, *amagatall* (1) y *fet*, *amagar*, *fet a amagar* o *conillets a amagar* (2)). En el lenguaje controlado, se podría evitar el primer significado proponiendo a los autores que usaran la palabra alternativa *escondrijo*. Las reglas se podrían formular como sigue en español:

   escondite *úsese sólo en el sentido de "juego del escondite"; úsese* escondrijo *si se quiere indicar el lugar donde se esconde alguna persona o cosa.*

   registro *úsese sólo en el sentido de "transcripción", ïnscripción" uno .ºficina de registro"; úsese* inspección *cuando se refiera, por ejemplo, a la investigación detallada de un local por parte de la policía.*

   explotar *úsese sólo en el sentido de .ªprovechar económicamente"; úsese* estallar *en el sentido de "deflagrar" (una bomba, etc.) o "reventar" (un globo, etc.).*

2. (a)

3. (a)

4. (b)

5. (a)

6. (a). Las preguntas se pueden plantear como en el problema 1.

7. (b)

8. (a)

9. **Solución 1 (n palabras):** El traductor profesional traduce un texto de n palabras por 0,05$\times n$ euros. El sistema de traducción automática lo traduce por 0,03$\times n$ y corregirlo cuesta 0,10$\times(10/100)\times n$, es decir, 0,01$\times n$ euros porque sólo 10 de cada 100 palabras son incorrectas. Por lo tanto, el sistema semiautomático cuesta sólo $(0{,}03{+}0{,}01)\times n$ =0,04$\times n$ euros, frente a los 0,05$\times n$ euros del traductor profesional.

   Solución 2 (1000 palabras): El traductor profesional traduce un texto de 1000 palabras por 0,05$\times$1000=50 euros. El sistema de traducción automática lo traduce por 0,03$\times$1000=30 euros, y corregirlo cuesta 10 euros, porque en 1000 palabras hay 1000$\times$10/100=100 palabras incorrectas y corregir cada una cuesta 0,10 euros: 0,10$\times$100=10.

Por lo tanto, el sistema semiautomático cuesta sólo 40 euros, frente a los 50 del traductor profesional.

10. (c)
11. (a)
12. (b)
13. (a)
14. (a)
15. (b)
16. (b)
17. (b)
18. (b)
19. (a)
20. (b)
21. (c)
22. (b)
23. (a)
24. (c)
25. (a)
26. (b)
27. (b)
28. (c)
29. (a)
30. (a)
31. (a)

Capítulo 7

¿Por qué es difícil la traducción automática? Ambigüedad

7.1. Los cuatro problemas de la traducción automática

¿Por qué es tan difícil para un sistema informático traducir como un profesional? Una clasificación interesante de los problemas de la traducción automática la proporciona Arnold (2003). Según este autor, la traducción automática tiene cuatro grandes problemas:

1. *La forma no determina completamente el contenido* (es decir, la *interpretación*): no siempre es fácil determinar la interpretación que se pretendía dar a lo que se ha escrito. Este es el *problema del análisis*, también llamado *ambigüedad*. Ejemplos: *Traían noticias de Grecia* (¿tema o procedencia?), *Ha vendido las naranjas que ha comprado a Juan* (¿Juan vende las naranjas o las compra?), *Trabaja en el estudio que le han encargado* (¿prepara un documento o está diseñando un espacio de trabajo?), etc.

2. *El contenido no determina completamente la forma*. Es decir, es difícil determinar cómo se tiene que expresar una interpretación concreta porque hay más de una manera de decir lo mismo en cualquier lengua. Este es el *problema de la síntesis*. Ejemplos: ¿cómo se dice en qué momento del día nos encontramos? Cada idioma lo hace de una forma distinta: español: *¿Qué hora es?*; portugués: *Que horas são?* (*¿Qué horas son?*); alemán: *Wie spät ist es?* (*¿Cómo de tarde es?*); alemán : *Wie viel Uhr ist es?* (*¿Cuántas del reloj son?*), etc.[1]

3. *Las lenguas divergen.* Es decir, hay diferencias irreducibles en la manera de expresar el mismo contenido en diferentes lenguas. Este es el *problema de la transferencia*, porque se manifiesta habitualmente en los

[1] Véase el ejemplo *me gusta nadar* en la p. 172

sistemas de traducción automática por transferencia (véase el epígrafe 8.3). Por ejemplo, el orden estándar de las oraciones en español es *sujeto–verbo–objeto*, mientras que en vasco o en turco es *sujeto–objeto–verbo*, en irlandés es *verbo–sujeto–objeto* y en malgache es *verbo–objeto–sujeto*. O por ejemplo, los idiomas difieren en la manera en la que expresan las relaciones entre dos nombres: mientras que en español se dice *presidente* ***de*** *Kazajistán*, en ruso se dice *prezident Kazakhstan****a***, en vasco se dice *Kazakstan****go*** *presidente*, y en kazajo se dice *Qazaqstan prezident****i***.

4. Construir un sistema de traducción automática conlleva la gestión de una gran cantidad de conocimiento, que se debe recopilar, y representar en un formato útil para su procesamiento mediante un ordenador. Este es el *problema de la descripción*.

De estos cuatro, dedicaremos el resto del capítulo a describir con más detalle el más importante para la traducción automática (y, en general, para cualquier programa que tenga que procesar textos en lenguaje natural): la ambigüedad inherente al lenguaje humano.

7.2. Ambigüedad

Podemos decir que un enunciado (una oración, un texto) es ambiguo cuando es susceptible de dos o más interpretaciones (Alcaraz Varó y Martinez Linares 1997).[2] Por tanto, un enunciado ambiguo puede tener más de una traducción a otro idioma, aunque a veces puede tener una sola traducción a otro idioma porque dicha traducción conserva la ambigüedad de la frase original;[3] a esto se le suele llamar *free ride* ("pase gratuito") y es más frecuente cuanto más cercanas son las lenguas involucradas en la traducción. En este capítulo nos fijaremos especialmente en la ambigüedad de las oraciones.

Una de las perspectivas más interesantes para analizar y ordenar los tipos de ambigüedad descritos más arriba nos la proporciona el llamado *principio de composicionalidad* (Radford et al. 2009, cap. 23):

> *La interpretación de una oración está determinada por la interpretación de las palabras que aparecen en la oración y por la estructura sintáctica de la oración.*

Este principio explica por qué la interpretación de la oración

[2] Don et al. (1996) lo expresan diciendo que la ambigüedad es "el fenómeno por el cual una expresión tiene más de un significado".

[3] Por ejemplo, la oración en español *Aprendió a afeitarse en dos minutos* se puede traducir al catalán como *Va aprendre a afaitar-se en dos minuts* sin resolver la siguiente ambigüedad: ¿es el tiempo que tardó en aprender o el tiempo que emplea en afeitarse?

(7.1) *El padre escurre los platos*

es diferente de la de la oración

(7.2) *La madre lee libros*

Las oraciones (7.1) i (7.2) tienen la misma sintaxis pero diferente interpretación porque contienen palabras diferentes con interpretaciones diferentes. También explica por qué la oración

(7.3) *El perro mordió al hombre*

no tiene la misma interpretación que la frase

(7.4) *El hombre mordió al perro*

Estas oraciones no quieren decir lo mismo porque, a pesar de tener las mismas palabras, la estructura sintáctica no es la misma.

Por eso, no es posible asignar una interpretación clara a oraciones sintácticamente incorrectas, como

(7.5) **Lee madre libros la*

aunque cada palabra tenga una interpretación independiente, y tampoco a una oración sintácticamente correcta que contenga alguna palabra a la que no podemos asignar una interpretación:

(7.6) *La madre *ingurplee libros*

Como veremos más adelante, existe una complicación adicional: en algunas ocasiones, hay partes de la estructura sintáctica que no se reflejan en ninguna palabra porque generan *categorías vacías* que no tienen una representación fonética o gráfica explícita. Por ejemplo, la oración

(7.7) *Tiene muchos amigos*

tiene un sujeto vacío (también denominado elíptico). En estos casos podemos considerar que las categorías vacías son palabras "de cero letras" que tienen una interpretación.

Si una oración es ambigua cuando tiene más de una interpretación posible, ésta puede tener dos causas básicas:

- una o más palabras de la oración tienen más de una interpretación posible (es decir, son *léxicamente ambiguas*).
- la oración tiene más de una estructura sintáctica posible (es decir, es *estructuralmente ambigua* o *sintácticamente ambigua*).

Las dos causas pueden concurrir. De hecho, estudiaremos tres casos: la ambigüedad debida a la ambigüedad de las palabras (explícitas o nulas); la ambigüedad debida a la existencia de más de una estructura sintáctica, y la ambigüedad debida a ambas causas.

7.2.1. Ambigüedad debida a la ambigüedad léxica

En muchas lenguas las palabras se flexionan y toman formas diferentes. Una palabra (y, en general, una unidad léxica de más de una palabra) se puede ver desde dos perspectivas; por un lado, la *forma superficial* de la palabra es la forma concreta que aparece en el texto: *cantábamos*; por otra parte, tenemos la *forma léxica*, que consiste en

- un *lema* o *forma canónica* (*cantar*),
- una *categoría léxica*,[4] clase de palabra, o parte de la oración (verbo) y
- unos *indicadores de flexión* que expresan las características morfológicas o flexivas (primera persona, nombre plural, tiempo pretérito imperfecto, modo indicativo).

Cuando dos formas léxicas diferentes tienen la misma forma superficial; es decir, cuando se escriben del mismo modo, se suele decir que son *homógrafas*; además, se denomina simplemente *homógrafa* a la forma superficial a la que corresponde más de una forma léxica; el fenómeno se denomina *homografía*. Por ejemplo, la palabra *río* es homógrafa porque tiene dos formes léxicas: *río*, sustantivo, masculino singular y *reír*, verbo, 1ª persona del singular, presente de indicativo. Podemos diferenciar tres tipos de ambigüedad por homografía:

1. *Ambigüedad entre categorías léxicas diferentes*. Por ejemplo, la palabra *ahorro* tiene dos formas léxicas posibles, cada una con una categoría léxica diferente: *ahorro*, sustantivo, masculino, singular; *ahorrar*, verbo, 1ª persona del singular, presente de indicativo.

2. *Ambigüedad dentro de la misma categoría léxica sin cambio de lema*. Por ejemplo, la palabra *canta* tiene dos formas léxicas con el mismo lema, la misma categoría léxica, pero distinta información de flexión: *cantar* verbo, 3ª persona del singular, presente de indicativo; *cantar*, verbo, 2ª persona del singular, imperativo.

3. *Ambigüedad dentro de la misma categoría léxica con cambio de lema*. Por ejemplo, la palabra en catalán *poden* tiene, entre otras, dos formas léxicas con la misma categoría léxica, la misma información de flexión, pero distinto lema: *poder*, verbo, 3ª persona del plural, presente de indicativo; *podar*, verbo, 3ª persona del plural, presente de indicativo.

Pero la homografía no es la única causa posible de ambigüedad léxica; hay palabras que son ambiguas a pesar de tener la misma forma léxica, porque lo que es ambiguo es la interpretación del lema. Estas palabras se denominan habitualmente *polisémicas* y este tipo de ambigüedad, *polisemia*.

[4] Véase la nota al pie de la pág. 110

Por ejemplo, la palabra *estación* (forma léxica: *estación*, sustantivo, femenino singular) es polisémica porque el lema correspondiente tiene más de una interpretación: lugar donde se paran temporalmente los trenes, parte del año comprendida entre un solsticio y un equinoccio, conjunto de instalaciones para un propósito determinado (por ejemplo, el esquí), etc. La polisemia afecta a todas las formas flexionadas de una determinada palabra del mismo modo (*estaciones* tiene exactamente la misma ambigüedad que *estación*), puesto que es una propiedad del *lema*.[5]

La ambigüedad de una oración puede ser causada por varios tipos básicos de ambigüedad léxica:

1. La oración contiene una o más unidades léxicas (por ejemplo palabras) polisémicas: si decimos que alguien

   (7.8) *Trabaja en el estudio que le encargaron*

   podemos referirnos a un investigador o a un decorador, dependiendo de qué interpretación asignemos a la palabra polisémica *estudio*. A la ambigüedad de estas unidades léxicas, también se la suele llamar *ambigüedad léxica pura*. Esta oración la tenemos que desambiguar si la queremos traducir, por ejemplo, al inglés, porque en el primer caso tendríamos que decir *study* y en el segundo, *studio*; por eso el efecto de la polisemia en traducción puede causar en muchos casos la llamada *ambigüedad de transferencia*. La ambigüedad léxica de transferencia es especialmente peligrosa cuando afecta a una palabra de la lengua origen no percibida como ambigua; por ejemplo, la palabra española *destino* se puede traducir al catalán como *destí* (suerte futura) o *destinació* (punto de llegada).

   Otro ejemplo: la oración

   (7.9) *Han puesto un banco nuevo en la plaza*

   puede tener dos interpretaciones, según la interpretación que se asigne a la palabra polisémica *banco* ("asiento estrecho y largo" o bien "institución financiera").

   Una ambigüedad que es muy parecida a la ambigüedad léxica pura se produce cuando una expresión idiomática se toma bien como tal o bien en sentido literal. Por ejemplo, la interpretación de la expresión

[5] Hay casos que no son tan sencillos. Por ejemplo, en inglés, la palabra *case*, un sustantivo singular, puede referirse a un tipo de contenedor (*a case of wine*) o a un ejemplo o situación particular (*It does not apply in this case*). Cada una de las palabras viene de una palabra latina diferente: la primera de la palabra femenina *capsa*, y la segunda de *casus*, el participio de *cado* 'caer'. Los diccionarios ingleses, que suelen agrupar las palabras polisémicas en una entrada, típicamente hacen dos entradas diferentes, y, de hecho, en lexicografía no es extraño referirse a *case* como un homógrafo.

catalana *enviar algú a pastar fang* puede ser la idiomática de decir a alguien que deje de molestar (en español *mandar a freír espárragos*) pero podría ser también la literal en un taller de alfarería.

2. La oración contiene un homógrafo que tiene dos o más interpretaciones pero la misma categoría léxica, y no afecta, por lo tanto, a la estructura de la oración. Hay tres situaciones posibles:

   - cambia sólo el lema pero no los indicadores de flexión: la palabra española *creo* puede ser la 1ª persona del singular del presente de indicativo del verbo *creer* o del verbo *crear*.
   - no cambia el lema pero sí los indicadores de flexión: la palabra española *cantamos* puede ser la 1ª persona del plural del presente de indicativo o del pretérito perfecto del verbo *cantar*.
   - cambian el lema y los indicadores de flexión: la palabra española *salan* puede ser la 3ª persona del plural del presente de indicativo del verbo salir o del presente de subjuntivo del verbo *salar*.

3. La oración contiene una *expresión anafórica*, como por ejemplo un pronombre, adjetivo posesivo, etc., la cual puede tener, en principio, más de una posible interpretación, pero esta interpretación está determinada por la relación de *correferencia* entre la expresión y su *antecedente* (un sintagma que se puede encontrar en la misma oración o en otra oración del texto) o porque se refiere a algún objeto o concepto exterior al texto. La relación que asigna una interpretación a una expresión anafórica se denomina *deixis*: cuando la interpretación es por *correferencia* con un antecedente que aparece anteriormente en el texto se denomina *anáfora*, y *catáfora* si el antecedente es posterior. En la frase

   (7.10) *Abrí [la puerta]$_i$ a [la cocinera]$_j$ y la$_{i/j?}$ hice pasar*

   los índices (i, j, $i/j?$) indican que el pronombre *la* se puede referir a la misma persona a la cual nos hemos referido con el sintagma nominal *la cocinera*, pero no hay ninguna razón sintáctica para que el referente no sea el mismo que el del sintagma *la puerta*: esta puede ser una posible causa de ambigüedad en la segunda oración coordinada.

4. La oración tiene constituyentes que no se reflejan como palabras, pero a los que hay que asignar una interpretación. En algunas lenguas románicas (en italiano, español y catalán, pero no en francés) es común la ausencia del sujeto cuando es de tercera persona. En este caso, la posición donde tendría que ir el sujeto se puede suponer ocupada por un pronombre sin forma superficial que da lugar a ambigüedad mediante mecanismos muy similares a los de la anáfora y, por lo tanto, se los puede considerar mecanismos léxicos. En el fragmento

(7.11) *Anna apuñaló a Marta. Joan vio como caía rodando*

quién cayó rodando, ¿Anna o Marta? ¿O alguna otra persona? El problema es que falta el sujeto de la oración subordinada *como caía rodando*. Esta omisión da lugar a una ambigüedad. Cuando se trata de la omisión del sujeto, se suele postular en lingüística la existencia de un pronombre especial llamado PRO, sin forma superficial, que hace de sujeto nulo,

(7.12) *Joan vio como PRO caía rodando*

y al cual se asigna interpretación mediante procesos deícticos[6] o anafóricos como los descritos para otras expresiones anafóricas.

Esta clase de ambigüedades se suele incluir dentro de un grupo de fenómenos más generales denominados *ambigüedades por elipsis*. Alcaraz Varó y Martinez Linares (1997) definen la *elipsis* como la omisión o la ausencia de alguna parte de una oración. Como veremos más abajo, a veces la elipsis da lugar a la existencia de más de un árbol de análisis sintáctico para la oración y por lo tanto estos tipos de elipsis no se pueden incluir propiamente en este apartado dedicado a la ambigüedad puramente léxica.

7.2.2. Ambigüedad estructural pura

La ambigüedad de una oración también puede ser debida al simple hecho de que tenga más de un árbol de análisis sintáctico. Se pueden distinguir varios casos:

1. *Ambigüedad estructural de origen coordinativo*: Por ejemplo, si decimos

   (7.13) *Pon las sábanas y las colchas limpias en el armario*

   hay dos posibles interpretaciones; en una las sábanas no están limpias, en la otra sí, según se considere que el adjetivo *limpias* modifica a los dos sustantivos coordinados o sólo al último (véase la fig. 7.1). La ambigüedad estructural asociada a las conjunciones coordinativas se suele denominar *de origen coordinativo*.

2. *Ambigüedad estructural de adjunción* (en inglés *attachment ambiguity*): se trata de un caso típico de ambigüedad estructural que se manifiesta cuando hay un *adjunto*[7] (normalmente un sintagma preposicio-

[6] Relacionados con la *deixis*.

[7] Un *adjunto* es un sintagma o constituyente que, conjuntamente con otro sintagma o constituyente, forma un constituyente del mismo tipo que este último (por ejemplo, un sintagma nominal más un sintagma preposicional forman un sintagma nominal); en un cierto sentido, el adjunto no es necesario sino opcional.

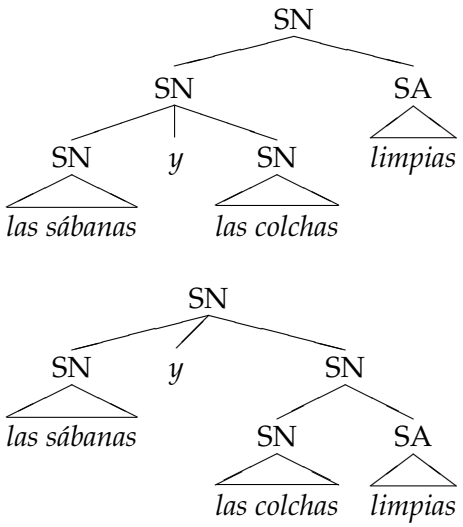


Figura 7.1: Dos árboles para la frase "Pon las sábanas y las colchas limpias en el armario" (SN = sintagma nominal, SA = sintagma adjetival).

nal) que se puede insertar de varias maneras en el árbol de análisis sintáctico de la oración. Por ejemplo, la frase

(7.14) *Juan ha traído noticias de Grecia*

se puede interpretar de dos maneras: en una, el sintagma preposicional *de Grecia* modifica *noticias*; en la otra, modifica *traer* (véanse los árboles de la figura 7.2). Más ejemplos:

(7.15) *Habló con el encargado de la limpieza de su casa*

(7.16) *Hay una bolsa de tela perdida en la Secretaría de la Escuela*

Tuson (1999) explica que esta última oración puede tener hasta 12 interpretaciones posibles.

Para saber más sobre ambigüedad estructural

Otros tipos de ambigüedad estructural:

1. *Ambigüedad estructural debida a la elipsis* de uno o más constituyentes de la oración, especialmente cuando esta oración tendría que tener, si se hubiera producido en forma explícita, una estructura paralela a la de una oración anterior (por ejemplo, en coordinaciones, comparaciones, etc.). Consideremos el ejem-

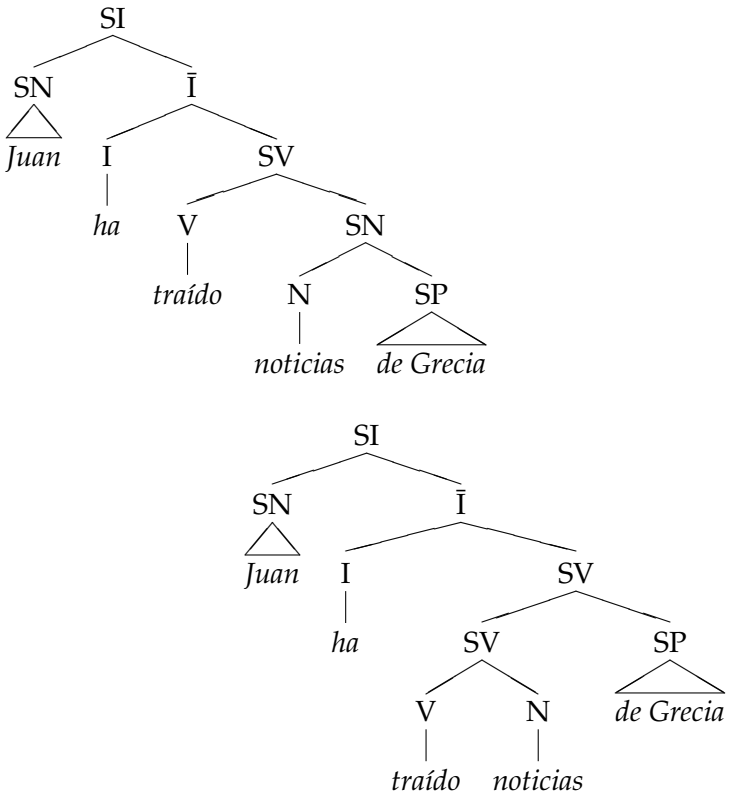


Figura 7.2: Dos árboles para la frase "Juan ha traído noticias de Grecia" (SI = sintagma inflexional, Ī = proyección intermedia de la inflexión, I = inflexión, SV = sintagma verbal, V = verbo, N = nombre, SP = sintagma preposicional).

plo siguiente, sacado de Radford et al. (2009, p.399):

(7.17) *Los escoceses aprecian el whisky más que los galeses*

La oración tiene dos interpretaciones:

(7.18)

(a) *Los escoceses aprecian el whisky más que los galeses (aprecian el whisky)*

(b) *Los escoceses aprecian el whisky más que (los escoceses aprecian) a los galeses*[a]

En estos dos casos, la ambigüedad está causada por el hecho de que son posibles dos estructuras sintácticas para la segunda oración coordinada: en la primera estructura, el sintagma *los galeses* es el sujeto mientras que en la segunda estructura es el objeto (véanse los árboles de la fig. 7.3).

2. *Ambigüedad estructural por movimiento de Qu..* A veces, el análisis sintáctico de una oración se complica por la presencia de fenómenos de movimiento de constituyentes. Consideremos la oración

(7.19) ¿Quién dice que vendrá?

Esta oración tiene, básicamente, dos interpretaciones. Una es

(7.20) ¿Quién dice que PRO vendrá?

y la otra

(7.21) *¿PRO dice que quién vendrá?

Es decir, en la primera, el pronombre interrogativo *quién* es el sujeto de la oración principal; en la segunda, es el sujeto de la oración subordinada, el cual ha experimentado el *movimiento de Qu* (en inglés *Wh-movement*) al principio de la oración, que es obligatorio en muchos idiomas — no en todos: el chino o el turco no lo hacen, por ejemplo — para las palabras con función interrogativa. En este caso, como en el ejemplo 7.17, la elipsis permite dos posicionamientos diferentes del pronombre *quién* antes del movimiento de Qu. Pero las ambigüedades causadas por el movimiento de Qu pueden producirse también sin elipsis, como en el ejemplo

(7.22) *¿Cómo dices que Jordi ha explicado que vendría?*

donde la posición inicial del adverbio interrogativo *Cómo* puede ser el resultado de la transformación por movimiento de Qu de tres estructuras hipotéticas diferentes; en cada una de ellas, el adverbio es adjunto de un sintagma verbal diferente:

(7.23)

(a) *¿Dices* cómo *ha explicado que vendría?*

(b) *¿Dices que ha explicado* cómo *vendría?*

(c) *¿Dices que ha explicado que vendría* cómo*?*

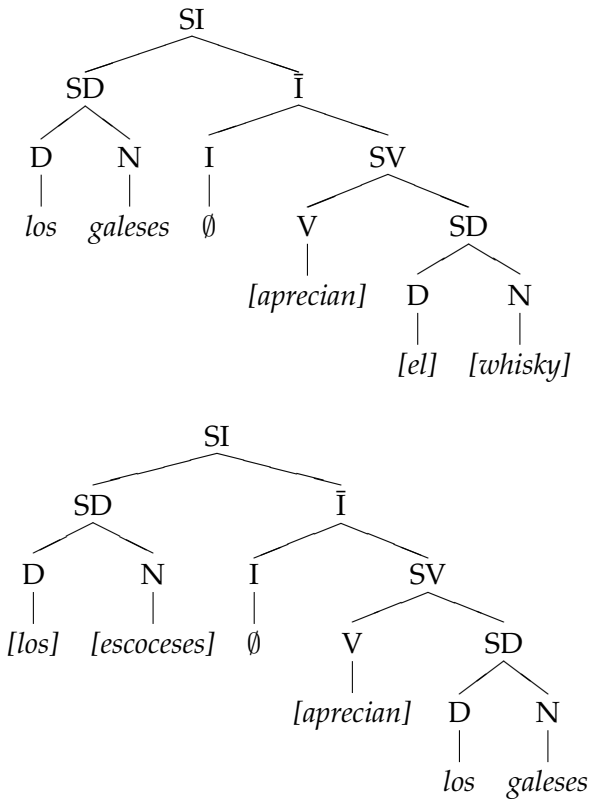


Figura 7.3: Dos árboles para la segunda parte (”los galeses”) de la comparación “Los escoceses aprecian más el whisky que a los galeses” (SD =sintagma determinante, D = determinante).

En la primera interpretación se pregunta por la manera de decirlo, en la segunda por la manera de explicarlo y en la tercera por la manera de venir.

Se producen también movimientos similares con los relativos; por ejemplo, estos se mueven *hacia afuera*, es decir, hacia la raíz del árbol de análisis sintáctico, desde las subordinadas sustantivas completivas con verbos del tipo de *decir*, *explicar*, etc. En la oración

(7.24) No me gusta la manera como dijiste que vivía todavía.

el primer *como* es un relativo que puede modificar *decir* en la oración *dijiste que vivía todavía* ("no me gusta la manera de decirlo") o puede modificar *vivía* pero ha sido movido fuera de la subordinada completiva *vivía todavía*, que modifica *la manera* ("no me gusta la manera como vivía, según lo que dijiste").

[a] De hecho, para evitar esta ambigüedad, se considera conveniente *pero no obligatorio* en español la solución alternativa con preposición *a los galeses* para la segunda interpretación.

7.2.3. Ambigüedades mixtas

Hay oraciones que son ambiguas porque contienen palabras ambiguas o porque tienen más de una estructura sintáctica posible. Estudiaremos dos casos:

1. La oración contiene palabras afectadas por ambigüedad léxica categorial con cambio de categoría (véase la pág. 122). Por ejemplo, la palabra catalana *deu* puede querer decir "nueve más uno" (numeral) o "tiene que dar o pagar" (verbo). O la palabra también catalana *cap* que puede ser un sustantivo ("parte superior del cuerpo"), un verbo (forma del verbo "cabre"; caber), un adjetivo o pronombre ("no hay ninguno"), o parte de la preposición compuesta "cap a' (hacia)'.

   Este tipo de ambigüedad léxica puede provocar a veces ambigüedad estructural, causada por la presencia de más de un análisis sintáctico aceptable (si, a pesar de los homógrafos, sólo hay un análisis aceptable, la ambigüedad pasa desapercibida para el receptor; esto es así porque habitualmente sólo se consideran estructuras aceptables cuando se quiere asignar interpretación a una oración). Por ejemplo, la frase inglesa

   (7.25) *Time flies like an arrow*

   quiere decir normalmente *El tiempo vuela (como una flecha)* pero también son posibles otras dos interpretaciones (semánticamente alocadas pero sintácticamente impecables): *A las moscas del tiempo les gusta una flecha* o *Cronometra las moscas como una flecha*. Esta variedad de interpretaciones se debe al hecho de que hay tres palabras en la frase

que pueden pertenecer a dos categorías léxicas diferentes: *time* puede ser verbo (*cronometrar*) y sustantivo (*tiempo*), *flies* puede ser verbo (*vuela*) y sustantivo (*moscas*) y *like* puede ser verbo (*gustar*) y preposición (*como*). De los 8 ($2 \times 2 \times 2$) análisis morfológicos posibles de la frase, tres resultan sintácticamente aceptables, con interpretaciones muy diferentes. Este tipo de ambigüedad se suele denominar *ambigüedad estructural de origen categorial*. En español —y en general en las lenguas románicas— son muy comunes las ambigüedades debidas a la combinación de una palabra que puede ser pronombre de tercera persona o artículo (*el*, *la*, *los*, *las*) y otra palabra que puede ser sustantivo o verbo conjugado. Por ejemplo, la oración catalana

(7.26) *La mata el vol*

puede querer decir dos cosas, según la elección de categorías léxicas ("el acto de volar le provoca la muerte" o "la planta siente aprecio por él").

2. Otro tipo de ambigüedad mixta sucede cuando la ambigüedad léxica categorial de algunas palabras se combina con mecanismos de elipsis como los descritos más arriba para construcciones coordinativas o comparativas. Por ejemplo, la oración

   (7.27) *Las gallinas han destrozado el sembrado, pero no las matas*

   tiene dos interpretaciones:

   (7.28)

   (a) *Las gallinas han destrozado el sembrado, pero (las gallinas) no (han destrozado) las matas.*

   (b) *[Las gallinas]$_i$ han destrozado el sembrado, pero (tú) no las$_i$ matas.*

   En (7.28a) *las matas* es un sintagma determinante compuesto de un artículo y un sustantivo, que hace de objeto del verbo elíptico *destrozado*, mientras que en (7.28b) *las matas* es un sintagma verbal compuesto de un pronombre (*las*) que se refiere a *Las gallinas* y un verbo (*matas*), sintagma que constituye un sintagma verbal en la segunda oración coordinada (véase la fig. 7.4).[8]

[8] Decir que la anáfora es sólo un proceso léxico es una simplificación. Hay involucrados aspectos sintácticos. Por ejemplo, en la oración *María habló con ella*, el pronombre *ella* no puede nunca referirse a *María*, pero en la oración *María habló con una amiga de ella*, sí que puede, y esto es debido al hecho de que en la estructura sintáctica de la primera oración hay una *barrera* a la correferencia que en la segunda no existe.

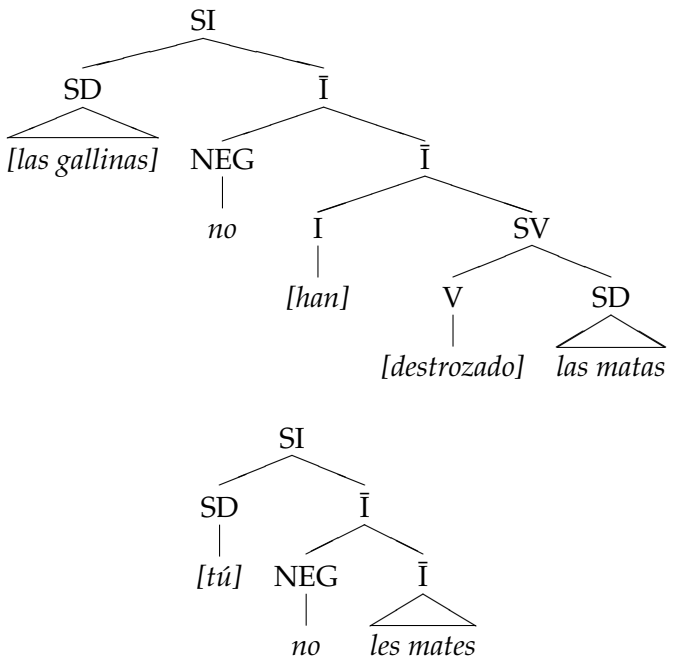


Figura 7.4: Dos árboles para la segunda oración coordinada en "Las gallinas han destrozado el sembrado pero no las matas"(NEG = negación). Los triángulos se usan para no tener que indicar todos los detalles de un subárbol concreto.

Para saber más sobre ambigüedades más complejas

Hay también ciertos tipos de ambigüedad que no se pueden explicar de manera sencilla con el principio de composicionalidad, como por ejemplo *la ambigüedad en el alcance de los cuantificadores*. Los cuantificadores son palabras como *alguno*, *todo*, *cada*. Cuando el alcance de un cuantificador (es decir, las palabras a las que afecta) es impreciso, una oración puede tener más de una interpretación. Consideremos el ejemplo en español de Hutchins y Somers (1992)

(7.29) *A todas las mujeres no les gustan los abrigos de piel.*

Este ejemplo puede tener dos interpretaciones

(7.30)

(a) No todas las mujeres aprecian los abrigos de piel.

(b) A ninguna mujer le gustan los abrigos de piel.

a pesar de no tener ninguna ambigüedad léxica ni estructural aparente. Este tipo de ambigüedad se puede explicar por el hecho de que el principio de composicionalidad por sí solo no es suficiente para especificar completamente la asignación de interpretación a una oración. En palabras de Radford et al. (1999, p.364) "tenemos que reconocer [la existencia] de un vacío inaceptable entre lo que proporciona la sintaxis y lo que la semántica necesita en el caso de oraciones que contengan sintagmas nominales cuantificados". La interpretación de las oraciones se suele explicar a veces en términos de *formas lógicas* (Radford et al. 2009, cap. 23); en el caso de las oraciones con cuantificadores, estas formas lógicas contienen por un lado, *variables* que pueden referirse a un rango de objetos que hay que considerar y, por otro, operaciones sobre estas variables. Pues bien, en estos casos, se puede asignar más de una forma lógica a una oración.

7.2.4. Estrategias de resolución de la ambigüedad

En general, los humanos usamos nuestros conocimientos, nuestras expectativas y nuestras creencias sobre el funcionamiento del mundo real (o de un mundo ficticio concreto, como en una novela) para elegir una de las interpretaciones como la más verosímil (es decir, para *resolver la ambigüedad*); cuando los conocimientos, las creencias y las expectativas están compartidas entre el emisor y el receptor, se puede usar la ambigüedad como un mecanismo muy eficiente para producir mensajes más cortos.

Como hemos visto, las causas de la ambigüedad son muy diversas; por eso, también son muy diversas las estrategias de resolución. Este epígrafe recoge unas notas —no exhaustivas— sobre las estrategias de resolución de algunos tipos de ambigüedad en sistemas automáticos de tratamiento del lenguaje humano.

Las estrategias de resolución de la ambigüedad suelen basarse en *restricciones* y *preferencias*. Como veremos, las restricciones son normalmente de naturaleza lingüística —por lo tanto, requieren un cierto nivel de análisis del texto— y permiten descartar ciertas interpretaciones, pero no eliminan

completamente la ambigüedad. Para acabar de resolver la ambigüedad, se usan preferencias: se asigna algún tipo de puntuación o valor a cada interpretación para elegir la mejor. Las preferencias se suelen basar frecuentemente en métodos estadísticos, basados en observaciones obtenidas de grandes cantidades de texto.

Resolución de la ambigüedad léxica categorial

La resolución de la ambigüedad léxica categorial de las palabras homógrafas, también conocida como etiquetado (de las palabras) con partes de la oración (en inglés, *part of speech (PoS) tagging*) está muy bien estudiada. La ambigüedad se reduce normalmente usando restricciones basadas en el conocimiento lingüístico, y, como normalmente con esto no suele ser suficiente, se establecen preferencias basadas en el estudio estadístico de la frecuencia de aparición conjunta en los textos de determinadas secuencias cortas de categorías léxicas.

En algunos casos, las restricciones pueden ser suficientes. Por ejemplo, la palabra *ahorro* puede ser sustantivo o verbo. Si aparece entre un artículo y un adjetivo como por ejemplo en *el ahorro doméstico* no hay ninguna duda de que se trata de un sustantivo: la secuencia determinante–verbo personal no está permitida.

Sin embargo, a veces, las restricciones sólo reducen la ambigüedad sin eliminarla completamente. Por ejemplo, la palabra *sobre* puede ser un sustantivo masculino singular, una preposición, y tres formas del verbo *sobrar* (presente de subjuntivo, 1ª y 3ª persona del singular, e imperativo de cortesía, 3ª persona del singular). En el contexto *Tráeme aquel sobre de la caja*, una vez hecho el análisis morfológico de las palabras de la oración, se podrían aplicar restricciones lingüísticas basadas en secuencias de dos categorías léxicas para reducir la ambigüedad. Por ejemplo, una preposición no puede ir seguida de otra preposición. Como *sobre* va seguido de *de*, que sólo puede ser preposición, podemos descartar que *sobre* sea una preposición. Pero no se pueden descartar el resto de formas léxicas: si *aquel* es un determinante, *sobre* puede ser un nombre (como es el caso); si *aquel* es un pronombre, *sobre* podría ser un verbo (como en las oraciones *Ahora nos han sobrado dos coches, pero puede ser que aquel sobre también más adelante.*).

Por lo tanto, hay que considerar el uso de preferencias. Por ejemplo, podemos usar la aproximación estadística. Si cogemos un corpus (conjunto) suficientemente grande de textos en el que un experto ha indicado la categoría léxica de cada palabra y contamos cuántas veces aparecen todas las secuencias posibles de dos categorías léxicas, podemos usar estas frecuencias para asignar la categoría de una palabra ambigua: de todas las secuencias de tres palabras posibles que se puedan formar con esta palabra, cogeremos la más frecuente.

Para saber más sobre estrategias de resolución de la ambigüedad

Resolución de la polisemia. La resolución de la polisemia (en inglés *word sense disambiguation*) consiste en asignar a una palabra polisémica, en un texto o discurso, una interpretación concreta, posiblemente diferente de las que podría tener en otros textos (o contextos). La desambiguación se efectúa usando información procedente de tres fuentes: el *cotexto* (interno al texto o discurso) y el *contexto* (externo al texto o discurso pero relacionado con él) y fuentes de conocimiento adicionales. En traducción automática, estamos interesados en elegir una de las interpretaciones posibles, porque es habitual que las palabras polisémicas tengan varias traducciones (la *ambigüedad de transferencia* mencionada en el apartado 7.2.1).

Se acepta comúnmente que la mayor parte de las palabras polisémicas de un texto (o de un fragmento del texto) suelen tener una única interpretación en un texto dado, pero este principio se tiene que concretar en un método específico para resolver la polisemia.

La resolución de la polisemia se ha abordado desde perspectivas muy diversas (véase Ide y Veronis (1998)). Es posible aplicar restricciones para resolver la polisemia pero se tienen que basar en un análisis de naturaleza bastante profunda. Por ejemplo, podemos decidir que la palabra *gato* es un animal y no una herramienta para levantar vehículos en la frase *El gato me miró desde debajo del coche*, porque *gato* es el sujeto de *miró*, y el verbo *mirar* requiere un sujeto animado, pero como puede verse, esto requiere que se haya hecho un análisis sintáctico y semántico.

Por eso, en general se usan métodos basados en preferencias. He aquí dos ejemplos:

- El uso de *redes semánticas* donde los conceptos se sitúan en los nodos (nudos) de la red y se agrupan jerárquicamente en superconceptos cada vez más generales (por ejemplo, los conceptos *manzana*, *pera*, *naranja* se agruparían bajo el concepto *fruta*): un ejemplo de redes semánticas es *Wordnet*, `http://wordnet.princeton.edu`, que se está generalizando a otras lenguas de Europa (`http://www.illc.uva.nl/eurowordnet/`). Cuando tenemos una palabra polisémica, le podemos asociar más de un concepto o *sentido*. Para elegir uno, podemos, por ejemplo, tomar todos los posibles sentidos de la palabra ambigua y asignarles el sentido asociado al concepto que está más cerca de los conceptos representados por las palabras vecinas en el texto. La información presente en diccionarios electrónicos preexistentes puede servir para construir estas redes o ser usada directamente para la resolución de la polisemia.
- La estadística de aparición conjunta de palabras en corpus bilingües de textos puede ayudar a resolver directamente la ambigüedad de transferencia cuando se dispone de diccionarios de transferencia o cuando los textos están alineados. Por ejemplo, si en un corpus bilingüe español–catalán la aparición de *destino* cerca de *incierto* en español coincide con la aparición de *destí* en catalán, podemos decir que la palabra *destino* tiene en este caso la interpretación de "suerte futura"; en cambio, si la aparición de *destino* cerca de *estación* o *aeropuerto* en español coincide con la aparición *destinació* en catalán, podemos elegir el sentido de "punto de llegada". Esta información podría servir para traducir después del español al inglés y elegir *destiny* o *destination* en cada caso con mucha probabilidad de éxito.

Resolución de la anáfora. La resolución de la anáfora —es decir, la determinación del *antecedente* de un pronombre o de otra expresión anafórica— se puede basar

también en restricciones y preferencias.

Las *restricciones* se pueden basar en información morfológica, sintáctica, o incluso semántica; todo depende del nivel de análisis que esté disponible:

- Un pronombre masculino no puede tener un antecedente femenino (restricción morfológica): *María* no puede ser el antecedente de *él* en la oración *María se pasó todo el día hablando de él*.
- La información sintáctica puede ser más relevante de lo que parece: si decimos

  (7.31) *Marta la vio*

  el antecedente de *la* no puede ser *Marta*, a causa de las llamadas *barreras*, restricciones asociadas a determinadas características de la estructura sintáctica de la oración. En cambio, si decimos

  (7.32) *Marta habló con quien la vió*

  no se puede descartar completamente que el antecedente de *la* sea *Marta*.
- Hay veces que sólo podemos recurrir a un análisis semántico; en el ejemplo (ya discutido en la sección 6.3)

  (7.33) *[Los soldados]$_i$ dispararon [a los niños]$_j$. Los$_{i/j?}$ vi caer*

  se tiene que usar información semántica para saber cuál es el antecedente de *los* en la segunda oración (*los soldados* o *los niños*).

Las restricciones no suelen ser suficientes, y suele ser necesario el establecimiento de *preferencias*. Por ejemplo, se pueden preferir

- los antecedentes más recientes,
- los antecedentes que hacen de sujeto a los que hacen de objeto, o
- los antecedentes que han sido introducidos explícitamente como el asunto del discurso o de la conversación: *Pues, en cuanto a Joan...*.

Esto se suele instrumentar a través de un sistema que asigna *puntuaciones* a cada una de las características: se suman las puntuaciones para todos los antecedentes posibles y se elige el que obtiene la puntuación más alta (Lappin y Leass 1994).

Resolución de la ambigüedad estructural. En principio, se podría decir que las personas resuelven la ambigüedad estructural —pura o de origen categorial— eligiendo, usando las interpretaciones asignadas a cada una de las estructuras posibles (principio de composicionalidad), cuáles son *aceptables* y, entre las aceptables, cuál es la más verosímil y por tanto preferible en una situación comunicativa determinada. Según este modelo, las personas consideraríamos siempre *todas* las estructuras sintácticas. Se podría argumentar en contra fácilmente diciendo que en frases complejas (por ejemplo, la oración 7.16) hay demasiadas estructuras a considerar. De hecho, hay experimentos psicolingüísticos que indican que a veces usamos estrategias puramente sintácticas, eligiendo entre las posibles estructuras incluso cuando no hemos oído o leído toda la oración, quizás para evitar un esfuerzo intelectual excesivo, puesto que puede haber muchísimas interpretaciones parciales. A cambio, tenemos que hacer el esfuerzo (presumiblemente más ligero) de predecir una entre las posibles continuaciones (sintácticas) de lo que hemos leído; según llegan palabras, las vamos encajando en la estructura predicha y usamos la sintaxis y la interpretación de las palabras para ir construyendo poco a poco la interpretación de la oración completa. La experiencia nos ayuda a hacer predicciones que en general tienen éxito, pero a veces hay oraciones "engañosas" que "nos llevan al huerto" (llamadas, por eso, en inglés

garden-path sentences, del inglés *lead up the garden path*) puesto que en cierto punto del proceso nos obligan a descartar la predicción hecha y reinterpretar lo que habíamos leído hasta aquel punto (el estudio de los movimientos oculares, en inglés *eyetracking*, durante la lectura dan pistas muy relevantes sobre la existencia de estos procesos). He aquí algunos ejemplos de oraciones que "nos llevan al huerto", con una continuación inesperada en las notas a pie de página:

(7.34) *Juan besó a Maria y su hermana...*[a]

(7.35) *Como Joan siempre corre un par de kilómetros...*[b]

(7.36) *En el otro accidente murieron sesenta y cinco...*[c]

(7.37) *The horse raced by the barn...*[d]

Estos procesos de selección puramente sintáctica dan como resultado que hay ciertas estructuras finales que se prefieren a otras, quizás porque simplifican la comprensión. Por ejemplo, si leemos

(7.38) *Aprendió a afeitarse en dos minutos*

podríamos considerar la interpretación en la que se habla de la duración del afeitado (lo que *aprendió* es a *afeitarse en dos minutos*) como más probable que la que interpreta que se habla de la duración del aprendizaje (*aprender a afeitarse* le costó *dos minutos*), puesto que en el segundo caso quizás habría sido más natural decir

(7.39) *Aprendió en dos minutos a afeitarse*

La regla que favorece que los adjuntos se asocien al último sintagma que los admita —y que permite, por lo tanto, ir construyendo el árbol de análisis sintáctico gradualmente sin tener que hacer grandes reorganizaciones— se suele denominar regla de *clausura tardía* —en inglés *late closure*—; por ejemplo, esta regla favorece el primero de los árboles de la figura 7.2. Otra regla que se suele usar es la de *adjunción mínima* —en inglés *minimal attachment*— que favorece el árbol sintáctico con el mínimo de nodos (puntos de ramificación). Estas estrategias son de utilidad en los sistemas de traducción automática por transferencia sintáctica pura (véase la sección 8.3), puesto que no se hace ningún procesamiento semántico.

El punto de vista puramente sintáctico se puede considerar una simplificación excesiva; muchas veces, las personas resolvemos la ambigüedad estructural usando restricciones semánticas o incluso lèxico-semánticas:

- Por ejemplo, el verbo *vender* admite un objeto directo y uno indirecto, pero el verbo *comer* sólo el directo, de forma que si decimos "Le presentó al hombre que vendía naranjas a Joan" se puede interpretar de dos maneras a causa de la ambigüedad estructural, pero si decimos la frase estructuralmente idéntica —y por lo tanto idénticamente ambigua— "Le presentó al hombre que comía naranjas a Joan" no hay más que una interpretación posible.
- Considerad estas dos frases estructuralmente idénticas afectadas por la misma ambigüedad estructural pura de adjunción:

  (7.40) *Tráeme las llaves del armario grande*

  (7.41) *Tráeme las llaves de la silla verde*

  En la oración 7.40, podemos dudar, puesto que no sabemos si las llaves son las que abren el armario o las que están allá guardadas. En cambio, en la oración 7.41 no consideramos la primera interpretación (aunque sea la preferida sintácticamente según la regla de clausura tardía), porque no es nada verosímil

que las sillas tengan cerradura (hemos usado información semántica basada en nuestras creencias sobre el mundo). Si el sistema que resuelve la ambigüedad es capaz de usar información semántica, podría elegir correctamente en este caso.

---

[a] ...le recriminó por haberlo hecho.
[b] ...le parecen poco.
[c] ...resultaron heridos.
[d] ...fell down.

7.3. Cuestiones y ejercicios

Para poder responder a las preguntas marcadas con (*) hace falta que os leáis los cuadros *Para saber más*.

1. Indicad qué clase de ambigüedad presentan estas frases (justificad muy brevemente vuestra respuesta):

   a) *Expulsarán al alcalde de la ciudad* (1: "El alcalde de la ciudad será expulsado." 2: "El alcalde será expulsado de la ciudad").

   b) *Había un gato bajo el automóvil* (1: "...porque acababan de reparar una rueda"; 2: "...y salió corriendo cuando lo puse en marcha")

   c) *María entró con una bolsa grande. Yo la puse encima de la mesa* (1: "Puse a María encima de la mesa"; 2: "Puse la bolsa encima de la mesa")

   d) *¿Qué quieres, galletas o pan de la tía Pepa?* (¿Las galletas son también de la tía Pepa?)

   e) *Pon una resma de papel en la impresora y conéctala.* (¿Tiene que conectar la resma de papel o la impresora?)

   f) *¿Os han dicho que vaya?* (¿Quién tiene que ir?).

   g) *Vale más que las* comas (1: "...que los signos de puntuación"; 2:"...que las ingieras")

   h) *El mecánico revisó la suspensión del auto de Garzón* (1: "Este mecánico es un experto en legislación y se ha leído la resolución judicial entera"; 2: "Los amortiguadores del coche de Garzón ya necesitaban una revisión").

   i) *A pesar de haber sido soldado, salió despedido del avión* (1: "El sistema fotográfico estaba fuertemente fijado al fuselaje pero se soltó del aparato cuando el avión giró en pleno vuelo"; 2: "A pesar de su pasado militar glorioso, el presidente lo destituyó antes de llegar al aeropuerto de destino").

j) *Los ladrones fueron atrapados en una fábrica incendiada por un policía* (1: "Los ladrones fueron capturados por el comisario en una fábrica abandonada"; 2: "La fábrica donde fueron capturados fue el objetivo de un agente pirómano").

k) *Coto privado de caza'* (1: "Esta área de caza no es pública"; 2: "Esta es una área sin caza").

l) *Quiero bailar y cantar canciones de cuna* (1: "Bailaremos durante un rato y después te cantaré para que te duermas"; 2: "Quiero tanto bailar canciones de cuna como cantarlas").

m) *El manifestante se recupera de la paliza que le dieron en el hospital* (1: "La manifestación fue un poco violenta y algunas personas han tenido que ser trasladadas al hospital"; 2: "¡Qué paliza le dio la enfermera en el quirófano!").

n) *Servirán pulpo a la gallega* (1: "Servirán pulpo a una señora de Galicia"; 2: "Servirán pulpo preparado al estilo gallego").

ñ) *No puedo ver bien la foto que me has enviado por correo electrónico porque no puedo cerrar todas las ventanas* (1: "Todavía entra sol y se refleja en la pantalla"; 2: "Tengo el escritorio lleno de documentos abiertos").

o) *–Hem rebut notícies que diuen que, per causa de la humitat i la calor en l'interior del temple, els bancs i els altars de fusta han rebrotat i els han crescut branques i fulles. – I les creus?'* (1: "¿Crees estas noticias?"2: "¿Los crucifijos también han rebrotado?").

p) *Se tienen que repasar las entradas y los gastos que se hayan hecho en euros* (1: "Las entradas se tienen que repasar todas; los gastos sólo si se han hecho en euros"; 2: "De las entradas, se tienen que repasar sólo las hechas en euros").

q) *Después de que la vendedora acabara la descripción de las ventajas de la urbanización proyectada, el comentario unánime de los inversores fue que les parecía muy interesante'*. (1: "Los inversores, la verdad, prestaban más atención a la vendedora que al producto"; 2: "Les gustó el estilo de la descripción"; 3: "La vendedora no hablaba claro, la descripción estaba incompleta, pero a pesar de todo, la urbanización era una inversión prometedora").

r) *A la trapecista, últimamente, no le salían los números* (1: "Siempre acababa cayendo en la red"; 2: "tenía más gastos que ingresos").

2. (*) Hay ambigüedades de tipo léxico que pueden ser siempre correctamente resueltas después de hacer un análisis morfológico. Sin embargo, hay otras que sólo pueden ser tratadas si se hace un análisis sintáctico (a pesar de que la ambigüedad sea de tipo léxico) e incluso las hay que requerirían un análisis semántico para resolverlas.

Elegid una lengua origen y una lengua meta (francés, inglés, alemán, catalán o español) y poned un ejemplo de oración para cada uno de los tres casos anteriores, donde sea necesario un determinado nivel de análisis para resolver una ambigüedad y producir la traducción correcta. Explicad qué información usa el sistema en cada caso para tomar una decisión.

3. (*) Indicad brevemente qué estrategias se podrían usar para resolver la ambigüedad sintáctica de adjunción. Para inspiraros, fijaos en los siguientes ejemplos:

   - *Aprendió en dos minutos a afeitarse*
   - *Aprendió a afeitarse en dos minutos*
   - *Tráeme las llaves del armario grande*
   - *Tráeme del armario grande las llaves*
   - *Tráeme las llaves de la silla verde*
   - *Toni comprará las naranjas que tiene que vender a Reme*
   - *Toni comprará a Reme las naranjas que tiene que vender*

4. Si una oración tiene sólo una ambigüedad léxica pura...

   a) ...tiene un único árbol de análisis sintáctico, pero más de un análisis morfológico.

   b) ...tiene un único árbol de análisis sintáctico y un único análisis morfológico, pero dos interpretaciones semánticas diferentes.

   c) ...tiene más de un árbol de análisis sintáctico.

5. La frase *"Me gusta más que la bata"* puede tener dos interpretaciones; en la primera se habla de una prenda de vestir; en la segunda, de una preferencia a la hora de preparar, por ejemplo, una salsa. Indicad de qué clase de ambigüedad se trata.

   a) Estructural de adjunción.

   b) Léxica categorial.

   c) Estructural de origen categorial.

6. La frase *Baja y sube en ascensor* puede querer decir "(baja) y (sube en ascensor)" o "(baja y sube) en ascensor". ¿De qué tipo de ambigüedad se trata?

   a) Léxica categorial.

   b) Estructural de origen categorial.

   c) Estructural de origen coordinativo.

7. En la oración *El coche se ha quemado con el garaje y el seguro no lo cubre* no se sabe cuál de las dos cosas está cubierta por el seguro, el garaje o el coche. La ambigüedad...

   a) ... se debe a la elipsis.

   b) ... se debe a la anáfora.

   c) ... es estructural de origen coordinativo.

8. ¿De qué clase es la ambigüedad de la oración *Vendió las naranjas que había comprado a María*?

   a) Estructural de origen coordinativo.

   b) Estructural de adjunción.

   c) Extrasentencial por anáfora.

9. Considerad el homógrafo *vendo* ("Te vendo un coche" "Y yo, ¿para qué quiero un coche vendado?"). ¿Se puede resolver la ambigüedad léxica a la que da lugar usando sólo información sintáctica (es decir, sobre las categorías léxicas que lo acompañan en la oración)?

   a) No, porque las dos formas *vendo* se escriben exactamente igual.

   b) No, porque las dos formas *vendo* tienen la misma categoría léxica y el mismo análisis morfológico, salvo por el lema, y, por lo tanto, pueden hacer exactamente las mismas funciones sintácticas.

   c) Sí, sólo mirando la categoría léxica de las palabras anteriores y la de los posteriores ya hay bastante para saber en cuál de los dos casos nos encontramos.

10. (*) Muchas veces, la ambigüedad léxica no es ni polisemia (*estación*, *bomba*), ni ambigüedad léxica con cambio de categoría gramatical (*sobre* [preposición y sustantivo], *río* [sustantivo y verbo]) sino que sucede porque dos formas *de la misma categoría léxica* son homógrafas: en catalán *volem* puede ser una forma del verbo *volar* o del verbo *voler*; en catalán *podeu* puede ser una forma del verbo *poder* o del verbo *podar*; en español, *creo* puede ser una forma del verbo *creer* o del verbo *crear*, *fui* puede ser una forma del verbo *ir* o del verbo *ser*, etc. Para resolver la ambigüedad de una palabra polisémica se tiene que usar información semántica; para resolver la ambigüedad léxica categorial suele ser suficiente con usar información sintáctica (por ejemplo, la categoría gramatical de la palabra anterior y posterior); pero, ¿es posible resolver la ambigüedad debida a la homografía de palabras de la misma categoría usando sólo la sintaxis o es necesario el uso de información semántica?

11. Los sistemas de traducción palabra por palabra pueden cometer, por ejemplo, errores debidos a la elección incorrecta de la categoría gramatical de una palabra léxicamente ambigua. Elegid dos lenguas, L_1 y L_2 y poned dos ejemplos de traducciones erróneas de L_1 a L_2, indicando la frase original, la frase mal traducida y la traducción correcta.

12. Si una forma superficial es ambigua pero tiene tan solo una forma léxica...

    a) ... se trata de una palabra homógrafa.

    b) ... hay algún error en el análisis morfológico.

    c) ... podemos decir que el lema es polisémico.

13. ¿Puede una oración tener más de una traducción a otro idioma a pesar de estar formada completamente por palabras que no son ni homógrafas ni polisémicas en la lengua origen?

    a) No: lo prohíbe el principio de composicionalidad semántica.

    b) Sí, aunque no contenga pronombres u otras expresiones anafóricas susceptibles de tener más de un antecedente posible.

    c) Sí, pero sólo si contiene pronombres u otras expresiones anafóricas susceptibles de tener más de un antecedente posible.

14. (*) En ausencia de información léxica o semántica, la ambigüedad estructural...

    a) ... es imposible de resolver.

    b) ... se puede resolver usando reglas derivadas de un estudio de las preferencias sintácticas observadas en experimentos psicolingüísticos.

    c) ... no puede afectar nunca al resultado de la traducción automática.

15. (*) ¿Es posible resolver en algunos casos la ambigüedad debida a una palabra homógrafa utilizando exclusivamente información morfológica?

    a) No, este tipo de ambigüedad exige un tratamiento semántico como mínimo.

    b) No, siempre hay que utilizar información de carácter sintáctico para resolverla.

    c) Sí, usando la información morfológica de las palabras adyacentes.

16. ¿Si una palabra tiene sólo una forma léxica y una única traducción a una determinada lengua, puede ser todavía ambigua?

    a) No.

    b) Sí, puede ser polisémica aunque la traducción a esta lengua de todas las interpretaciones sea la misma.

    c) Sí; puede ser homógrafa y tratarse de un pase gratuito.

17. Si traducimos automáticamente la frase *Ayer cantamos las mismas canciones* y obtenemos en inglés *Yesterday we sing the same songs* o en francés *Hier nous chantons las mêmes chansons*, ¿qué tipo de ambigüedad ha sido mal resuelta?

    a) Una ambigüedad léxica por homografía de una palabra.

    b) Una ambigüedad léxica pura por polisemia de una palabra.

    c) Una anáfora.

18. Si traducimos automáticamente la frase catalana *Hilari no coneix bé la Mariona: cada dia troba sorprenent el que fa* y obtenemos en inglés *Hilari does not know Mariona well: every day he finds what he does astonishing* o en francés *Hilari ne connaît bien Mariona: tous les jours elle trouve ce qu"il fait etonnant*, ¿qué tipo de ambigüedad ha sido mal resuelta?

    a) La anáfora de un pronombre vacío.

    b) La anáfora del pronombre *que*.

    c) Una ambigüedad sintáctica de la oración subordinada "el que fa".

19. El principio de composicionalidad dice que la interpretación de una oración está determinada por las interpretaciones de las palabras y por la sintaxis. Cuando se produce una ambigüedad porque no queda clara la adscripción de un sintagma preposicional, como *trae la llave del armario grande*, ¿cuál es la razón?

    a) La existencia de más de una estructura sintáctica posible.

    b) La ambigüedad categorial de la preposición.

    c) La polisemia de la preposición.

20. La ambigüedad léxica categorial de una palabra...

    a) ...no se puede resolver nunca si no se usa información semántica sobre el texto o sobre las palabras contiguas.

    b) ...no se puede resolver si no se hace el análisis sintáctico completo de la frase, ya que ésta es la única manera de elegir el análisis morfológico correcto.

c) ... se intenta resolver normalmente con reglas basadas en las categorías léxicas de las palabras que la acompañan en la frase.

21. ¿Qué tipo de ambigüedad se produce en el pronombre débil *li* de la frase en catalán *Vaig veure Mario i sa madre; li vaig dir que m'agradava molt el seu fill*?

    a) Homografía, porque el pronombre puede, en principio, estar sustituyendo un nombre u otro.

    b) Una anáfora.

    c) Polisemia.

22. Una de estas tres no es un tipo de ambigüedad léxica:

    a) La homografía.

    b) La ambigüedad de adjunción.

    c) La polisemia.

23. ¿De qué clase es la ambigüedad que presenta la oración *Expulsarán al portavoz del partido*?

    a) Léxica, debida al hecho de que la palabra *expulsar* es polisémica.

    b) Estructural de origen coordinativo: no sabemos si el sintagma preposicional *del partido* modifica sólo al segundo elemento *el portavoz* o a todo el sintagma *Expulsarán al portavoz*.

    c) Estructural de adjunción: el sintagma preposicional *del partido* puede ser un adjunto del sintagma verbal *Expulsarán al portavoz* o del sintagma nominal *el portavoz*.

24. Solo una de estas tres afirmaciones es correcta. ¿Cuál?

    a) Una oración puede ser ambigua sin que ninguna de sus palabras sea ambigua por sí misma.

    b) Una oración sólo puede ser ambigua si al menos una de sus palabras es ambigua.

    c) El hecho que una oración sea ambigua implica necesariamente que las traducciones de las diversas interpretaciones a otra lengua tienen que ser diferentes.

25. ¿Qué tipo de ambigüedad se da en la oración *La mata la envidia* (1: "la envidia la está matando"; 2: "la planta le tiene envidia a ella")?

    a) Ambigüedad léxica por polisemia.

    b) Ambigüedad léxica debida a la anáfora.

c) Ambigüedad estructural debida a la ambigüedad léxica categorial.

26. Cuando un adjetivo presenta la misma ambigüedad (por ejemplo, puede tener más de una traducción), independientemente de cómo se encuentre flexionado en género y número, diremos que el adjetivo es...

    a) ... anafórico.
    b) ... homógrafo.
    c) ... polisémico.

27. Según Arnold (2003) los problemas a los cuales se enfrenta la traducción automática son cuatro. Indicad cuál de las afirmaciones siguientes es falsa:

    a) El problema del análisis se refiere a la dificultad para resolver la ambigüedad de un enunciado.
    b) El problema de la síntesis se refiere a la ambigüedad de los textos traducidos automáticamente.
    c) El problema de la descripción consiste en que es impracticable describir de forma suficiente y computacionalmente eficiente todo el conocimiento necesario para traducir.

28. Un traductor automático por transferencia morfológica avanzada ...

    a) ...resuelve la polisemia mediante el uso de un analizador morfológico.
    b) ...resuelve la polisemia mediante el uso de un desambiguador léxico categorial.
    c) ...no puede resolver la polisemia con ninguno de los programas mencionados en las otras dos opciones.

29. ¿Qué tipo de ambigüedad se da en la oración "*Aston Family Man era el bajo de The Wailers*"(Aston era el más bajo del grupo; Aston tocaba el bajo en el grupo)?

    a) Ambigüedad léxica por polisemia.
    b) Ambigüedad léxica categorial dentro de la misma categoría léxica.
    c) Ambigüedad léxica categorial entre categorías léxicas diferentes.

30. Indicad cuál de las afirmaciones siguientes es falsa:

    a) Hay casos en los que no hace falta resolver la ambigüedad para producir una traducción adecuada en la lengua meta.

b) La ambigüedad siempre representa un problema a la hora de traducir entre dos lenguas.

c) La ambigüedad es uno de los problemas a los que tiene que enfrentarse un traductor automático.

7.4. Soluciones

1. *a*) Ambigüedad sintáctica o estructural (pura) de adjunción: el sintagma preposicional *de la ciudad* se puede insertar en dos posiciones diferentes de la oración: puede modificar *alcalde* o *expulsarán [el alcalde]*.

   b) Ambigüedad léxica pura (polisemia) de la palabra *gato*.

   c) Ambigüedad léxica por anáfora: el pronombre *la* puede tener dos antecedentes: *Maria* y la *bolsa*.

   d) Ambigüedad sintáctica o estructural (pura) de origen coordinativo: el sintagma *de la tía Pepa* puede modificar a los dos sintagmas nominales coordinados (*galletas y pan*) o sólo al segundo (*pan*).

   e) Ambigüedad léxica por anáfora: el pronombre *la* puede tener dos antecedentes: *resma [de papel]* y la *impresora*.

   f) Ambigüedad por elipsis: el sujeto de *vaya* puede ser *yo*, *él*, *ella*, etc. En el caso de los pronombres de tercera persona, la interpretación estará determinada por los antecedentes que se les asignen (ambigüedad léxica por anáfora).

   g) Ambigüedad sintáctica o estructural de origen categorial debido al hecho de que las palabras *las* (artículo o pronombre) y *comas* (sustantivo o verbo) son homógrafos afectados de ambigüedad léxica categorial. De las cuatro combinaciones posibles, dos son sintácticamente aceptables.

   h) La oración es ambigua porque dos de sus palabras presentan ambigüedad léxica pura (polisemia): *auto* puede ser un automóvil o un tipo de resolución judicial; *suspensión* puede ser la acción de suspender (la resolución) o el sistema de amortiguadores del automóvil. De las cuatro combinaciones posibles, dos tienen cierto sentido.

   i) La oración es ambigua por la ambigüedad léxica (homografía) de *soldado*. En la primera interpretación es un participio en forma pasiva *haber sido soldado*; en la segunda es un sustantivo masculino singular. También interviene la ambigüedad léxica pura (polisemia) de *despedir* (en la primera *lanzar*; en la segunda, *dejar sin trabajo*). Dos de las cuatro combinaciones tienen sentido.

j) Ambigüedad estructural pura de adjunción. El sintagma preposicional *por un policía* puede modificar el sintagma verbal *atrapados en una fábrica incendiada* o sólo el sintagma verbal *incendiada*.

k) Ambigüedad mixta. Por un lado, léxica: la palabra *privado* puede ser un adjetivo (interpretación 1) o un participio (interpretación 2). Por otro lado, estructural: en la primera interpretación, el sintagma preposicional *de caza* modifica el sintagma nominal *coto privado* (*[[coto privado] [de caza]]*); en el segundo, sólo el participio *privado* (*[[coto] [[privado] [de caza]]]*).

l) Ambigüedad estructural de origen coordinativo. El sintagma nominal *canciones de cuna* puede modificar sólo el segundo sintagma verbal *cantar* o el sintagma verbal completo *bailar y cantar* (es decir, *canciones de cuna* puede ser objeto directo sólo del segundo verbo o de los dos).

m) Ambigüedad estructural pura de adjunción. El sintagma preposicional *en el hospital* puede modificar el sintagma verbal *le dieron* o el sintagma verbal *se recupera de la paliza que le dieron*.

n) Ambigüedad estructural de adjunción: el sintagma preposicional *a la gallega* puede modificar el nombre *pulpo* para formar el sintagma nominal *pulpo a la gallega* o modificar el sintagma verbal *servirán pulpo* (con núcleo *servirán*) y formar el sintagma verbal *servirán pulpo a la gallega*.

ñ) La oración es ambigua por polisemia (ambigüedad léxica) del sustantivo *ventana* (de la pared/ del sistema operativo)

o) Ambigüedad estructural de origen categorial. En la primera interpretación de la frase en catalán *les* es un pronombre y *creus* es un verbo, y forman juntos un sintagma verbal; en la segunda *les* es un artículo y *creus* es un sustantivo y forman juntos un sintagma nominal.

p) Ambigüedad estructural de origen coordinativo. El sintagma (oración subordinada de relativo) *que se hayan hecho en euros* puede modificar al segundo sintagma nominal *los gastos* o al sintagma nominal completo *las entradas y los gastos*.

q) Ambigüedad de la oración debida a la ambigüedad léxica por anáfora. El pronombre *la* puede tener tres antecedentes: *la vendedora*, *la descripción* o *la urbanización proyectada* y, por lo tanto, tres interpretaciones diferentes.

r) Ambigüedad de la oración por polisemia (ambigüedad léxica) de la palabra *número* (parte de una actuación / cuentas económicas)

2. (*) Ejemplos del español al catalán:

- Ambigüedad que se puede resolver después de hacer un análisis morfológico de la oración: en *mi trabajo*, el homógrafo *trabajo* puede ser sustantivo o verbo, pero la presencia de *mi* (determinante posesivo) desambigua el homógrafo perfectamente.
- Ambigüedad léxica que necesita un análisis sintáctico para ser resuelta: la expresión multipalabra *sesenta y cinco* puede ser un único numeral (*65*) o dos numerales coordinados (*60 y 5*). El análisis morfológico no es suficiente para detectar que en la frase *En el lugar donde murieron sesenta y cinco quedaron restos* es el primer caso y en la frase *Murieron sesenta y cinco quedaron malheridos* es el segundo.
- Ambigüedad léxica que necesita un análisis semántico para resolverla: la palabra polisémica *destino* en la oración *El destino estaba escrito en el pasaje arrugado que encontraron* se traduciría al catalán por *destinació* y en cambio en la oración *El destino estaba escrito en el libro sagrado que encontraron* se traduciría al catalán por *destí*; la elección exige identificar relaciones semánticas entre las interpretaciones de las palabras.

3. (*) Véase el cuadro *Para saber más* de la sección 7.2.4. En el caso de las frases "Toni comprará...", parece lógico usar la regla de *clausura tardía*, puesto que se corresponde bastante bien con las interpretaciones preferidas por las personas.

4. (b)

5. (c). Las palabras *la* y *bata* pueden pertenecer cada una a dos categorías léxicas diferentes. De las cuatro combinaciones resultantes, dos son sintácticamente aceptables.

6. (c)

7. (b). El pronombre átono *lo* puede referirse a *garaje* o a *coche*.

8. (b). El sintagma preposicional "a María" puede ser el objeto indirecto de la oración principal y de la subordinada.

9. (b)

10. La solución semántica es más potente y general pero exige un análisis muy profundo de la frase. En algunos casos, la sintaxis podría dar pistas que permitirían una desambiguación muy aproximada. Por ejemplo, si *fui* va seguido de la preposición *a*, es muy probable que se trate del verbo *ir*; por otro lado, si va seguido de un participio pasado, es muy probable que se trate del verbo *ser*: se podría hacer una categoría

gramatical especial para el verbo ser y usar técnicas de desambiguación categorial. Si encontramos en catalán *podem* (*volem*) seguido de infinitivo, es mucho más probable que se trate del verbo *poder* (*voler*) que del verbo *podar* (*volar*); de nuevo, habría que usar una categoría gramatical especial, en este caso para los verbos modales.

11. Se pueden encontrar muchos ejemplos; por ejemplo, entre $L_1 =$ español y $L_2 =$ catalán, tenemos:
    - Ayer por la mañana vino tarde $\rightarrow$ *Ahir pel demà vi vesprada (Ahir de matí va venir tard).
    - Río porque no llegó a la meta $\rightarrow$ *Riu perquè no va arribar a la fique (Ric perquè no va arribar a la meta).
12. (c)
13. (b). Puede tener más de un árbol de análisis sintáctico (ambigüedad estructural).
14. (b)
15. (c)
16. (b)
17. (a). *Cantamos* puede ser presente o pasado.
18. (a). El pronombre átono que hace de sujeto en catalán de *fa*.
19. (a)
20. (c)
21. (b). El pronombre *li* puede tener los antecedentes *Mario* y *la seua mare*
22. (b)
23. (c)
24. (a)
25. (c)
26. (c)
27. (b)
28. (c)
29. (c)
30. (b)

Capítulo 8

Técnicas de traducción automática

Este capítulo describe las técnicas o, dicho de otro modo, las estrategias básicas usadas por los programas de traducción automática.

Hay dos grandes grupos de sistemas de traducción automática:

- Por un lado, los sistemas de traducción automática **basados en reglas** (en inglés, *rule-based machine translation*) o **basados en conocimiento** (en inglés *knowledge-based machine translation*). En estos sistemas, la información necesaria para realizar la traducción automática (diccionarios, reglas) la han escrito personas expertas de manera *deductiva*: es decir, han pensado en cómo automatizar el proceso de traducción automática y han deducido la información necesaria para realizarla. Entre estos sistemas, podemos distinguir:
  - Los sistemas de traducción automática *indirecta por transferencia* (apartado 8.3), entre los cuales, podemos distinguir, de acuerdo con el nivel de abstracción lingüística:
    - los sistemas de transferencia morfológica avanzada (a veces denominados de "traducción directa", a pesar de que no lo son; apartado 8.3.1);
    - los sistemas de transferencia sintáctica (apartado 8.3.3), y los
    - sistemas de transferencia semántica (apartado 8.3.5).
  - Los sistemas de traducción automática *por interlingua* (apartado 8.4).
- Por otro lado, los sistemas de traducción automática **basados en corpus** (en inglés *corpus-based machine translation*). En estos sistemas (véase el apartado 8.5), la información necesaria para realizar la traducción automática se *aprende* automáticamente de manera *inductiva* a partir de un *corpus* paralelo, es decir, de grandes cantidades de textos

y sus traducciones, previamente *segmentados* y *alineados* para poner cada oración de un texto en correspondencia con su traducción en el otro.[1] Los sistemas basados en corpus más comunes son los de *traducción automática estadística*, en los que lo que se aprende son modelos probabilísticos de traducción. Durante el decenio de 2010 se está investigando en sistemas que usan el llamado *aprendizaje profundo* (en inglés *deep learning*) basado en *redes neurales artificiales*, las cuales se basan vagamente en cómo funciona el cerebro humano.

8.1. Funcionamiento de la traducción automática

Los sistemas de traducción automática *reales*, es decir, los que se usan en la realidad, son el resultado de hacer una serie de aproximaciones sobre la traducción automática *ideal* para hacer el problema de la traducción computacionalmente abordable.

La mayoría de los sistemas de traducción automática, independientemente de si son basados en reglas o en corpus, adoptan la que podríamos denominar **aproximación oracional**, según la cual *traducir textos es traducir oraciones*. Esta aproximación excluye el tratamiento de algunos aspectos de la estructura del discurso.

Una vez hecha esta aproximación general, el resto de aproximaciones dependen del tipo de sistema de traducción automática. La mayoría de los sistemas de traducción automática basados en conocimiento abordan la traducción como una aplicación del *principio de composicionalidad semántica* (PCS, capítulo 7), el cual afirma que la interpretación (el significado) de una oración se construye composicionalmente a partir de las interpretaciones de las palabras, siguiendo los agrupamientos dictados por su árbol de análisis sintáctico, y también a la inversa, que las oraciones se pueden construir composicionalmente a partir de las interpretaciones (Tellier 2000). Traducir una oración, en este esquema, implica:

- hacer el análisis sintáctico completo,
- asignar interpretación a cada palabra,
- construir composicionalmente una interpretación de la oración,
- analizarla para obtener palabras y un árbol de análisis sintáctico para la lengua meta (LM), y
- generar una oración en LM a partir de las palabras y el árbol.

[1] Los textos del corpus de aprendizaje pueden además ser anotados o procesados con algún tipo de procesador lingüístico como por ejemplo un analizador morfológico o sintáctico.

Este es básicamente el *modus operandi* de los sistemas de *interlingua* (que se discutirán en la sección 8.4) y constituye la **aproximación composicional**. No olvidemos que esta descripción asume que la ambigüedad léxica (múltiples interpretaciones de las palabras) y estructural (más de un árbol de análisis sintáctico) ha sido idealmente resuelta.

Por otro lado, los sistemas de traducción automática indirecta por transferencia (que se discutirán en el apartado 8.3) son el resultado de una serie de aproximaciones (muchas de ellas inevitables) sobre un modelo ideal y teóricamente motivado basado en el *principio de composicionalidad semántica*. Estos sistemas también se pueden ver como el resultado de una serie de refinamientos inevitables sobre un sistema de traducción *palabra por palabra* (véase el apartado 8.2), es decir, como una serie de operaciones adicionales que se deben realizar además de ir sustituyendo cada palabra por un equivalente constante. Por ejemplo, para producir traducciones aceptables, rápidas e inteligibles, incluso entre lenguas muy parecidas, se tiene que añadir un procesamiento léxico robusto (por ejemplo, para tratar expresiones multipalabra o para elegir equivalentes adecuados para palabras léxicamente ambiguas) y un procesamiento estructural local o global que se base en reglas simples y bien formuladas para algunas transformaciones estructurales (reordenamientos, concordancia, etc.).

Como en el caso de los traductores profesionales, los sistemas de traducción automática no siempre necesitan *comprender* las frases en lengua origen (LO), es decir, construir una interpretación explícita. Esta noción, que puede parecer polémica, no lo es tanto: quién traduce como profesional un manual de mecánica del automóvil o un texto de física teórica lo puede hacer sin tener que entender completamente las disciplinas correspondientes. Los sistemas de *transferencia* sintáctica (véase el apartado 8.3.3) toman un atajo y van directamente del árbol de análisis sintáctico y las palabras en LO al árbol y las palabras en LM. Lo hacen aplicando transformaciones al árbol de análisis sintáctico (*transferencia estructural*) y sustituyendo las palabras (*transferencia léxica*), sin construir una representación explícita de la interpretación; esta es la **aproximación de transferencia**. Dependiendo del tipo concreto de sistema de traducción automática por transferencia, todavía son posibles más aproximaciones, como veremos más abajo.

Por último, los sistemas de traducción automática estadística (que se discutirán en el apartado 8.5) hacen **asunciones de independencia estadística** para poder modelar estadísticamente el proceso de traducción. Por ejemplo, el *modelo de traducción*, que se usa para estimar la probabilidad de que un segmento de texto en LM sea la traducción de un segmento de texto en LO, asume que la traducción de un segmento es independiente de la traducción del resto de segmentos de la oración; es decir, asume que no hay que tener en cuenta el contexto para traducir cada uno de los segmentos de la oración en LO.

8.2. Traducción directa y traducción indirecta

Las estrategias de traducción automática se pueden dividir en dos grandes grupos, las *directas* y las *indirectas*. La estrategia *directa* se denomina así porque la traducción de una frase se produce directamente, sin que se genere una representación intermedia de la frase; a veces también se la suele denominar vagamente traducción *palabra por palabra*. La estrategia *indirecta* produce, a partir de la frase en LO, una representación intermedia de la frase que después se usa para traducirla. Veremos más adelante de qué naturaleza son estas representaciones intermedias.

Una formalización posible de la traducción automática directa más sencilla posible es la traducción *palabra por palabra*: el sistema lee el texto original palabra a palabra de izquierda a derecha,[2] sustituye cada palabra original por un equivalente fijo (de una palabra, de más palabras, o incluso de cero palabras) en LM sin tener en cuenta el contexto y escribe las palabras una a una y en el mismo orden en el texto meta. Por ejemplo, si la frase tiene N palabras,

$$m_1\ m_2\ m_3\ \cdots\ m_N$$

la traducción *palabra por palabra* es

$$T(m_1)\ T(m_2)\ T(m_3)\ \cdots\ T(m_N)$$

donde $T(m)$ representa el equivalente fijo de la palabra m en la LM, que puede tener cero, uno o más palabras. Por ejemplo, la traducción *palabra por palabra* al inglés de la frase *Este ejercicio práctico es muy sencillo*, con $m_1 =$ *Este*, $m_2 =$ *ejercicio*, etc., podría ser *This exercise practical it is very simple* (incorrecta), donde, por ejemplo, $T(m_4) = T(es) =$ *it is*.

Ningún sistema real de traducción automática usa este modelo tan rudimentario de traducción, puesto que es incapaz de producir traducciones automáticas útiles, ni siquiera para lenguas muy similares: todos los sistemas van más allá y realizan operaciones adicionales. Por eso mismo, el modelo *palabra por palabra* se puede usar como modelo de referencia o *modelo cero* a la hora de estudiar qué más hacen los sistemas existentes, o qué más se debe hacer para producir traducciones automáticas útiles para un par de lenguas.

8.3. Traducción indirecta por transferencia

Muchos de los sistemas indirectos basados en reglas son sistemas de *transferencia*. Un *sistema de traducción automática indirecta por transferencia*,

[2] Se asume que el texto está ya segmentado en palabras, operación que puede no ser trivial en algunos idiomas como por ejemplo el japonés o el chino, que no usan espacios en blanco para separar las palabras.

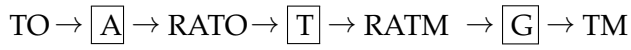


Figura 8.1: Fases de análisis (A), transferencia (T) y generación (G) en un sistema de traducción indirecta por transferencia (TO = texto origen; RATO = representación abstracta del texto origen; RATM = representación abstracta del texto meta; TM = texto meta).

o, abreviadamente, *sistema de transferencia* es el que hace las traducciones en tres fases bien diferenciadas llamadas *análisis*, *transferencia* y *generación*; cada una de estas fases es realizada por un módulo (un subprograma) del sistema:

- El módulo de *análisis* es un módulo monolingüe que produce, a partir de la frase en LO, una *representación abstracta del texto origen* (RATO). En la RATO se eliminan todos los detalles de la frase en LO que no se consideran relevantes para la traducción y se destacan aquellas características y relaciones que sí lo son. Por ejemplo, podría convenir que las frases inglesas "Sam gave a book to Leslie" y "Sam gave Leslie a book" (Arnold et al. 1993) tuvieron la misma RATO.

- El módulo de *transferencia* es un módulo bilingüe que lee la RATO y genera otra representación abstracta similar, pero para la LM, la *representación abstracta del texto meta* (RATM).

- El módulo de *generación* (o, menos comúnmente, *síntesis*) genera a partir de la RATM un texto *concreto*: la traducción en bruto.

Estas tres fases se esquematizan en la figura 8.1.

Las representaciones abstractas *acercan* las dos lenguas eliminando algunos detalles específicos y destacando características generales que pueden así ser tratadas más fácilmente por el módulo de transferencia. Por ejemplo, los adjetivos catalanes van normalmente detrás de los sustantivos mientras que los ingleses van normalmente delante; traducir comporta, por lo tanto, cambiar los adjetivos de posición. Pero, para ello, hay que identificar qué palabras son adjetivos y sustantivos: la fase de análisis se puede encargar de esta tarea para que la fase de transferencia pueda aplicar reglas generales sin preocuparse de cuáles son los adjetivos o los sustantivos concretos.

La arquitectura de transferencia es el modelo estándar para la traducción automática basada en reglas o conocimiento contemporánea, y lo ha sido durante muchos años (Arnold et al. 1993).

Los sistemas de transferencia se clasifican según la naturaleza de las representaciones abstractas que utilizan: se puede hablar, en orden de profundidad del análisis, de sistemas de *transferencia morfológica*, de *transferencia sintáctica* o de *transferencia semántica*. La elección de la profundidad del análisis se debe fundamentar en la naturaleza y la profundidad de las divergencias de traducción (Vandooren 1993) entre las lenguas implicadas.

La arquitectura de transferencia tiene tres características interesantes que merecen ser mencionadas:

Funcionamiento como cadena de montaje. El sistema de transferencia funciona como una *cadena de montaje*: como los tres módulos trabajan de izquierda a derecha y en una única pasada, no hace falta que un módulo espere a que el anterior acabe con el texto: pueden trabajar paralelamente; ello hace que los sistemas de esta naturaleza sean muy rápidos.

Modularidad. La división en *módulos* o etapas bien diferenciadas (análisis, transferencia y generación) permite su reutilización. Por ejemplo, si hemos construido un sistema de transferencia que traduce del inglés al español:

$$\text{en} \rightarrow \boxed{\text{A}_{\text{en}}} \rightarrow \boxed{\text{T}_{\text{en}\rightarrow\text{es}}} \rightarrow \boxed{\text{G}_{\text{es}}} \rightarrow \text{es}$$

podemos aprovechar el módulo de análisis del inglés (A_{en}) para construir un sistema del inglés al catalán:

$$\text{en} \rightarrow \boxed{\text{A}_{\text{en}}} \rightarrow \boxed{\text{T}_{\text{en}\rightarrow\text{ca}}} \rightarrow \boxed{\text{G}_{\text{ca}}} \rightarrow \text{ca}$$

o usar el módulo de generación del español G_{es} para construir un sistema del neerlandés al español:

$$\text{en} \rightarrow \boxed{\text{A}_{\text{nl}}} \rightarrow \boxed{\text{T}_{\text{nl}\rightarrow\text{es}}} \rightarrow \boxed{\text{G}_{\text{es}}} \rightarrow \text{es}$$

Reversibilidad parcial. Si se separan los datos lingüísticos usados por cada uno de los tres módulos, es posible una *reversibilidad parcial*. Si en el sistema inglés–español de arriba separamos los *datos lingüísticos* de cada uno de los tres módulos del *software* que procesa estos datos (dA_{en}, $\text{dT}_{\text{en}\rightarrow\text{es}}$, dG_{es}) podemos definir un *motor* genérico de traducción (A, T, G) que vale para cualquier par de lenguas:

$$\text{en} \rightarrow \boxed{\begin{array}{l}\text{dA}_{\text{en}}\\ \text{A}\end{array}} \rightarrow \boxed{\begin{array}{l}\text{dT}_{\text{en}\rightarrow\text{es}}\\ \text{T}\end{array}} \rightarrow \boxed{\begin{array}{l}\text{dG}_{\text{es}}\\ \text{G}\end{array}} \rightarrow \text{es}$$

Si ahora queremos escribir el sistema de traducción inverso, español–inglés, es decir,

$$\text{es} \rightarrow \boxed{\begin{matrix}\text{dA}_{\text{es}}\\ \text{A}\end{matrix}} \rightarrow \boxed{\text{dT}_{\text{es}\rightarrow\text{en}}\\ \text{T}} \rightarrow \boxed{\begin{matrix}\text{dG}_{\text{en}}\\ \text{G}\end{matrix}} \rightarrow \text{en}$$

podríamos aprovechar que hay grandes parecidos entre los datos lingüísticos de este sistema y los del sistema anterior:

- los datos que necesitamos para el análisis del español dA_{es} son muy similares a los datos de generación del español dG_{es} del sistema existente: para analizar el español podemos reciclar una buena parte de los datos que se usaban para generarlo en el sistema anterior;
- los datos que necesitamos para la generación del inglés dG_{en} son muy similares a los datos de análisis del inglés dA_{en} del sistema existente: para generar el inglés podemos reciclar una buena parte de los datos que se usaban para analizarlo en el sistema anterior;
- los datos de transferencia del español al inglés $\text{dT}_{\text{es}\rightarrow\text{en}}$ son muy similares a los datos de transferencia del inglés al español $\text{dT}_{\text{en}\rightarrow\text{en}}$ del sistema existente: para transferir de español a inglés podemos usar una buena parte de los datos que se usaban para transferir desde el inglés al español en el sistema anterior (por ejemplo, podemos "dar la vuelta" a los diccionarios bilingües y aprovechar muchas entradas).

8.3.1. Sistemas de transferencia morfológica avanzada

En los sistemas de transferencia *morfológica avanzada* —también llamados sistemas de *transferencia sintáctica parcial* o *transformers* (Arnold et al. 1994, 4.2)— la fase de *análisis* analiza morfológicamente las palabras de la frase y las desambigua en caso de ambigüedad léxica categorial pero sólo identifica las relaciones (sintácticas) entre ellas usando patrones muy sencillos.[3] La sección 8.3.2 da más detalles sobre los procesos y los métodos de análisis y generación morfológicos.

De hecho, los sistemas de transferencia morfológica avanzada se pueden ver como el resultado de hacer una tercera aproximación, añadida a las dos (*aproximación oracional* y *aproximación de transferencia*) discutidas en el apartado 8.1:

> **Aproximación de transferencia parcial:** Cuando las lenguas involucradas no son demasiado diferentes sintácticamente (por ejemplo, cuando están emparentadas), no es necesario realizar

[3] Algunos sistemas de transferencia morfológica avanzada disponibles en Internet son: SDL Transcend (`http://www.freetranslation.com`) Reverso (`http://www.reverso.net`), y Apertium (`http://www.apertium.org`).

el análisis sintáctico completo: la transferencia léxica es completa pero la transferencia estructural es parcial y local y sólo se hace donde es necesaria.

La fase de *transferencia* puede consistir en un reordenamiento local (*transferencia estructural*) de algunas secuencias de palabras (por ejemplo, cuando se traduce del inglés al español, los pares adjetivo–sustantivo se podrían reordenar a sustantivo–adjetivo) y en la conversión de las formas léxicas de la LO en las correspondientes de la LM mediante el uso de un diccionario bilingüe (*transferencia léxica*).

La *fase de generación* podría efectuar la sustitución de las formas léxicas de la LM por las correspondientes formas superficiales.

Como los sistemas de transferencia morfológica avanzada no identifican realmente las relaciones sintácticas entre las palabras de la frase en la lengua origen, para hacer los reordenamientos tienen que identificar las secuencias de palabras que necesitan ser reordenadas. La capacidad de un sistema de transferencia morfológica para producir traducciones aceptables dependerá de su capacidad para detectar secuencias de palabras que se correspondan con los sintagmas que necesitan ser reordenados. Imaginemos que queremos traducir del inglés al español y hemos decidido que se deben usar estas reglas de reordenamiento:

R_1 (en) **adj subst** $\rightarrow$ (es) **subst adj**

R_2 (en) **subst**$_1$ **subst**$_2$ $\rightarrow$ (es) **subst**$_2$ **prep_de subst** $_1$

Por ejemplo, la regla R_1 reordenaría "tall driver" y produciría "conductor alto" y la regla R_2 reordenaría "truck driver" y produciría "conductor de camión".

Ahora, pensemos qué le sucedería a "tall truck driver". Si se aplica primero la regla R_1 a "tall truck" ya no podemos aplicar la R_2. Si se aplica primero la R_2 y después la R_1, se obtiene la traducción correcta: "conductor alto de camión". Cuando tenemos más de una regla, no sabemos en qué orden hay que aplicarlas. Si tenemos "tall gasoline truck driver" ("conductor alto de camión de gasolina"), no hay ningún orden de aplicación de R_2 i R_1 que dé una traducción aceptable. Esto sugiere la necesidad de una nueva regla que detecte y reordene el patrón largo adjetivo–sustantivo–sustantivo–sustantivo, por ejemplo:

R_3 (en) **adj subst**$_1$ **subst**$_2$ **subst**$_3$ $\rightarrow$ (es) **subst**$_3$ **adj prep_de subst**$_2$ **prep_de subst**$_1$

Esta regla podría reordenar correctamente esta secuencia de cuatro palabras. Como se puede observar, las reglas de reordenamiento intentan descubrir unidades sintácticas (sintagmas) usando un número limitado de patrones que representan las secuencias de palabras que pueden formar estas

unidades; el problema es que los sintagmas pueden ser, en principio, indefinidamente largos,[4] y el conjunto de reglas de reordenamiento tiene que ser forzosamente limitado.

Queda, además, por determinar, en los casos en los que se puede aplicar más de una regla, cuál tiene que aplicarse antes; en una oración larga y con muchas reglas disponibles, esto puede ser un problema grave. Una técnica observada en algunos programas es la siguiente: (a) los reordenamientos se van aplicando según se recorre la frase de izquierda a derecha; (b) los reordenamientos de secuencias más largas tienen prioridad, y (c) las palabras afectadas por un reordenamiento no vuelven a estar involucradas en ningún otro reordenamiento. Así sólo se visita cada palabra de la frase una vez. Si no se puede aplicar un reordenamiento a la primera palabra pendiente de procesar, se traduce aisladamente y se continúa con la siguiente palabra.

Para poder traducir *tall truck driver* siguiendo este esquema, haría falta una regla que combinara R_1 y R_2:

R_4 (en) **adj subst**$_1$ **subst**$_2$ $\rightarrow$ (es) **subst**$_2$ **adj prep_de subst**$_1$

La identificación de patrones de categorías morfológicas que se correspondan con los sintagmas más frecuentes puede servir, además de para hacer reordenamientos, para resolver la concordancia de número y género. Por ejemplo, si usamos el patrón sustantivo–adjetivo para identificar una clase de sintagmas nominales sencillos, podemos hacer que la traducción correcta al catalán del sintagma nominal español *postre buenísimo* sea *postres boníssimes*, puesto que el género y el número del adjetivo que modifica a un sustantivo tiene que concordar y el sustantivo español *postre* (masculino singular) se corresponde con el sustantivo catalán *postres* (femenino plural). Como una vez identificada la clase de sintagma queda claro que el núcleo es el primer elemento (el sustantivo), ya se puede propagar el género y el número del primer elemento al segundo (el adjetivo):

(es) **subst adj** $\rightarrow$ (ca) **subst adj**
asigna género meta: **subst** $\rightarrow$ **adj**
asigna número meta: **subst** $\rightarrow$ **adj**

El reordenamiento y la concordancia se pueden combinar en la misma regla; por ejemplo, cuando se traduce del inglés al español la secuencia adjetivo–sustantivo, la regla podría tener esta forma:

[4] En la gramática de la lengua, si una regla que extiende un sintagma se puede aplicar repetidamente a un determinado tipo de sintagma, este sintagma se puede alargar indefinidamente. Un ejemplo clásico de esto lo dan las oraciones adjetivas de relativo; la serie de sintagmas nominales "el coche", "el coche que trajo el hombre", "el coche que trajo el hombre que vino del pueblo", "el coche que trajo el hombre que vino del pueblo que visitamos durante el viaje", etc., demuestra que no hay límites a la longitud de un sintagma (nominal, en este caso).

(en) **adj subst** → (es) **subst adj**
asigna género meta: **subst** → **adj**
asigna número meta: **subst** → **adj**

Las figuras 8.2, 8.3 y 8.4 ilustran el funcionamiento de las fases de análisis, transferencia y generación, respectivamente, de un sistema de transferencia morfológica avanzada (es decir, con reconocimiento de patrones sencillos que representan sintagmas) del inglés al español.

La estrategia de *transferencia morfológica avanzada* se puede ver como una formalización de la estrategia mal llamada *directa* que se usaba en los programas de traducción automática de primera generación (años cincuenta y sesenta del siglo pasado): análisis morfológico rudimentario, consulta del diccionario bilingüe, y ajustes locales como por ejemplo reordenamientos. Si pidiéramos a una persona no experta que diseñara un sistema de traducción automática, el primer diseño no sería muy diferente del que se ha descrito. El resultado (Hutchins y Somers 1992, sección 4.2), a veces erróneamente denominado *traducción directa* en vista de su simplicidad, es lo que se podría esperar "de una persona que contara únicamente con un diccionario bilingüe muy barato y con un conocimiento muy rudimentario de la gramática de la lengua meta", con "errores frecuentes de naturaleza léxica en la traducción y estructuras sintácticas inadecuadas" que reflejan "las estructuras propias de la lengua de origen".

8.3.2. Análisis y generación morfológicas

El *análisis morfológico* es el proceso que determina, para cada palabra de un texto (forma *superficial* o *flexionada*) una o varias *formas léxicas*, consistentes en una *forma canónica* o *lema* e información sobre la categoría léxica de la palabra y la flexión (véase la sección 7.2.1). La *generación* morfológica hace la operación inversa. Por ejemplo, el análisis morfológico de la forma superficial catalana *vius* (ambigua) daría como resultado dos formas léxicas: *viure*, verbo, presente de indicativo, 2ª persona del singular y *viu*, adjetivo, masculino, plural, donde los lemas son, respectivamente, *viure* y *viu*.

Un analizador morfológico, por lo tanto, debe tener la siguiente información sobre la lengua de los textos que se tienen que analizar: el vocabulario o conjunto de lemas, los paradigmas de flexión, y la correspondencia entre lemas y paradigmas.

A veces el análisis morfológico puede ser más difícil de lo que parece, como por ejemplo, en el caso de la morfología verbal de las lenguas románicas. Fijaos en el imperativo español *demos*: si va seguido del pronombre enclítico *le*, forma con este una única palabra y recibe un nuevo acento ortográfico: *démosle*; si el pronombre es *nos*, además se pierde una consonante: *démonos*; con dos pronombres, puede ser *démonoslos*, etc. Otras veces se da el problema de la ambigüedad léxica categorial (véase el apar-

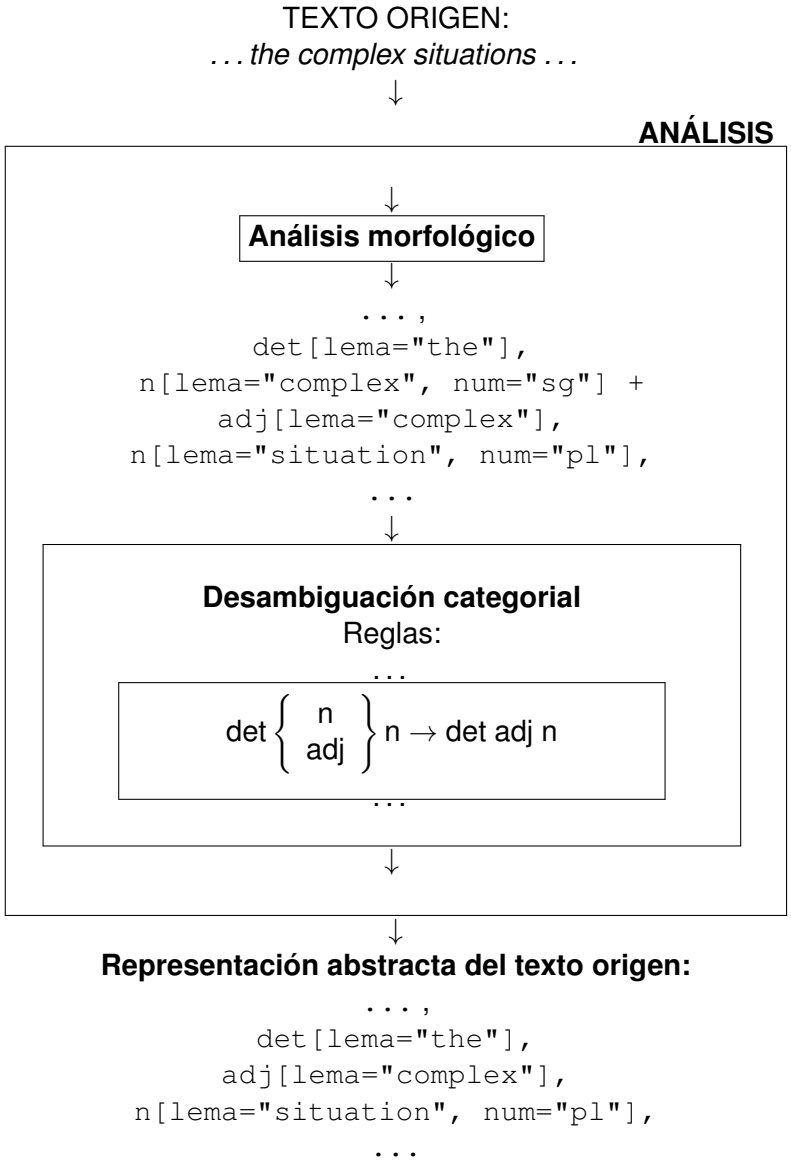


Figura 8.2: Fase de análisis de un sistema sencillo de transferencia morfológica avanzada. El segmento de texto *the complex situations* contiene la palabra *complex* que puede ser un adjetivo (adj) o un sustantivo (n); entre las reglas del desambiguador categorial hay una regla que en este caso asigna la categoría de adjetivo cuando va entre un determinante y un sustantivo.

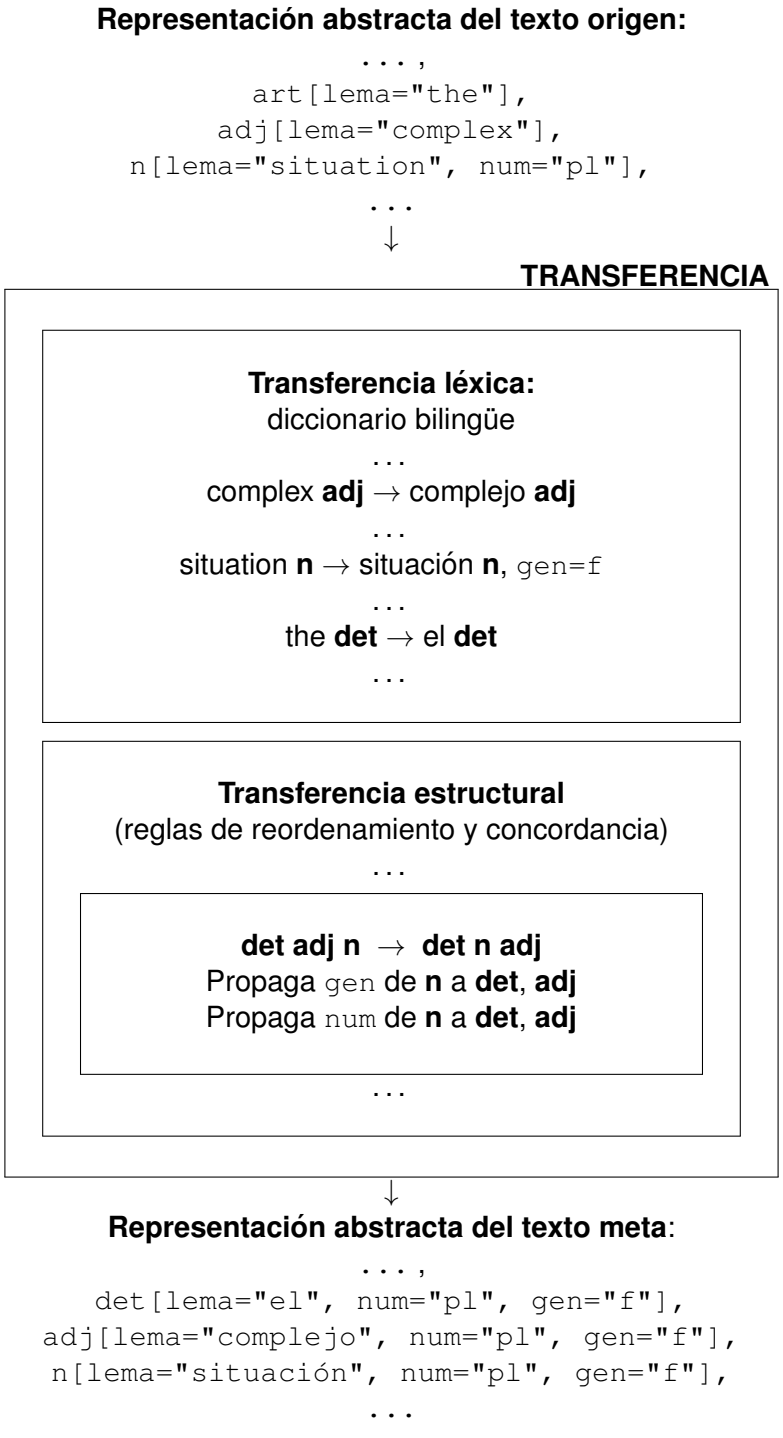


Figura 8.3: Fase de transferencia de un sistema sencillo de transferencia morfológica avanzada con reglas de transferencia estructural que realizan el reordenamiento y aseguran la concordancia, entre las cuales hay una regla para reordenar la secuencia determinante–adjetivo–sustantivo y asegurar la concordancia de género y número en la lengua meta.

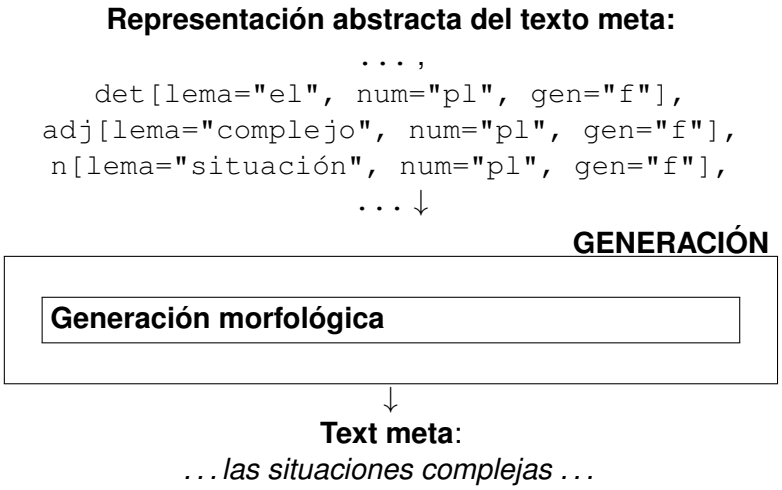


Figura 8.4: Fase de generación de un sistema sencillo de transferencia morfológica avanzada.

tado 7.2.1): es decir, la palabra puede pertenecer a dos categorías léxicas diferentes y hay que usar información sobre las categorías morfológicas de las palabras anteriores y posteriores (en ausencia de información sintáctica) para deshacer la ambigüedad.

Para saber más sobre el análisis morfológico

Cuando los mecanismos de flexión de las lenguas son, como en la mayor parte de las lenguas indoeuropeas, por modificación de las terminaciones (*desinencias*) de las palabras, un método atractivo consiste en procesar la forma superficial letra a letra de izquierda a derecha y producir la forma léxica incrementalmente, añadiendo a cada paso más información. Se asume que las primeras letras de la palabra son la *raíz* y, por lo tanto, determinan el lema, y que las últimas letras determinan la forma gramatical. Imaginemos que seguimos este método con la palabra catalana *angoixaven*:

- Cuando sólo hemos visto *ang-* hay todavía muchas posibilidades: puede ser, entre otros, cualquier forma de las palabras *angina*, *angle*, *anglés*, *angoixa*, *angoixar*, *angost*, *anguila* y *angula*. Como máximo, podemos decir que el lema empieza por `ang`.
- Cuando hemos leído *ango-* el abanico de posibilidades se hace reduce: puede ser una forma de *angoixa*, de *angoixar* o de *angost*. Ya podemos decir que el lema empieza por `ango`.
- Cuando hemos leído *angoi-* ya sabemos que el lema es *angoixar* o *angoixa*, es decir, que comienza por `angoixa`.

- Leer *angoix-* o *angoixa-* no nos permite determinar con seguridad más información sobre la forma léxica; en el caso de *angoixa-* se pueden descartar algunas formas del verbo angoixar como *angoixem* o *angoixí*, pero todavía nos quedan muchas. Todavía puede ser nombre o verbo.
- Cuando hemos visto *angoixav-* ya podemos decir que el lema es *angoixar*, que se trata de un verbo, y que estamos, con tota seguridad, ante una forma del imperfecto de indicativo. El analizador ya puede decirnos `angoixar<verb><imp>`.
- Después de ver *angoixave-* todavía no sabemos la persona del verbo (puede ser la segunda del singular o la tercera del plural).
- Finalmente, cuando vemos *angoixaven* ya sabemos que es la tercera del plural. El analizador produce: `angoixar<verb><imp><3p><pl>`.

El proceso se puede resumir en el alineamiento siguiente:

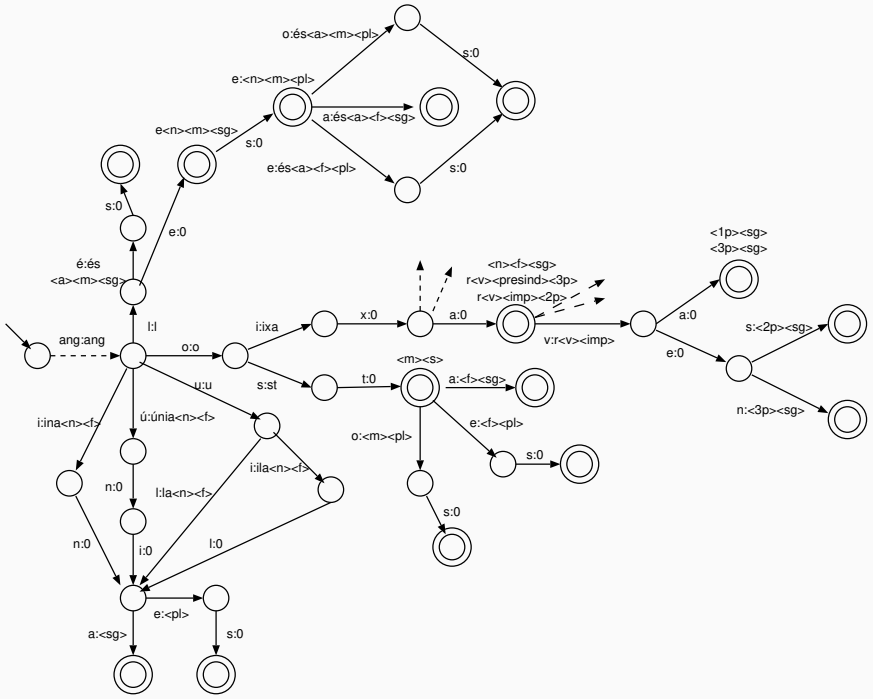
```
a n g o i   x a v           e n
a n g o ixa - - r<verb><imp> - <3p><pl>
```

Si la forma superficial hubiera sido *angostes*, todo el proceso hasta *ango-* habría sido idéntico. En el caso de *angoixem*, el proceso habría sido idéntico hasta *angoixe-*:

```
a n g o i   x e m
a n g o ixa - - r<verb><presind><1p><pl>
              r<verb><pressubj><1p><pl>
              r<verb><imp><1p><pl>
```

En este último caso, la palabra tiene tres análisis morfológicos diferentes. Por otro lado, la parte final del proceso de *angoixaven* y de *cantaven* sería muy similar, ya que los dos verbos se conjugan según el mismo paradigma.

Todo esto permite representar el analizador morfológico como lo que los matemáticos llaman *grafo dirigido acíclico* (GDA). Un *grafo dirigido* tiene dos partes: un conjunto de nodos, nudos o *vértices* (representados gráficamente como puntos o círculos) y un conjunto de flechas cada una de les cuales va de un vértice a otro.



El grafo es *acíclico* si, siguiendo las flechas, no se puede pasar dos veces por el mismo *vértice*. El GDA de un analizador morfológico tiene un vértice inicial único (indicado con una flecha que no viene de ningún lugar y del cual sólo pueden salir flechas) que representa el estado inicial del analizador antes de empezar a leer una palabra. Los otros vértices representan el estado del analizador después de haber leído una o más letras. Las flechas que salen de un vértice cualquiera del grafo tienen dos etiquetas separadas por "`:`". La primera (de entrada) indica la letra que se lee; la segunda (de salida) indica lo que se tiene que producir (cero, uno o más símbolos). Los estados finales (representados con dos círculos concéntricos) indican estados en los cuales el analizador determina que ha leído una forma superficial completa si no quedan más letras para leer; en algunos estados finales se escriben símbolos adicionales para completar el análisis y, en el caso de un homógrafo (ambigüedad léxica), todas las opciones. El analizador lee la forma superficial letra a letra, va de estado en estado, y va produciendo la forma léxica, hasta que lee toda la palabra; si llega a un estado final, acepta la palabra y devuelve la forma léxica. En la figura anterior se puede ver una parte del analizador morfológico, correspondiendo a las palabras que empiezan por *ang-*. En informática teórica, las máquinas idealizadas que representan esta clase de GDA se llaman *transductores de estados finitos p-subsecuenciales sin ciclos*, donde *p* indica el número de opciones posibles en los estados finales. Los *generadores morfológicos* se pueden organizar de forma muy similar: leen símbolo a símbolo la forma léxica y producen la forma superficial.

8.3.3. Sistemas de transferencia sintáctica

En estos sistemas, la representación abstracta (RALO) que se obtiene en el análisis incluye un árbol de análisis sintáctico de la frase en LO (o una entidad equivalente), que describe las relaciones sintácticas existentes entre

las partes de la frase, además de la información morfológica necesaria para hacer la traducción. Es decir, se hace un análisis morfológico y un análisis sintáctico (en inglés *parsing*) de la frase en LO. El análisis sintáctico se explica con más detalle en el apartado 8.3.4. En la fase de transferencia, se aplican reglas de *transferencia estructural* que transforman la representación sintáctica de la frase de entrada (la RALO) en una representación sintáctica de la traducción (la RALM) usando reglas de transformación de estructuras, y de *transferencia léxica* que traducen las palabras en LO a palabras en LM usando un diccionario bilingüe. Finalmente, la fase de generación transforma esta representación en la frase en LM.

La estrategia de transferencia sintáctica resuelve una buena parte de los problemas de los sistemas directos y de los de transferencia morfológica, ya que es capaz de determinar la extensión y la estructura de cada uno de los sintagmas de la frase en LO y manipular cada sintagma como una unidad, independientemente de la estructura o de la longitud. De hecho, como ya se ha dicho, los sintagmas *tienen estructura*; es decir, las relaciones entre los elementos de un sintagma no son puramente lineales, sino jerárquicas: los sintagmas están hechos de sintagmas. Los sistemas de transferencia morfológica no pueden tener en cuenta esta estructura interna, y, como resultado, necesitan muchísimas reglas para reordenar adecuadamente las palabras de las frases.[5]

Imaginemos el sintagma nominal siguiente en inglés: *The old professor's smart girlfriend's red car*; el sintagma es demasiado largo para la mayor parte de los programas de transferencia morfológica, ya que involucra una secuencia demasiado larga de reordenamiento. Si, en cambio, tenemos un sistema de TA por transferencia sintáctica y suponemos que la gramática contiene las reglas siguientes:[6]

$\mathrm{SN} \rightarrow \mathrm{SG\ SN}$
$\mathrm{SN} \rightarrow \mathrm{A\ SN}$
$\mathrm{SN} \rightarrow \mathrm{N}$
$\mathrm{SG} \rightarrow \mathrm{SN\ GS}$

donde SN es un sintagma nominal, SG un "sintagma de genitivo", A un adjetivo, N un sustantivo y GS la partícula de genitivo sajón (*'s*, *'*). La estructura de este sintagma nominal se podría representar, usando un árbol de análisis sintáctico (sin tener en cuenta los determinantes, para simplificar) como se ve en la figura 8.5. Un sistema de transferencia sintáctica del

[5] Los sistemas de transferencia morfológica por reordenamiento de patrones asumen que una oración es una secuencia lineal de sintagmas de estructura lineal; este modelo de la sintaxis de una oración puede ser muy limitado en muchas aplicaciones de traducción automática.

[6] La sintaxis generativa actual (véase Chomsky (1996), Ramos (1992)) predice muchos aspectos de las lenguas naturales postulando la existencia de un conjunto de reglas universales muy sencillas o *principios* con variaciones que en cada lengua están determinadas por *parámetros*; la gramática que se considera en esta discusión no es, en principio, la postulada por esta formulación, sino una adecuada para la tarea concreta.

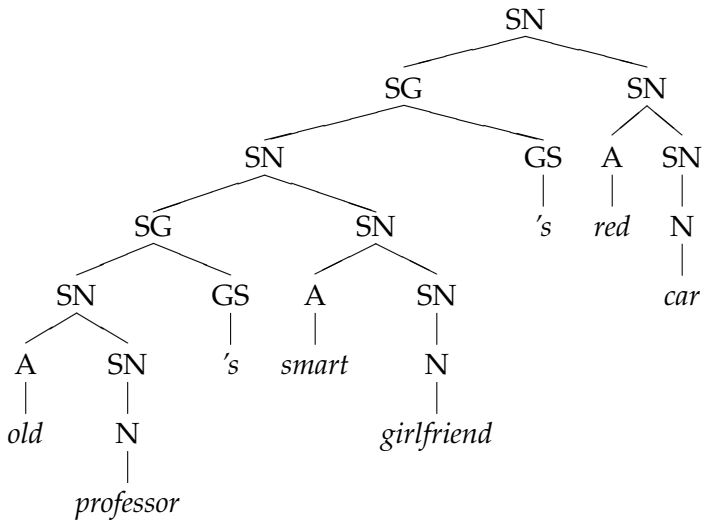
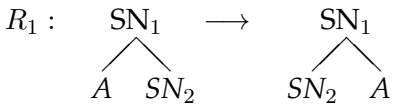
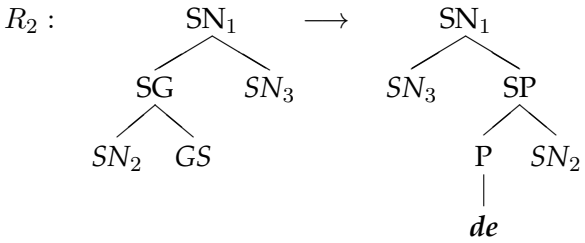


Figura 8.5: Árbol de análisis sintáctico del sintagma nominal *The old professor's smart girlfriend's red car*.

inglés al español podría usar dos reglas para transformar subárboles (partes del árbol), una para mover los adjetivos:[7]



y otra para reordenar los sintagmas nominales que contienen un genitivo sajón:



La aplicación de la regla R_2 al SN en la raíz del árbol de análisis sintáctico de la frase da como resultado el árbol que se ve en la figura 8.6. Después sería necesario aplicar la regla R_1 al SN que genera *red car*, después la regla

[7]Evidentemente, no siempre se deben mover los adjetivos; la traducción correcta de *the last car* es *el último coche*, sin cambiar el orden. Un sistema real de transferencia sintáctica tendría que considerar estos casos de manera especial.

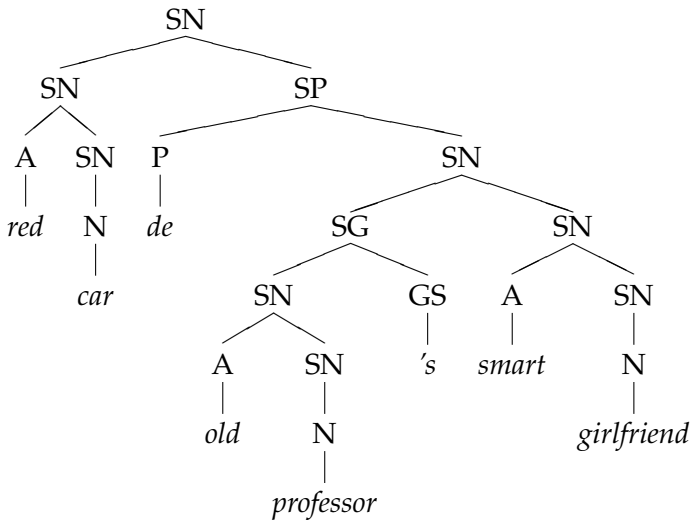


Figura 8.6: Árbol de análisis sintáctico del sintagma *The old professor's smart girlfriend's red car* después de aplicarle la regla R_2 al SN principal o raíz (véase el texto y la figura 8.5).

R_2 al SN que genera *old professor's smart girlfriend*, etc. El resultado final se muestra en la figura 8.7 y se corresponde con el árbol de análisis sintáctico de la traducción, *El coche rojo de la amiga inteligente del profesor viejo*, que se podría generar directamente a partir del árbol.

Cuando se traduce entre lenguas emparentadas, como por ejemplo del español al catalán, pocas veces se dan situaciones como esta, ya que, en general, el orden de las palabras no suele cambiar tan radicalmente; una construcción española que sí que podría requerir la identificación y el desplazamiento de un sintagma nominal completo para traducirlo al catalán sería la construcción de relativo posesivo con *cuyo*, ya que el catalán no tiene. Una posible solución usa frases preposicionales del tipo de *del qual* postpuestas a la traducción del sintagma nominal que sigue a *cuyo*. Fijaos en estos dos ejemplos, donde se ha hecho un análisis sintáctico parcial[8] para marcar los sintagmas nominales con corchetes:

$$[_{SN} \text{ las hijas }][_{SN} \text{ cuyo } [_{SN} \text{ padre }]] \rightarrow$$
$$[_{SN} \text{ les filles }][_{SN} \text{ el pare } [_{SP} \text{ de les quals }]] \ldots]]$$

$$[_{SN} \text{ la comunidad }][_{SN} \text{ cuyas } [_{SN} \text{ señas de identidad básicas }]] \rightarrow$$
$$[_{SN} \text{ la comunitat }][_{SN} \text{ les senyes d'identitat bàsiques } [_{SP} \text{ de la qual }]] \ldots]]$$

[8] Es decir, sin construir el árbol completo.

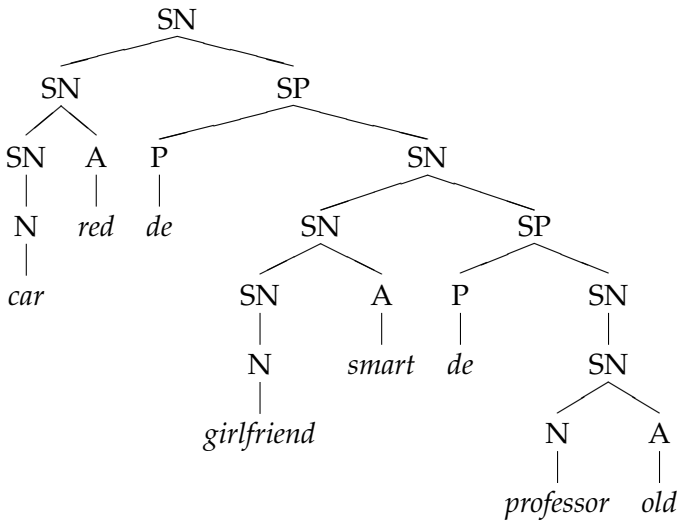


Figura 8.7: Árbol de análisis sintáctico del sintagma *The old professor's smart girlfriend's red car* después de haber hecho todos los reordenamientos posibles con las reglas R_1 y R_2 (véase el texto y las figuras 8.5 y 8.6).

El sintagma nominal que sigue a *cuyo* (y concuerda en género y número) puede ser, en principio, indefinidamente largo; hay que dentificarlo correctamente, moverlo como una unidad y añadirle la frase relativa *del qual* de manera que concuerde ahora (en género y número) con el antecedente (el sintagma nominal anterior al *cuyo*). Estas operaciones se pueden resolver de manera natural en un sistema de transferencia sintáctica.

8.3.4. Análisis sintáctico

El análisis sintáctico presupone la existencia de una *gramática* de la LO, es decir, de un conjunto de reglas que describen cómo se construyen, sintagma por sintagma, las oraciones válidas de la LO. Debe tenerse en cuenta que escribir una gramática que cubra completamente todas las posibles frases sintácticamente correctas de una lengua es una tarea que está muy lejos de ser trivial; por ese motivo, los analizadores deben ser *robustos* y ser capaces de entregar análisis parciales o incompletos para oraciones que no estaban previstas. El analizador sintáctico actúa después del analizador morfológico, y obtiene el *árbol de análisis sintáctico* (o los árboles, si hay más de uno) a partir de la secuencia de categorías léxicas de la frase; cada árbol indica una posible combinación y orden de aplicación de reglas que da lugar a la frase en cuestión. Las reglas se suelen corresponder normalmente

con los subárboles básicos con los cuales se construyen los árboles de análisis sintáctico de todas las frases sintácticamente aceptables; es decir, estas reglas especifican cómo se puede construir un sintagma o *constituyente* a partir de otros sintagmas y de categorías léxicas.

Para saber más sobre el análisis sintáctico

Los algoritmos de análisis sintáctico pueden ser *ascendentes* (inglés *bottom-up*) cuando construyen el árbol empezando por las hojas —las categorías léxicas de cada palabra— y acabando por la raíz —que corresponde a la oración completa—, o *descendentes* (inglés *top-down*), en caso contrario (recuérdese que los árboles de análisis sintáctico están "cabeza abajo": la raíz está arriba y las hojas, abajo). Si la frase es estructuralmente ambigua (véase el apartado 7.2.2), tiene más de un árbol de análisis sintáctico: algunos analizadores producen todos los árboles posibles; otros, eligen uno (usando alguna estrategia de desambiguación sintáctica como las descritas en 7.2.4).

Para que sea práctico, el algoritmo de análisis sintáctico debe ser rápido y eficiente; por ejemplo, es conveniente que funcione de manera que pueda construir progresivamente los árboles leyendo la oración de izquierda a derecha, ya que así (a) puede comenzar a trabajar antes de que el analizador morfológico haya analizado toda la oración y (b) puede proveer al módulo de transferencia con análisis parciales que le pueden servir para ir preparando la traducción parcial de las partes ya analizadas.

Como ejemplo, describiremos un tipo bastante extendido de analizador ascendente, denominado usualmente GLR (*generalized LR* o LR generalizado, donde LR es la abreviación de *left-to-right, rightmost derivation*, "de izquierda a derecha y con derivación por la derecha"). Los analizadores GLR leen las categorías léxicas de izquierda a derecha, las *desplazan* a un tipo de memoria especial denominada pila (es decir, las *apilan*) y cuando la cima de la pila contiene elementos que según la gramática y el contexto inmediato posterior se pueden agrupar en un subárbol, los desapila, los *reduce* a un subárbol, y deja el subárbol en la cima de la pila. Para saber cuándo tiene que desplazar una categoría léxica a la pila o cuándo y cómo tiene que reducir la cima de la pila, tiene en cuenta una o más categorías léxicas de las que está a punto de leer y uno o más elementos de la cima de la pila, y hace la acción que le indica una *tabla de análisis sintáctico* que se construye a partir de la gramática que se haya propuesto para la lengua origen y que el analizador consulta en cada paso del análisis. El uso de la tabla permite construir el programa analizador independientemente de la gramática concreta: si cambia la gramática, sólo cambia la tabla.

Un ejemplo servirá para ilustrar todos estos conceptos. Imaginemos la siguiente gramática simplificada que acepta un buen número de oraciones simples en español:

$$
\begin{array}{rcl}
O & \longrightarrow & SN\ SV \\
SN & \longrightarrow & \textbf{det}\ \bar{N} \\
SN & \longrightarrow & \bar{N} \\
SN & \longrightarrow & SN\ SP \\
\bar{N} & \longrightarrow & \textbf{n adj} \\
\bar{N} & \longrightarrow & \textbf{n} \\
SV & \longrightarrow & \textbf{v} \\
SV & \longrightarrow & \textbf{v}\ SN \\
SV & \longrightarrow & SV\ SP \\
SP & \longrightarrow & \textbf{prep}\ SN
\end{array}
$$

La gramática es ambigua, es decir, capaz de generar dos árboles de análisis sintáctico

para oraciones como por ejemplo *El hombre lleva la llave del armario grande*.

Usando un algoritmo estándar que no viene al caso detallar aquí, la gramática se transforma en la tabla de análisis sintáctico correspondiente:

Si la cima de la pila es...	Y hay a la vista...	La acción pertinente es...
•	**det** o **n**	apilarlo
• $[_O \cdots]$	•	analisis finalizado
• $[_{SN} \cdots]$	**v** o **prep**	apilarlo
$\cdots$ **det** $[_{\bar{N}} \cdots]$	**v**, **prep** o •	reducir a $[_{SN}$ **det** $[_{\bar{N}} \cdots]]$
$\cdots [_{\bar{N}} \cdots]$ (sense **det**)	**v**, **prep** o •	reducir a $[_{SN} [_{\bar{N}} \cdots]]$
$\cdots$ **det**	**n**	apilarlo
$\cdots$ **n**	**adj**	apilarlo
	v, **prep** o •	reducir a $[_N$ **n**$]$
•$[_{SN} \cdots][_{SV} \cdots]$	•	reducir a $[_O [_{SN} \ldots][_{SV} \ldots]]$
	prep	apilarla
$\cdots$ $[_{SN} \cdots]$ $[_{SP} \cdots]$	**v**, **prep** o •	reducir a $[_{SN} [_{SN} \cdots][_{SP} \cdots]]$
$\cdots$ **v**	**prep** o •	reducir a $[_{SV}$ **v**$]$
	det o **n**	apilarlo
$\cdots$ **prep**	**det** o **n**	apilarlo
$\cdots$ **n adj**	**v**, **prep** o •	reducir a $[_{\bar{N}}$ **n adj**$]$
$\cdots$ $[_{SV} \cdots]$ $[_{SP} \cdots]$	**prep** o •	reducir a $[_{SV} [_{SV} \cdots][_{SP} \cdots]]$
$\cdots$ **v** $[_{SN} \cdots]$	•	reducir a $[_{SV}$ **v**$[_{SN} \cdots]]$
	prep	CONFLICTO: reducir a $[_{SV}$ **v**$[_{SN} \cdots]]$ o apilarla
$\cdots$ **prep** $[_{SN} \cdots]$	**v** o •	reducir a $[_{SP}$ **prep**$[_{SN} \cdots]]$
	prep	CONFLICTO: reducir a $[_{SP}$ **prep**$[_{SN} \cdots]]$ o apilarla

Esta tabla indica qué hay que hacer en cada paso del análisis. En la tabla, el símbolo • indica tanto el principio como el final de la oración (el análisis de una oración comienza apilando • en la pila). Cuando la situación a la cual se llega no está prevista en la tabla, es porque la oración no es correcta de acuerdo con la gramática dada; esta situación de error se debe resolver de forma que el análisis pueda continuar, aunque el resultado sea un análisis parcial, puesto que nos interesa producir una traducción aproximada; el tratamiento de las situaciones de error es complejo y cae fuera del alcance de este libro. Cuando en una situación hay más de una acción posible —circunstancia que puede ser debida, como en el ejemplo, a que la gramática es ambigua— se puede hacer una de estas dos cosas: elegir siempre una acción fija o bien "duplicar" el analizador de forma que cada copia continue el análisis por cada uno de los caminos.

Veamos como se haría el análisis de la oración (no ambigua)

(8.1) El hombre trae la llave.

De esta oración, el analizador sintáctico, sólo ve la secuencia de categorías léxicas:

(8.2) **det n v det n** •

El análisis, paso a paso, es el siguiente:

Pila	Entrada restante…	Acción
•	**det n v det n** •	apilar **det**
• **det**	**n v det n** •	apilar **n**
• **det n**	**v det n** •	reducir a $[_{\bar{N}}$ **n**]
• **det** $[_{\bar{N}}$ **n**]	**v det n** •	reducir a $[_{SN}$ **det**$[_{\bar{N}}$ **n**]]
• $[_{SN}$ **det**$[_{\bar{N}}$ **n**]]	**v det n** •	apilar **v**
• $[_{SN}$ **det**$[_{\bar{N}}$ **n**]] **v**	**det n** •	apilar **det**
• $[_{SN}$ **det**$[_{\bar{N}}$ **n**]] **v det**	**n** •	apilar **n**
• $[_{SN}$ **det**$[_{\bar{N}}$ **n**]] **v det n**	•	reducir a $[_{\bar{N}}$ **n**]
• $[_{SN}$ **det**$[_{\bar{N}}$ **n**]] **v det** $[_{\bar{N}}$ **n**]	•	reducir a $[_{SN}$ **det**$[_{\bar{N}}$ …]]
• $[_{SN}$ **det**$[_{\bar{N}}$ **n**]] **v** $[_{SN}$ **det**$[_{\bar{N}}$ **n**]]]	•	reducir a $[_{SV}$ **v**$[_{SN}$ …]]]
• $[_{SN}$ **det**$[_{\bar{N}}$ **n**]] $[_{SV}$ **v**$[_{SN}$ **det**$[_{\bar{N}}$ **n**]]]	•	reducir a $[_{O}[_{SN}$ …$][_{SV}$ …]]
• $[_{O}[_{SN}$ **det**$[_{\bar{N}}$ **n**]]$[_{SV}$ **v**$[_{SN}$ **det**$[_{\bar{N}}$ **n**]]]]	•	aceptar

8.3.5. Sistemas de transferencia semántica

Los sistemas de traducción automática basados en la transferencia sintáctica (es decir, en la aproximación que dice que se pueden transformar por un lado las estructuras sintácticas —*transferencia estructural*— y por otro lado sustituir el léxico —*transferencia léxica*— haciendo ambas operaciones independientemente) suelen funcionar bien en casos sencillos, pero en general se hace necesario un análisis más profundo (Hovy 1993), puesto que la correspondencia entre las relaciones sintácticas (*sujeto*, *objeto directo*, *objeto indirecto*, etc.) y las relaciones semánticas (*agente*, *paciente*, *destinatario* etc.) de los constituyentes de una frase pueden variar de una lengua a otra. He aquí algunos ejemplos:

- En la frase *me gustan los limones*, quien produce el placer, es decir, el agente, (*los limones*) hace de sujeto en la oración, mientras que en la equivalente inglesa (*I like lemons*) hace de objeto directo o en el equivalente portugués (*Eu gosto de limões*) aparece como complemento preposicional; quien experimenta el placer hace de objeto en español y de sujeto en inglés y en portugués. La semántica de estas frases se podría resumir en la estructura abstracta

```
dar_placer(agente=limones,dest=yo)
```

- Pero todavía puede haber más formas de expresar sintácticamente el hecho de que a alguien le produce placer realizar una acción:
  - la catalana *m'agrada nadar* o la española *me gusta nadar*, donde la acción de nadar es el sujeto y el receptor de placer, el objeto indirecto;
  - la del inglés *I like swimming* o el francés *j'aime nager*, donde el receptor de placer es sujeto y la acción de nadar, el objeto;

- la del portugués, que introduce la acción con preposición: *eu gosto de nadar*, o
- la neerlandesa *ik zwem graag* o la alemana *ich schwimme gern* donde la acción de nadar pasa a ser el verbo principal y la acción de gustar se convierte en un adverbio.

La semántica de todas estas frases se podría resumir en la estructura abstracta

```
dar_placer(agente=nadar(agente=yo),experimentador=yo).
```

- La noción de *tener un nombre* también suele expresarse de manera sintácticamente divergente. Por ejemplo, en catalán se dice *Em dic Joan* o *Em diuen Joan*; en inglés *My name is Joan* o *I am called Joan*; en alemán *Ich heiße Joan*, etc.[9]
- En algunas lenguas, encontramos verbos que a veces se suelen llamar *ergativos*, como por ejemplo en inglés el verbo *sink* (*hundirse*); cuando lleva sujeto y objeto, no es especial: el sujeto es el agente y el objeto el paciente, *we sank the ship*:

  ```
  hundirse(agente=nosotros,paciente=barco),
  ```

  pero cuando lleva únicamente un sujeto, éste corresponde al *paciente* de la acción: *the ship sinks*

  ```
  hundirse(paciente=barco),
  ```

  y el agente queda sin especificar. En español también existen verbos ergativos como por ejemplo *hervir* (*Yo hiervo la leche / La leche hierve*).

Es cierto que estos casos se podrían tratar de manera particular en un sistema de transferencia sintáctica, haciendo la transformación correspondiente en el árbol de análisis sintáctico para cada verbo específico (es decir, haciendo una transferencia estructural dependiente del léxico) pero hay casos en los que también conviene olvidar la estructura sintáctica concreta de la frase en LO y fijarse más bien en la semántica, como por ejemplo cuando hay que resolver ambigüedades causadas por la anáfora o la elipsis (véanse las páginas 124 y 125). Por ejemplo, las dos frases inglesas mencionadas más arriba *Sam gave a book to Leslie* y *Sam gave Leslie a book* tienen una sintaxis diferente, pero quieren decir exactamente lo mismo: *Sam le dio un libro a*

[9] En catalán de Mallorca se usa una estructura similar con el verbo defectivo *nòmer*: *Jo nom Joan*.

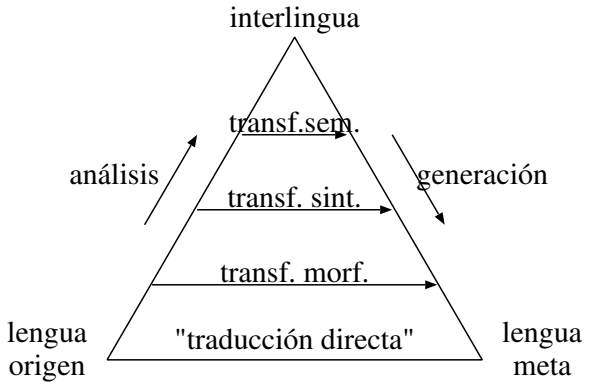


Figura 8.8: Cuanto más profundo y complejo es el análisis del texto origen, más sencilla es (menos esfuerzo requiere) la transferencia a la representación correspondiente de la lengua meta y más compleja la generación. El análisis del texto origen es tan profundo en los sistemas de interlingua clásicos que no es necesaria la transferencia.

Leslie; lo que más cuenta es quién hace la acción, a qué objeto afecta y quién es el destinatario, pero no el orden en el que estas entidades aparecen en la frase en LO (Arnold et al. 1993). Los sistemas de transferencia semántica construyen representaciones intermedias más profundas; el análisis y la generación son más complejas, pero la transferencia se simplifica.

8.4. Sistemas basados en *interlingua*

Los sistemas denominados de *interlingua*[10] aparecen en el caso extremo en que el análisis de la frase en lengua origen es tan profundo que la traducción se puede generar directamente a partir de esta sin hacer transferencia (véase la fig. 8.8). En particular, se habla de *interlingua* cuando la representación interna que se obtiene del análisis es independiente de cuáles sean la LO y la LM, es decir, la interlingua es *lingüísticamente neutral*.

Las interlinguas pueden ser de muchos tipos. Los sistemas clásicos usan representaciones estructurales más o menos complejas para representar las relaciones semánticas entre los elementos de la frase. Pero las interlinguas no tienen por què ser el resultado de un análisis profundo: lo que deben ser necesariamente es *neutrales*; por ejemplo, algunos sistemas históricos como

[10]Se tiene que tener en cuenta que el término *interlingua* se puede referir también a una lengua internacional planificada —no tan famosa como el *esperanto*— muy basada en el latín y con vocabulario europeo, y que no tiene nada que ver con la traducción automática.

DLT (Hutchins y Somers 1992, cap. 17) usan como *interlingua* una lengua *pivote* "natural" como el esperanto, con anotaciones que resuelven algunas ambigüedades típicas.[11]

En el intento de representar los significados de todas las frases de todas las lenguas, las interlinguas clásicas acabarían por ser "modelos del mundo". Esto hace que, actualmente, sólo se hayan desarrollado sistemas de interlingua clásicos para ámbitos temáticos muy concretos.

Una de las ventajas más importantes de todos los sistemas de interlingua respecto a los sistemas de transferencia es la facilidad con la que se puede añadir una lengua nueva a un sistema de traducción automática multilingüe. Imaginemos tres lenguas que denominaremos L_1, L_2 y L_3. Un sistema completo de transferencia que tradujera entre estas tres lenguas en los dos sentidos tendría tres módulos de análisis (que denominaremos A_1, A_2 y A_3), tres módulos de generación (que denominaremos G_1, G_2 y G_3) y seis módulos de transferencia (que denominaremos T_{12}, T_{13}, T_{23}, T_{31}, T_{32} y T_{21}).[12] Añadir un cuarto idioma L_4 al sistema comporta:

- Crear un nuevo módulo de análisis (A_4).
- Crear un nuevo modulo de generación (G_4).
- Construir 6 nuevos modulos de transferencia (T_{14}, T_{24}, T_{34}, T_{41}, T_{42} y T_{43}). Nótese que para esta última fase son necesarios varios expertos bilingües en sistemas de transferencia.[13]

La figura 8.9 ilustra el coste de añadir L_4 al sistema de transferencia. En cambio, en un sistema de interlingua no hay módulos de transferencia; un sistema trilingüe basado en una interlingua tendría sólo seis módulos: tres de análisis (A'_1, A'_2 y A'_3) y tres de generación (G'_1, G'_2 y G'_3). Queda claro que los módulos de análisis y de generación en estos sistemas son más complejos que en el caso de transferencia (puesto que tienen que hacer transformaciones hacia estructuras lingüísticamente neutrales), pero también está clara la ventaja del sistema de interlingua a la hora de añadir la lengua L_4: sólo hay que diseñar dos módulos nuevos, A'_4 y G'_4, y para diseñarlos sólo necesitamos una persona que conozca bien la lengua L_4 y la interlingua I que usa el sistema. La figura 8.10 ilustra el coste de añadir L_4 al sistema.

[11] Esta aproximación puede ser particularmente útil cuando las lenguas entre las que debe traducir el sistema tienen una gran similitud sintáctica y semántica, como en el caso de las lenguas románicas, con la excepción, quizás, del rumano.

[12] En general, para N lenguas $L_1, L_2, \ldots, L_N$ habría N modulos de análisis, N modulos de generación y $N(N-1)$ modulos de transferencia.

[13] En el caso general de añadir una lengua a un conjunto de N lenguas, hacen falta $2N$ nuevos módulos de transferencia.

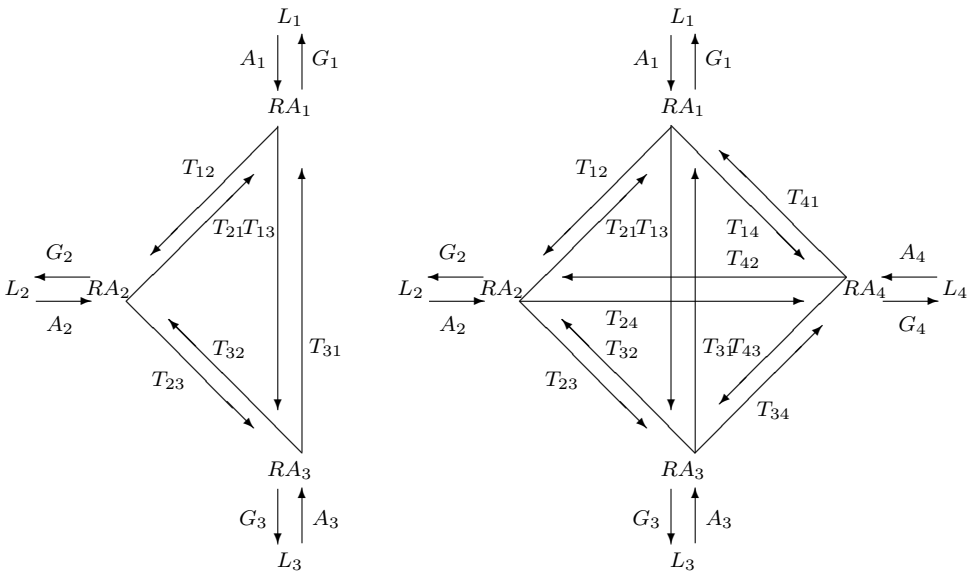


Figura 8.9: Coste de añadir una cuarta lengua L_4 a un sistema de transferencia. Las entidades RA_1 a RA_4 son las representaciones abstractas (tanto RALO como RALM) que usan los módulos de transferencia.

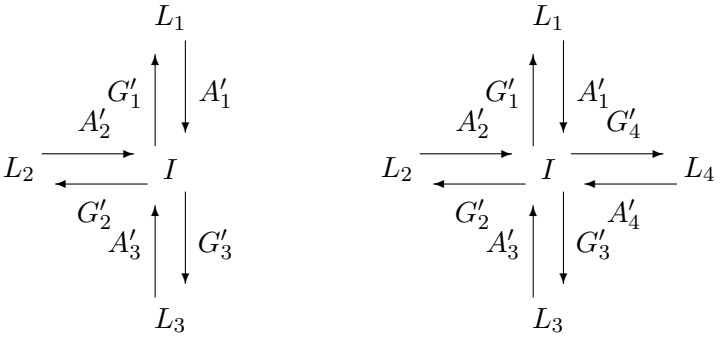


Figura 8.10: Coste de añadir una cuarta lengua L_4 a un sistema de interlingua.

8.5. Sistemas de traducción automática basados en corpus

Todas las técnicas de traducción automática descritas hasta ahora son de naturaleza *deductiva*, es decir, están basadas en teorías y conocimientos lingüísticos sobre la traducción. Pero recientemente (sobre todo en los primeros años del tercer milenio) se está produciendo un crecimiento espectacular de las técnicas *inductivas* de traducción automática, en las cuales el sistema *aprende* automáticamente a traducir entre dos lenguas a partir de un corpus paralelo suficientemente grande de oraciones en LO acompañadas de su traducción a la LM (véase el apartado 10.2). Estas aproximaciones inductivas también reciben el nombre de traducción *automática basada en corpus*.

8.5.1. Sistemas de traducción automática estadística

La principal técnica de traducción automática basada en corpus es la *traducción automática estadística* (en inglés *statistical machine translation*; SMT), que fue inventada hacia finales de los ochenta por un grupo de investigadores de IBM (Brown et al. 1990); los sistemas actuales son una evolución de estos.

A la hora de traducir hay una diferencia fundamental entre los sistemas basados en reglas o conocimiento y los sistemas estadísticos: mientras que los sistemas basados en reglas producen únicamente una traducción, los sistemas estadísticos generan una gran cantidad de *hipótesis de traducción* (idealmente todas las posibles) y utilizan modelos estadísticos para *puntuar* las hipótesis generadas y escoger la mejor de todas. Los principales modelos estadísticos que se usan para puntuar las hipótesis de traducción son el *modelo de traducción* y el *modelo de lengua*, los cuales se explican más abajo. La combinación de estos modelos hace que la hipótesis de traducción que recibe la puntuación *global* más alta no sea necesariamente la hipótesis de traducción mejor según cada modelo por separado.

El **modelo de traducción** se aprende a partir de un corpus paralelo con las oraciones ya alineadas como el que se muestra en la figura 8.11. Primero, se deben obtener los *alineamientos entre las palabras* (véase un ejemplo en la figura 8.12), para después estimar el modelo de traducción a partir de estos alineamientos.

A pesar de que parece una tarea difícil para un ordenador, los alineamientos entre las palabras se pueden obtener automáticamente sin usar ningún conocimiento sobre las lenguas de los textos a alinear mediante un proceso iterativo. La figura 8.13 ilustra este proceso con un corpus pequeño de tres oraciones paralelas; si os fijáis, sin tener ningún conocimiento de las lenguas (porque han sido inventadas) las personas también somos capaces de obtener estos alineamientos. El proceso empieza asumiendo que, para

Inglés	Español
It has been exciting in many ways .	Ha sido un trabajo apasionante en varios sentidos .
As the shadow rapporteurs know , this has been my first report during my time in Parliament and it has been a good learning experience .	Como bien saben los ponentes alternativos , éste ha sido el primer informe en el que he trabajado durante mi mandato parlamentario , y me ha venido muy bien como experiencia formativa .
It has also been very challenging to work on three reports and therefore also with other rapporteurs .	También ha sido un gran desafío trabajar en tres informes , y por lo tanto con otros ponentes .
It has been exciting .	Ha sido emocionante .

Figura 8.11: Oraciones paralelas inglés-español extraídas del corpus paralelo Europarl (`http://www.statmt.org/europarl/`) con las actas del Parlamento Europeo del período 1996–2011.

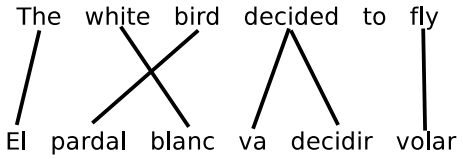
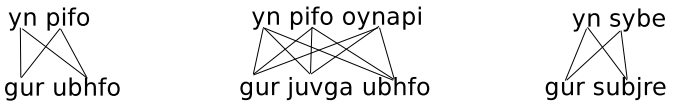
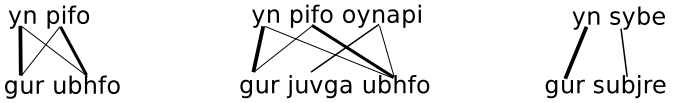


Figura 8.12: Alineamiento entre las palabras de la oración en inglés *The white bird decided to fly* y las palabras de la oración en catalán *El pardal blanc va decidir volar*.

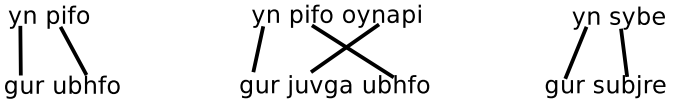


(a) Inicialización: Todos los alineamientos son igualmente probables.

(b) Primera iteración: El alineamiento entre *yn* y *gur* gana fuerza.



(c) Segunda iteración: El alineamiento entre *pifo* y *ubhfo* gana fuerza.



(d) Tercera y última iteración: La estructura de alineamiento que estaba "oculta" queda al descubierto.

Figura 8.13: Ejemplo que ilustra el proceso iterativo que permite obtener el alineamiento entre las palabras de las oraciones de un corpus paralelo. En este ejemplo el corpus consta de tres oraciones paralelas en dos lenguas inventadas. Estas oraciones paralelas son: *yn pifo–gur ubhfo*, *yn pifo oynapi–gur juvga ubhfo* i *yn sybe–gur subjre*. El grosor de las líneas que conectan las palabras representa la probabilidad del alineamiento.

cada oración paralela, todas las palabras de la oración en LM pueden ser traducción de cada una de las palabras de la oración en LO y, por lo tanto, les asigna la misma probabilidad. En cada iteración el programa alineador visita todas las oraciones paralelas del corpus y va refinando estas probabilidades hasta que la estructura de alineamiento queda definida. Este refinamiento se produce porque en cada iteración las probabilidades de la iteración anterior se usan para acumular evidencia en todo el corpus sobre la probabilidad de las correspondencias entre palabras y, además, porque las palabras que son traducción mutua suelen aparecer juntas en las mismas oraciones paralelas, lo cual no sucede si dos palabras no son traducción una de la otra.

Una vez obtenidos los alineamientos entre las palabras, podemos apren-

der modelos probabilísticos que indican, por ejemplo, la probabilidad de que la traducción de una determinada palabra en una lengua sea la traducción de una determinada palabra en la otra (un modelo de traducción de palabras o diccionario bilingüe probabilístico), o la probabilidad de que la traducción de una secuencia (segmento) de palabras en una lengua sea la traducción de una secuencia (segmento) de palabras en la otra (un modelo de traducción de segmentos). Este último modelo lo usan los sistemas de traducción automática estadística basados en segmentos bilingües (en inglés *phrase-based statistical machine translation*; Koehn (2010)), los cuales son los más usados en la actualidad.

Pero para producir buenas traducciones no podemos usar el modelo de traducción únicamente porque las traducciones serían poco naturales, gramaticales y fluidas. El motivo es que el modelo de traducción no tiene en cuenta el orden en el que aparecen los segmentos traducidos en la LM, ni el contexto en el que aparecen los segmentos en LO a la hora de puntuar sus posibles traducciones. Estas deficiencias se mitigan parcialmente con el uso de un modelo de la LM.

Un **modelo de lengua** es un modelo probabilístico que sirve para medir la verosimilitud de una oración o texto en LM; es decir, su fluidez o gramaticalidad.[14] Estos modelos se aprenden de forma automática a partir de un corpus de texto en LM y se basan en contar la frecuencia de segmentos de longitud fija, normalmente segmentos de hasta cinco palabras, para evitar asignar una verosimilitud nula a oraciones que, a pesar de que son correctas, no aparecen en los corpus de entrenamiento.[15] El modelo de lengua tiene en cuenta el orden de las palabras y, por lo tanto, asigna una verosimilitud mayor a la oración *Me gusta comer jamón del bueno* que a la oración *del bueno comer jamón Me gusta*, a pesar de que contienen los mismos segmentos de texto (*del bueno*, *comer jamón* y *Me gusta*). Además, tiene en cuenta, aunque indirectamente, el contexto en el que aparecen las palabras en LM, de forma que asigna una verosimilitud mayor a la oración en español (LM) *No piensa con la cabeza* que a la oración *No piensa con el cabo*, donde los segmentos *la cabeza* y *el cabo* son dos posibles traducciones del segmento en catalán *el cap* que aparece en la oración en LO *No pensa amb el cap*.

[14]Se asume que el modelo de lengua se aprende de textos naturales y gramaticalmente correctos en LM.

[15]Para evitar asignar verosimilitudes nulas a una oración, además de utilizar segmentos de pocas palabras, estos modelos también usan técnicas de suavizado (en inglés *smoothing*) de las probabilidades.

Para saber más sobre sistemas de traducción automática estadística

Además de los modelos de traducción y de la LM, los sistemas de traducción automática estadística basados en segmentos bilingües combinan otros modelos para establecer la puntuación global de una hipótesis de traducción. A continuación se describen muy brevemente estos modelos y para qué se usan:

Modelo de reordenamiento léxico: Su función es modelar diferentes operaciones de reordenamiento que se pueden hacer a la hora de colocar las traducciones de los segmentos en LO. Las probabilidades de estas operaciones dependen de los segmentos concretos que se están reordenando y son tres: traducción monótona (cuando no hay reordenamiento), reordenamiento (cuando la posición de la traducción del segmento en cuestión y la del anterior se intercambian) y traducción discontinua (cuando la traducción del segmento se mueve a otra posición en la oración en LM; es decir, cuando no es ninguna de las otras dos operaciones).

Ponderación léxica: Los segmentos usados para traducir pueden ser muy largos (normalmente hasta 7 palabras), lo que hace muy difícil estimar bien su probabilidad de traducción porque los segmentos largos suelen aparecer pocas veces en los corpus de entrenamiento; esto hace necesario el uso de otro modelo para estimar la calidad de los segmentos bilingües. Este modelo usa las probabilidades de traducción entre las palabras (un diccionario bilingüe probabilístico) para obtener un indicador de la calidad de los segmentos bilingües. Por ejemplo, la calidad del segmento bilingüe (*la comisión de balances de financiación*, *the funding balance comission*), donde el alineamiento entre las palabras es *la–the*, *comisión–comission*, *balances–balance* y *financiación–funding*, depende de las probabilidades de traducción de las palabras que han sido alineadas.

Número total de palabras de la oración: Cuando se puntúan las hipótesis de traducción se multiplican muchas probabilidades, es decir valores entre 0 y 1, de forma que cuanto más larga sea una traducción más probabilidades se multiplican y más fácil es llegar a tener una puntuación muy cerca de cero. Esto hace que los sistemas prefieran las traducciones cortas. Para evitar esto se introduce un modelo que cuenta el número de palabras en la hipótesis de traducción y que hace que tenga relación con el número de palabras de la oración origen.

Número de segmentos: Este modelo es similar al anterior, pero contando el número de segmentos bilingües que se han usado para producir una hipótesis de traducción. Cuanto más largos sean los segmentos, menos segmentos se usarán y más contexto tendrán; y al revés, cuanto más cortos sean los segmentos más segmentos harán falta para producir la hipótesis de traducción.

Todos estos modelos (y los anteriores) se combinan para obtener una puntuación global para cada hipótesis de traducción y poder escoger así la mejor. Esta combinación se hace asignando un peso (importancia) a cada modelo que se obtiene mediante un proceso automático (*tuning*) que intenta maximizar la *calidad* de las traducciones proporcionadas por el sistema al traducir un corpus de *desarrollo*.

Consultad el libro de Koehn (2010) para saber más sobre los modelos que se usan para traducir, el proceso de *tuning* y las medidas automáticas de la calidad que usan.

Para saber más sobre sistemas basados en corpus

Ha habido otras aproximaciones inductivas a la traducción automática, como, por ejemplo, los sistemas de *traducción automática basada en ejemplos*, a pesar de que a estas alturas ya no se usan. La *traducción automática basada en ejemplos* intenta construir *plantillas* de traducción a partir de los ejemplos observados en el corpus de oraciones paralelas y *generalizarlas* para que sirvan en nuevas situaciones. Por ejemplo, si sabemos que el sustantivo inglés *ski* se traduce por *esquí* y que la locución sustantiva *ski station* se traduce por *estació d'esquí* podemos generalizar esta última locución sustituyendo *ski* por cualquier otros sustantivo N, de forma que la traducción de "*N station*" es "estació de N"; así, si la traducción de *train* es *tren*, la traducción de *train station* es *estació de tren*, etc. (ejemplo extraído de Carl et al. 2001). Fijaos que la traducción automática basada en ejemplos puede necesitar que la muestra de frases y traducciones esté, además, anotada lingüísticamente (en el ejemplo, indicando qué palabras o estructuras funcionan como un nombre).

8.6. Cuestiones y ejercicios

Los ejercicios marcados con (*) son más difíciles.

1. Los sistemas de traducción palabra por palabra pueden cometer, por ejemplo, errores en la concordancia de género o de número. Elegid dos lenguas L_1 y L_2 y poned al menos dos ejemplos de traducciones palabra por palabra de L_1 a L_2 con problemas de concordancia.

2. (*) CasCat es un sistema de traducción automática del español al catalán que usa reglas que reordenan secuencias de formas léxicas según las categorías léxicas. Las reglas se aplican de la manera usual: de izquierda a derecha, reordenando la secuencia más larga posible, y sin que se solapen las áreas reordenadas. He aquí algunas frases españolas con *cuyo*, las traducciones producidas por CasCat, y, donde la traducción es incorrecta, una alternativa aceptable.

   a) *La chica cuyos compañeros murieron es china*
   La noia els companys de la qual van morir és xinesa

   b) *La chica cuyos compañeros de clase murieron es china*
   La noia els companys de classe de la qual van morir és xinesa

   c) *La chica cuyos compañeros mayores murieron es china*
   La noia els companys grans de la qual van morir és xinesa

   d) *La chica cuyos compañeros de clase de francés murieron es china*
   **La noia els companys de classe de la qual de francès van morir és xinesa*
   (*La noia els companys de classe de francès de la qual van morir és xinesa*)

e) *La chica cuyos compañeros mayores de classe murieron es china*
**La noia els companys grans de la qual de classe van morir és xinesa*
(*La noia els companys grans de classe de la qual van morir és xinesa*)

f) *La chica cuyos compañeros mayores de clase de francés murieron es china*
**La noia els companys grans de la qual de classe de francès van morir és xinesa*
(*La noia els companys grans de classe de francès de la qual van morir és xinesa*)

Las traducciones inaceptables están marcadas con un asterisco. Proponed un conjunto de reglas de reordenamiento que expliquen el conjunto de traducciones observado. ¿En qué casos se "rompen"sintagmas?

3. La multinacional WorldTrans ha decidido ampliar su sistema de traducción automática multilingüe LetTrans (que traduce correspondencia comercial entre cualesquier dos lenguas de un grupo de quince) y añadir la capacidad de traducir del suajili a las quince lenguas y de las quince lenguas hacia el suajili. En una oferta de trabajo, WorldTrans pide expertos en suajili, pero no pide ningún experto en traducción entre suajili y ninguna de las quince lenguas. ¿Qué clase de sistema de traducción automática es LetTrans? Justificad vuestra respuesta.

4. (*) Imaginad que tenéis un sistema de traducción automática que trabaja con dos lenguas, digamos A y B, en los dos sentidos de traducción: $A \rightarrow B$ y $B \rightarrow A$, que traducimos un texto origen T en lengua A a la lengua B mediante este traductor automático, generando un texto T', y que después usamos este mismo sistema para traducir T' de nuevo a la lengua A; denominaremos T'' al nuevo texto en lengua A.

$$T \overset{A \rightarrow B}{\longrightarrow} T' \overset{B \rightarrow A}{\longrightarrow} T'' \tag{8.3}$$

El texto T'' será previsiblemente diferente del texto T. Elegid dos lenguas A y B e indicad qué cambios son previsibles, clasificándolos según la naturaleza lingüística de los fenómenos que han causado los cambios, explicando la razón del resultado si hace falta con un ejemplo. Debéis indicar *tres tipos diferentes* de cambio.

5. (*) Imaginad que sois parte de un equipo de desarrollo de un sistema de traducción automática del inglés al catalán basado en la estrategia de transferencia morfológica avanzada (apartado 8.3.1). Los informáticos del proyecto os piden consejo sobre las reglas de reordenamiento del sistema, puesto que, por motivos técnicos, sólo pueden añadir tres.

Indicad cuáles serían las 3 reglas que propondríais, teniendo en cuenta que tienen que producir, como mínimo, tres oraciones bien traducidas en el corpus de oraciones siguientes (la traducción ideal se indica entre paréntesis, a pesar de que no siempre se podrá conseguir):

a) *A dark autumn night* (*Una nit fosca de tardor*)

b) *A high tide* (*Una marea alta*)

c) *A magic dark silhouette* (*Una silueta fosca màgica*)

d) *An autumn tide* (*Una marea de tardor*)

e) *A dark magic silhouette* (*Una silueta màgica fosca*)

f) *A dark autumn high tide* (*Una marea alta de tardor fosca*)

g) *A dark night* (*Una nit fosca*)

Dejad a un lado la concordancia y centraos sólo en los reordenamientos. Señalad cuál sería la traducción del sistema para todas las oraciones anteriores usando el conjunto de reglas que habéis propuesto.

6. ¿Cuál es la operación inversa del análisis morfológico?

   a) La obtención de la forma léxica de una palabra a partir de la forma superficial.

   b) La generación morfológica.

   c) La transferencia morfológica.

7. La traducción automática por transferencia es siempre...

   a) ... morfológica.

   b) ... directa.

   c) ... indirecta.

8. (*) Dos traducciones posibles de la palabra catalana *cap* al español son *cabe* o *cabeza*. ¿Cómo podría hacer la elección adecuada un sistema de traducción automática?

   a) Poniendo la traducción más probable, basada en las frecuencias de uso de las palabras.

   b) Usando información morfosintáctica, puesto que en la posición concreta de la frase podría ir sólo un verbo o un sustantivo.

   c) No podría, porque las dos traducciones son siempre posibles en cualquier frase.

9. ¿Cuáles de las siguientes representaciones intermedias son más costosas de obtener a partir de las frases?

a) Los árboles de análisis sintáctico correspondientes.

b) Las secuencias de categorías morfológicas correspondientes.

c) Las estructuras semánticas superficiales correspondientes.

10. El análisis morfológico toma una oración y...

a) ... produce un árbol de análisis.

b) ... produce, para cada palabra, todas las formas superficiales correspondientes.

c) ... produce, para cada palabra, todas las tripletas lema–categoría–información morfológica posibles.

11. ¿Cuáles son las fases básicas de un sistema de traducción automática indirecta?

a) Análisis, generación y traducción.

b) Análisis, transferencia y generación.

c) Análisis y transferencia.

12. ¿Cuáles de los siguientes tipos de traducción automática facilitan más la adición de una nueva lengua?

a) Los sistemas de transferencia morfológica avanzada.

b) Los sistemas de transferencia semántica superficial.

c) Los sistemas de *interlingua*.

13. ¿Cuál de los siguientes tipos de sistema de traducción automática tienen la fase de transferencia más sencilla posible?

a) Los sistemas de transferencia morfológica avanzada.

b) Los sistemas de transferencia semántica superficial.

c) Los sistemas de *interlingua*.

14. Elegid un idioma meta (francés, inglés o alemán) y un idioma origen (catalán o español). Para los idiomas elegidos, dad un ejemplo de traducción palabra a palabra inaceptable en *tres* de estos cinco casos:

a) homografía mal resuelta de una palabra

b) polisemia mal resuelta de una palabra

c) problemas de concordancia

d) ambigüedad estructural mal resuelta

e) problemas con el orden de las palabras

15. *Interlingua*, además de ser el nombre de la representación intermedia de los sistemas indirectos sin transferencia, es el nombre de una lengua artificial de raíz fundamentalmente latina, con una flexión simplificada, y con un vocabulario diseñado para ser comprensible a muchos europeos. Una característica importante de interlingua es que los determinantes (*un*, *le*, *alcun*, *iste*, *mi*, *tu*, etc.) y los adjetivos son invariables. Los plurales de los nombres se hacen con *-s* o *-es*. Imaginad que tenemos un sistema de transferencia morfológica avanzada que traduce de interlingua al catalán (o al español) usando estas cuatro reglas:

    R_1 detecta **det**–**n** y escribe trad(**det**)–trad(**n**), haciendo concordar trad(**det**) en género y en número con trad(**n**)

    R_2 detecta **det**–**n**–**adj** y escribe trad(**det**)–trad(**n**)-trad(**adj**), haciendo concordar trad(**det**) y trad(**adj**) en género y en número con trad(**n**)

    R_3 detecta **n**–**adj** y escribe trad(**n**)–trad(**adj**), haciendo concordar trad(**adj**) en género y en número con trad(**n**)

    R_4 detecta **adj**–**n** y escribe trad(**n**)–trad(**adj**), haciendo concordar trad(**adj**) en género y en número con trad(**n**)

    Si no se puede usar información de concordancia, la traducción de los determinantes y los adjetivos se hace en masculino singular. Indica qué traducciones al catalán (o al español) producirá este sistema para las frases siguientes y por qué:

    a) *Un longe viage*
    b) *Un longe viages*
    c) *Un viages longe*
    d) *Longe viages*
    e) *Un governamento non democratic*
    f) *Un governamentos non democratic*
    g) *Tu melior ideales*
    h) *Un bon solution*

16. (*) La traducción de una oración se puede ver como una interpretación de esta (es decir, como la expresión en la lengua meta de su significado). El *principio de composicionalidad semántica* postula que la interpretación de una oración se construye combinando las interpretaciones de las palabras siguiendo precisamente las agrupaciones sucesivas (constituyentes) que indica el árbol de análisis sintáctico de la oración, partiendo de las palabras y yendo hacia la raíz del árbol.

Indicad en qué tipo (o tipos) de sistema de traducción automática encontramos un diseño que aplica, exactamente o aproximadamente, el principio de composicionalidad. Razonad brevemente la respuesta.

17. El software que llevan instalado las naves de la confederación galáctica incluye un programa que traduce una de las lenguas mayoritarias del planeta Zkanagg, el tazkannwat, al español. El sistema es un sistema de transferencia morfológica avanzada estándar, que lee los textos de izquierda a derecha, palabra a palabra, busca en la entrada los patrones de categorías léxicas que contiene en su catálogo, selecciona el más largo, reordena y concuerda las palabras del patrón, los escribe, y continúa después de la zona reordenada. Algunas traducciones son erróneas porque el sistema no tiene un catálago demasiado completo de reglas. Fijaos en los ejemplos y decid cuáles son los patrones que detecta y cuáles las reglas de reordenamiento asociadas.

(8.4) *Thlong u knaar uw phlagyw.*
Adquirió el navegante el-OBJ control-OBJ
TA: El navegante adquirió el control (correcta).

(8.5) *Thlong u knaar qimratt uw phlagyw.*
Adquirió el navegante estelar el-OBJ control-OBJ
TA: El navegante estelar adquirió el control (correcta).

(8.6) *Thlong u knaar na Zkannag uw phlagyw.*
Adquirió el navegante de Zkannag el-OBJ control-OBJ
TA: El navegante de Zkannag adquirió el control (correcta).

(8.7) *Thlong u knaar qimratt na Zkannag uw*
Adquirió el navegante estelar de Zkannag el-OBJ
phlagyw.
control-OBJ
TA: *El navegante estelar adquirió de Zkannag el control.
Correcta: El navegante estelar de Zkannag adquirió el control.

18. Las palabras no son todas igualmente frecuentes en los textos. De hecho, si ordenamos las palabras de un gran corpus de texto real (de cualquier tipo y de cualquier idioma) por el número de veces que aparecen, empezando por la más frecuente, el número de apariciones se reduce drásticamente según vayamos bajando por la lista. Típicamente, la palabra más frecuente puede llegar a constituir el 10 % de todo el texto, pero la segunda sólo cubre alrededor del 5 %, la tercera alrededor del 3 %, etc.; cuando llegamos a la 100ª palabra más

frecuente ya tenemos que hablar del 0,1 % (una vez cada 1.000 palabras), y si llegamos a la posición 1000, del 0,01 % (una vez cada 10.000 palabras). En resumen, la distribución no es nada homogénea: unas pocas palabras son las más frecuentes y la mayoría son muchísimo menos frecuentes. De hecho, es típico que la mayoría de las palabras sean *hapax legomena*, es decir, palabras que han aparecido sólo una vez en todo el corpus. ¿Si tuvierais que supervisar la construcción de los diccionarios de un sistema de traducción automática, para qué os pordrían servir estas constataciones estadísticas?

19. (*) Los sistemas de traducción automática entre dos lenguas con sintaxis similar no necesitan hacer demasiados reordenamientos porque el orden de las palabras no varía demasiado de una lengua a otra. A pesar de esto, la traducción palabra por palabra no es practicable porque el género y el número gramatical de algunos sustantivos varía y los adjetivos, artículos, etc., que lo acompañan no concordarían correctamente: esp. *una señal muy clara* → cat. **una senyal molt clara* (correcto: *un senyal molt clar*); esp. *me gusta la leche fría* → ital. **mi piace la latte fredda* (correcto: *mi piace il latte freddo*). Una manera de identificar zonas donde se tiene que establecer la concordancia es detectar secuencias de palabras, de manera similar a cómo se hace en los sistemas de transferencia morfológica, pero sin reordenarlas. Por ejemplo, detectar la secuencia **art–subst** puede servir para propagar el género y el número del sustantivo al artículo. Fijaos en las frases españolas siguientes y las traducciones al catalán hechas por un sistema que usa esta estrategia y deducid cuáles son las secuencias que detecta y cuales no. Justificad vuestra respuesta.

    a) *Nos ofreció un postre → Ens va oferir unes postres*

    b) *Nos ofreció un postre buenísimo → Ens va oferir unes postres boníssimes*

    c) *Nos ofreció un buen postre → *Ens va oferir un bon postres*

    d) *Nos ofreció un postre típico buenísimo → *Ens va oferir unes postres típiques boníssim*

    e) *Nos ofreció un postre muy bueno → *Ens va oferir unes postres molt bo*

20. Indica cuál de estas afirmaciones es falsa.

    a) Los sistemas de transferencia sintáctica hacen análisis sintáctico sin hacer análisis morfológico.

    b) Los sistemas de transferencia sintáctica sólo usan información bilingüe en una de las tres fases.

c) La fase de transferencia de un sistema de transferencia sintáctica realiza transformaciones de árboles de análisis sintáctico de acuerdo con reglas determinadas.

21. Elige la secuencia que está en el orden temporal correcto:

    a) Preedición, postedición, traducción por transferencia, diseminación.

    b) Preedición, traducción por transferencia, diseminación, postedición.

    c) Preedición, traducción por transferencia, postedición, diseminación.

22. ¿En qué tipo de sistema de traducción automática tendrían básicamente la misma representación las frases *David es visto por Daniel* y *Daniel ve a David*?

    a) En un sistema de transferencia morfológica.

    b) En un sistema de transferencia semántica o de interlingua clásico.

    c) En un sistema de transferencia sintáctica.

23. ¿Cuántas lenguas naturales debe conocer el equipo de expertos que tiene que incorporar una nueva lengua a un sistema de traducción automática basado en interlingua que ya tiene 7?

    a) Siete.

    b) Una.

    c) Ocho.

24. Cuanto más profundo es el análisis en un sistema de traducción automática…

    a) …más compleja es la transferencia.

    b) …más sencilla es la generación.

    c) …más sencilla es la transferencia.

25. Si una oración tiene sólo una ambigüedad léxica pura, tiene sólo un único árbol de análisis sintáctico. Por lo tanto, si se traduce esta oración con un sistema de traducción automática indirecta por transferencia sintáctica…

    a) …el sistema se bloqueará porque sólo opera a nivel sintáctico.

    b) …la ambigüedad léxica no afecta al resultado porque no afecta a la sintaxis.

c) ...puede todavía producirse un error en la traducción a causa de la ambigüedad léxica de transferencia.

26. Un sistema de traducción automática por transferencia traduce en cualquier sentido entre cuatro lenguas. Si queremos añadir una quinta lengua para que traduzca en cualquier sentido entre cinco lenguas, ¿cuántos módulos nuevos hay que escribir?

    a) 4 de transferencia, uno de análisis y uno de generación

    b) 5 de transferencia, uno de análisis y uno de generación

    c) 8 de transferencia, uno de análisis y uno de generación

27. ¿En cuál de las tres fases de un sistema de transferencia se usan los diccionarios bilingües?

    a) En la de análisis.

    b) En la de generación

    c) En la de transferencia.

28. Un amigo mío ha diseñado un sistema de traducción automática entre el español y el portugués, pero a pesar de que me asegura que no ha programado ningún tratamiento de la ambigüedad estructural, su sistema traduce perfectamente un montón de oraciones con este tipo de ambigüedad. ¿Esto es posible?

    a) No. Probablemente ha diseñado también un módulo de preedición y el sistema elimina automáticamente cualquier causa de ambigüedad.

    b) Sí, esto puede ocurrir cuando se dan los llamados *pases gratuitos*; seguro que, si insistimos, encontraremos alguna oración mal traducida.

    c) Sí, si se trata de oraciones en las que esta ambigüedad se debe a palabras polisémicas y el programa tiene un diccionario bastante completo.

29. Los informáticos que participan en el diseño de un sistema de traducción por interlingua te informan de que cada una de las fases del sistema se tiene que ejecutar en un ordenador diferente. ¿Cuántos ordenadores tenemos que comprar?

    a) Dos, uno para la fase de análisis y otro para la de generación.

    b) Dos, uno para la fase de análisis y otro para la de transferencia.

    c) Tres, uno para la fase de análisis, otro para la de transferencia y un tercero para la de generación.

Primera palabra		Segunda palabra		Expresión	
fondos	(410)	*estructurales*	(203)	*fondos estructurales*	(63)
precio	(415)	*máximo*	(202)	*precio máximo*	(2)
algunos	(403)	*sectores*	(211)	*algunos sectores*	(1)
hacia	(409)	*ellos*	(204)	*hacia ellos*	(0)
otra	(411)	*crisis*	(203)	*otra crisis*	(0)

Tabla 8.1: Frecuencias de aparición de pares de palabras sobre economía.

30. ¿Cuántos módulos de análisis y de generación tenemos que añadir en total a un sistema basado en transferencia que ahora mismo permite traducir entre 4 lenguas, si queremos incorporar una lengua más de forma que el sistema pueda traducir (tanto en un sentido como en el otro) entre todas las lenguas existentes y la nueva?

    a) 2

    b) 4

    c) 6

31. Si una forma superficial tiene sólo una forma léxica, pero dos posibles traducciones a otra lengua...

    a) ... se trata de una palabra homófona.

    b) ... se trata de una palabra homógrafa.

    c) ... probablemente un sistema automático tendrá que recurrir a información estadística o reglas sobre el contexto para elegir una de las soluciones.

32. (*) En un corpus de textos en español sobre economía de 925.461 palabras estudiamos cuándo aparecen palabras conjuntamente. En concreto, y para poner un ejemplo, estudiamos pares de palabras gramaticalmente válidas donde la primera palabra aparece unas 400 veces en total en el corpus y la segunda palabra aparece unas 200. Fijaos en la tabla 8.1 de frecuencias de aparición de algunos pares. A pesar de que tanto la primera palabra como la segunda palabra de cada par tienen frecuencias similares, en algún caso las frecuencias de aparición conjunta son muy elevadas y en otros casos son mucho más reducidas. ¿Podríais explicar la causa de esta variación? ¿Para qué aplicación de la informática a la traducción podrían servir los resultados de un estudio numérico como este?

33. Elige una lengua origen (catalán, español, inglés, francés o alemán) y una lengua meta (catalán, español, inglés, francés o alemán) y da *tres* ejemplos de frases que se pueden traducir aceptablemente *palabra por palabra* pero tales que si cambiamos *una palabra* de las frases por otro de la misma categoría, la traducción *palabra por palabra* resulte incorrecta. En cada una de las frases, la razón lingüística por la cual la segunda traducción es incorrecta tiene que ser diferente.

34. (*) Estudiad los siguientes sintagmas nominales en maorí (una lengua polinesia hablada en Nueva Zelanda):

    (8.8) *Te* *whare* .
    Art. def. sg. casa .
    La casa.

    (8.9) *Ngā* *whare* .
    Art. def. pl. casa .
    Las casas.

    (8.10) *Te* *whare* *nui* .
    Art. def. sg. casa grande.
    La casa grande.

    (8.11) *Te* *whare* *nui* *o* *te* *aroha* .
    Art. def. sg. casa grande de Art. def. sg. amor .
    La casa grande del amor.

    (8.12) *Ngā* *whare* *nui* .
    Art. def. pl. casa grande.
    Las casas grandes.

    Como en los ejemplos, en maorí la mayoría de los nombres y adjetivos son invariables. Imaginad que sois parte de un equipo de desarrollo de un sistema de traducción automática del maorí al español (o al catalán) basado en la estrategia que hemos denominado en el apartado 8.3.1 *transferencia morfológica avanzada*.[16] Especifica completamente *dos* reglas (indicando posibles reordenamientos y operaciones para asegurar la concordancia) que permiten dar la traducción correcta de las oraciones de arriba y de las siguientes. No os preocupéis de la contracción preposición–artículo.

[16] Es decir, lee las oraciones palabra por palabra de izquierda a derecha y hace el análisis morfológico de cada palabra, prueba a detectar la secuencia más larga de palabras que concuerda con alguna secuencia de categorías léxicas que tiene en su catálogo, procesa la secuencia, y continúa inmediatamente después de la secuencia procesada.

(8.13) *Ngā whare nui o te aroha* (Las casas grandes del amor)

(8.14) *Te hau o te aroha* (El viento del amor)

(8.15) *Ngāpukapuka o te whare* (El libro de la casa)

(8.16) *Ngā ingoa o te pukapuka nui* (Los nombres del libro grande)

(8.17) *Te ingoa o ngā whare* (El nombre de las casas)

35. Se quiere construir un sistema de traducción automática que traduzca entre cualquiera de las dos lenguas del grupo formado por el portugués, el gallego, el catalán, el español y el italiano. Además, se requiere que se puedan añadir fácilmente otras lenguas como el occitano, el sardo o el asturiano. No se busca la perfección sino más bien traducciones en bruto rápidas y fáciles de entender o de corregir (es decir, con pocos errores). Teniendo en cuenta las lenguas implicadas, argumentad a favor y en contra de usar un sistema de interlingua clásico (con análisis semántico profundo) o un sistema de transferencia, indicando en cada caso como tendrían que ser las representaciones intermedias usadas.

36. Tenemos un sistema de traducción automática multilingüe que traduce en cualquier dirección entre las lenguas que considera. Para añadir una nueva lengua hemos escrito 6 módulos. ¿Cómo era el sistema antes de la adición de la nueva lengua?

    a) De interlingua con 4 lenguas (hemos añadido la quinta).

    b) De transferencia con 2 lenguas (hemos añadido la tercera)

    c) De transferencia con 4 lenguas (hemos añadido la quinta)

37. ¿Cuál de los tres módulos de un sistema de traducción automática de transferencia español–inglés contiene las reglas que indican que el pasado de *bring* es *brought* y que el plural de *foot* es *feet*?

    a) El de transferencia.

    b) El de análisis.

    c) El de generación.

38. ¿Qué tipo de sistema de traducción automática por transferencia analiza los textos originales hasta llegar a categorías como por ejemplo *agente*, *paciente*, *destinatario*, *instrumento*, *experimentador*, etc.?

    a) Los de transferencia morfológica avanzada.

    b) Los de transferencia sintáctica.

c) Los de transferencia semántica.

39. Tenemos un sistema basado en interlingua que traduce entre 6 idiomas (L_1, L_2, ..., L_6) y queremos incorporar el idioma L_7. Los expertos que trabajarán con ese sistema...

    a) ...tienen que saber traducir entre la lengua L_7 y las otras seis.

    b) ...no necesitan saber nada de las lenguas L_1 a L_6.

    c) ...tienen que escribir 12 módulos de transferencia más, 6 desde la lengua L_7 y 6 hacia la lengua L_7.

40. ¿Cuál de los tres módulos de un sistema de traducción automática indirecta por transferencia es monolingüe y trata con la lengua meta?

    a) El de generación.

    b) Todos los módulos son bilingües, no hay ninguno monolingüe.

    c) El de transferencia.

41. ¿En cuál de los tres módulos de un sistema de traducción automática indirecta por transferencia se hacen los reordenamientos de las palabras de la lengua original para que el orden sea el adecuado en la lengua meta?

    a) En el de análisis.

    b) En el de transferencia.

    c) En el de generación.

42. Un traductor automático por transferencia morfológica avanzada ...

    a) ...resuelve la polisemia mediante el uso de un analizador morfológico.

    b) ...resuelve la polisemia mediante el uso de un desambiguador léxico categorial.

    c) ...no puede resolver la polisemia con ninguno de los programas mencionados en las otras dos opciones.

43. Indicad cuál de las afirmaciones siguientes es cierta. Por norma general, los sistemas de traducción automática ...

    a) ...traducen cada una de las oraciones una a una sin tener en cuenta el resto de oraciones del texto a traducir.

    b) ...traducen directamente (palabra por palabra) de la lengua origen a la lengua meta.

    c) ...necesitan construir una interpretación completa del texto antes de traducirlo.

44. Los sistemas de traducción automática estadística ...

    a) ...aprenden a traducir a partir de diccionarios bilingües hechos a mano y de textos monolingües en la lengua meta.

    b) ...aprenden a traducir a partir de textos *comparables* en ambas lenguas (textos que hablan de lo mismo, pero que no son traducción mutua) y de textos monolingües en la lengua meta.

    c) Ninguna de las otras respuestas es correcta.

45. ¿Para qué usan los sistemas de traducción automática estadística el *modelo de lengua*?

    a) Para medir la verosimilitud (fluidez) de las traducciones.

    b) Para almacenar las diferentes alternativas de traducción de un segmento de texto.

    c) Los sistemas de traducción automática estadística no usan ningún *modelo de lengua*.

8.7. Soluciones

1. Por ejemplo, L_1=español y L_2=catalán: *un buen postre* $\rightarrow$ **un bon postres* (*unes bones postres*); *una señal inequívoca* $\rightarrow$ **una senyal inequívoca* (*un senyal inequívoc*).

2. Las traducciones observadas se pueden explicar con las tres reglas siguientes:

   - R_1: **cuyo n** $\rightarrow$ **art n de art qual**
   - R_2: **cuyo** $\mathbf{n}_1$ **de** $\mathbf{n}_2$ $\rightarrow$ **art** $\mathbf{n}_1$ **de** $\mathbf{n}_2$ **de art que**
   - R_3: **cuyo n adj** $\rightarrow$ **art n adj de art qual**

   Las reglas que se aplican en cada caso son:

   a) R_1

   b) R_2

   c) R_3

   d) R_2; no abarca el segmento *de francés* y rompe el sintagma;

   e) R_3; no abarca el segmento *de clase* y rompe el sintagma;

   f) R_3; no abarca el segmento *de clase de francés* y rompe el sintagma.

3. LetTrans es un sistema de interlingua: para añadir el suajili sólo se necesitan expertos en suajili y en la interlingua de LetTrans. Si fuera un sistema de transferencia sería necesaria la participación de expertos bilingües en suajili y cada una de las quince lenguas que ya hay en el sistema.

4. Tipos de cambios (por ejemplo, catalán→español→catalán):
   - Cambio de una palabra por un sinónimo por causa de la elección diferente de equivalentes en un sentido y en otro, *darrer→último→últim* o incluso por uno que no lo es, *direcció→dirección→adreça*.
   - Cambio de una palabra por otra a causa de una homografía en alguna de las dos lenguas *com aquest→como este→menjo aquest*; *riu sec→río seco→ric sec*.
   - Pérdida de palabras: *en tinc dos→tengo dos→tinc dos*; *hi van arribar tard→llegaron tarde→llegaron tarde*.
   - Cambios de concordancia: *La dona cosia el coixí cansada→La mujer cosía la almohada cansada→La dona cosia el coixí cansat.* Cuando traduce del catalán al español, *cansada* no concuerda con almohada y se traduce independientemente, pero a la inversa *almohada* sí que concuerda con *cansada* y el sistema los traduce como si formaran un sintagma.
5. Por ejemplo, con las reglas

   $R_1 : a\ n \rightarrow n\ a$,
   $R_2 : a_1\ a_2\ n \rightarrow n\ a_2\ a_1$
   $R_3 : n_1\ n_2 \rightarrow n_2$ "de" n_1

   se traducen bien todas excepto la (a) y la (f), que quedarían: "*Una [tardor fosca]$_{R_1}$ nit" y "*Una [tardor fosca]$_{R_1}$ [marea alta]$_{R_1}$" porque las reglas son incapaces de reconocer los sintagmas completos.
6. (b)
7. (c)
8. (b), véase el apartado 7.2.4.
9. (c)
10. (c)
11. (b)
12. (c)
13. (c)
14. (sólo a modo de ejemplo) Si la lengua origen es el catalán y la lengua meta es el inglés, tenemos:
    a) homografía mal resuelta de una palabra: *ara rius → now *rivers* en vez de *now you laugh*.

b) polisemia mal resuelta de una palabra: *rebrem el president a l'estació* → *we will welcome the president at the *season* en vez de *at the station*.

c) problemas de concordancia: *aquella gent estava feliç* → **that people* **was happy* en vez de *those people were happy*.

d) ambigüedad estructural mal resuelta: *Dóna'm la clau d'aquell sistema* → *give me the key *from that system* en lugar de *give me the key to that system*.

e) problemas con el orden de las palabras: *¿Jo he estat sempre un professional responsable* → *I have been always a professional responsible* en vez de *I have always been a responsible professional*

15. Se indican las traducciones y entre corchetes, la regla aplicada en cada caso:

a) *Un* $[_{R_4}$ *longe viage*] → *Un viatge llarg*

b) *Un* $[_{R_4}$ *longe viages*] → **Un viatges llargs*

c) $[_{R_2}$ *Un viages longe*] → *Uns viatges llargs*

d) $[_{R_4}$ *Longe viages*] → *Viatges llargs*

e) $[_{R_1}$ *Un governamento*] *non democratic* → *Un govern no democràtic*

f) $[_{R_1}$ *Un governamentos*] *non democratic* → **Uns governs no democràtic*

g) *Tu* $[_{R_4}$ *melior ideales*] → **El teu millors ideals*

h) *Un* $[_{R_4}$ *bon solution*] → **Un bona solució*

16. Entre los tipos de sistemas de traducción automática indirectos, el primero que empieza a aplicar, al menos parcialmente, el principio de composicionalidad es el de transferencia sintáctica, puesto que construye la traducción usando como paso intermedio un árbol de análisis sintáctico de la oración original. Por lo tanto, los sistemas con análisis más avanzados (transferencia semántica, interlingua) también lo aplican.

    Pero los sistemas de transferencia sintáctica no aplican exactamente el principio de composicionalidad semántica, ya que se basan en la aproximación de que se pueden traducir separadamente, por un lado, las palabras (transferencia léxica) sustituyéndolas por sus equivalentes y, por otro lado, los árboles, transformando la estructura. Esta aproximación puede no funcionar porque a veces las transformaciones de los árboles dependen de la interpretación de palabras concretas y de partes de la oración. En este sentido, los sistemas de transferencia semántica y de interlingua intentan construir una representación semántica a partir del árbol y de la semántica de las palabras, de forma que hacen una interpretación más general del principio.

17. (8.18) *Thlong u knaar uw phlagyw.*
Adquirió el navegante el-OBJ control-OBJ
TA: El navegante adquirió el control (correcta).

R_1: **verbo art nom** $\rightarrow$ **art nom verbo**

Resultado correcto.

(8.19) *Thlong u knaar qimratt uw phlagyw.*
Adquirió el navegante estelar el-OBJ control-OBJ
TA: El navegante estelar adquirió el control (correcta).

R_2: **verbo art nom adj** $\rightarrow$ **art nom adj verbo**

Resultado correcto.

(8.20) *Thlong u knaar na Zkannag uw phlagyw.*
Adquirió el navegante de Zkannag el-OBJ control-OBJ
TA: El navegante de Zkannag adquirió el control (correcta).

R_3: **verbo art nombre prep nombre-propio** $\rightarrow$ **art nombre prep nombre-propio verbo**

Resultado correcto.

(8.21) *Thlong u knaar qimratt na Zkannag uw*
Adquirió el navegante estelar de Zkannag el-OBJ
phlagyw.
control-OBJ
TA: *El navegante estelar adquirió de Zkannag el control.
Correcta: El navegante estelar de Zkannag adquirió el control.

No ha sido capaz de detectar el patrón **verbo art nombre adj prep nombre-propio** y aplica la regla R_2 que es la más larga que concuerda. El resultado es que el sintagma preposicional *de Zkannag* queda detrás del verbo.

18. Como el objetivo del equipo que diseña los diccionarios es que tengan la cobertura más alta posible (es decir, que dejen el mínimo posible de palabras sin traducir), la única estrategia razonable es la de ordenar las palabras de la lengua original por frecuencias de aparición e ir introduciéndolas en el diccionario en este orden, de forma que en cada momento siempre estamos aumentando la cobertura del diccionario lo más rápidamente posible.

19. Veamos qué pasa con cada una de las oraciones:

a) Nos ofreció un postre → *Ens va oferir unes postres*: La traducción es correcta. Parece que reconoce la secuencia (1) **art–subst** y propaga el número y el género del sustantivo al adjetivo.

b) Nos ofreció un postre buenísimo → *Ens va oferir unes postres boníssimes*: La traducción es correcta. Parece que reconoce la secuencia (2) **art–subst–adj** y propaga el número y el género del sustantivo tanto al artículo como el adjetivo,

c) Nos ofreció un buen postre → **Ens va oferir un bon postres*: No funciona. No reconoce la secuencia **art–adj–subst**, y traduce palabra por palabra.

d) Nos ofreció un postre típico buenísimo → **Ens va oferir unes postres típiques boníssim*: funciona incorrectamente porque no reconoce la secuencia completa **art–subst–adj–adj**; en cambio, sí reconoce la secuencia más corta (2) **art–subst–adj** y propaga el género del sustantivo sólo al artículo y al primer adjetivo. Después, el sistema continúa traduciendo palabra por palabra.

e) Nos ofreció un postre muy bueno → **Ens va oferir unes postres molt bo*: funciona incorrectamente porque no reconoce la secuencia completa **art–subst–adv–adj**; en cambio, sí reconoce la secuencia más corta (1) **art–subst** y propaga el género del sustantivo sólo al artículo. Después, el sistema continúa traduciendo palabra por palabra.

El sistema sólo ha usado dos secuencias (1: **art–subst** y 2: **art–subst–adj**) para intentar hacer la concordancia.

20. (a)

21. (c), véase el apartado 6.5.

22. (b)

23. (b)

24. (c)

25. (c)

26. (c)

27. (c)

28. (b)

29. (a)

30. (a)

31. (c)

32. Si la distribución de las palabras fuera al azar, la frecuencia de todos los pares de palabras sería la misma y muy baja. Pero hay palabras que tienden a estar juntas (colocaciones, unidades léxicas multipalabra, unidades terminológicas) más que al azar.

    Por ejemplo, la palabra "fondos" aparece ante la palabra "estructurales" 63 veces de las 203 veces que aparece "estructurales", es decir, unas 3 de cada 10 veces, cuando al azar aparecería 410 veces por cada 925.461, es decir, unas 4 veces cada 10.000. Por lo tanto, aparece casi mil veces más frecuentemente que el azar.

    Se puede demostrar que, a pesar de ser menos frecuentes, "precio máximo" o "algunos sectores" también tienden a estar juntos por encima del azar, tal vez por ser colocaciones propias del tema económico.

    Un estudio de bigramas (parejas) como estos puede servir:

    - primariamente, para identificar unidades terminológicas ("fondos estructurales", "Real Decreto", "política monetaria"), colocaciones ("hacer frente", "tomar posiciones"), o nombres de entidad ("Nueva York", "Rodrigo Rato", "Unión Europea") propias del texto en cuestión.
    - secundariamente, para decidir automáticamente, para una palabra que tiene varias traducciones, cuál es la traducción que "suena más natural" delante o detrás de la traducción de otra.

33. Los siguientes ejemplos están tomados para el par español–catalán; en cada caso, la primera frase es un ejemplo de traducción correcta y el segundo de traducción incorrecta:

    Homografía: Le traje un [sombrero] → Li vaig portar un [barret]; Le traje un [traje] → Li vaig portar un [vaig portar]* (correcto: vestit).

    Polisemia: El [canto] de la sirena → El [cant] de la sirena; El [canto] de la moneda → El [cant] de la moneda*. (correcto: cantell, viu)

    Concordancia de género o número : [La] indicación era [inequívoca] → [La] indicació era [inequívoca]; [La] señal era [inequívoca] → [La] senyal era [inequívoca]* (correcto: el, inequívoc)

    Anáfora: Cualquier [indicación] es importante para quien la comprenda → Qualsevol [indicació] és important per a qui [la] comprenga; Cualquier [señal] es importante para quien la comprenda → Qualsevol [senyal] és important per a qui [la] comprenga.*

34. Dos reglas son suficientes (el resto va bien palabra por palabra):

    R_1:
    - detectar "determinante nombre";
    - propagar el número (sing./pl.) del determinante maorí (te/ngā) al nombre español;
    - propagar el género (masc./fem.) del nombre español al determinante español.

    R_2:
    - detectar "determinante nombre adjetivo";
    - propagar el número (sing./pl.) del determinante maorí (te/ngā) al nombre y al adjetivo españoles;
    - propagar el género (masc./fem.) del nombre español al determinante y al adjetivo españoles.

35.
    - Ventajas de interlingua:
        - son necesarios menos módulos nuevos cuando se añade una lengua nueva al sistema (sólo uno de análisis y uno de generación).
        - no hacen falta expertos bilingües para construir módulos de transferencia (es poco verosímil que existan expertos asturiano–catalán o asturiano–sardo).
    - Desventajas de interlingua:
        - Visto el parecido sintáctico entre las lenguas involucradas, parece excesivamente costoso hacer el esfuerzo de diseñar una representación de interlingua, hacer el análisis y la generación completa de los textos (léxica, sintáctica, semántica) cuando una transferencia morfológica completa y sintáctica parcial sería suficiente.
    - Ventajas de transferencia:
        - Las lenguas son lo bastante similares como para que un sistema de transferencia morfológica completa y sintáctica parcial con pocas reglas dé resultados aceptables.
    - Desventajas de transferencia:
        - Por supuesto, cada vez que se añade una lengua a un sistema con N lenguas se deben escribir $2N$ módulos de transferencia y hacen falta expertos bilingües para construirlos todos.

    Los inconvenientes de interlingua se atenuarían si en vez de una representación interlingual semántica (basada en nociones como por ejemplo *agente*, *paciente*, *destinatario*, *tiempo*, etc.) fuera más bien de naturaleza léxica. Incluso, podría ser similar a una lengua humana.

El latín clásico no, porque a pesar de ser el origen de todas las lenguas del sistema tiene una sintaxis —verbo final— y morfología –declinación— muy diferentes; la lengua artificial llamada interlingua —"le lingua international facile e de aspecto natural elaborate por linguistas profesional como denominator commun del linguas le plus diffundite in le mundo"[17]— con anotaciones sintácticas y marcas de desambiguación podría ser una mejor opción.

36. (b)
37. (c)
38. (c)
39. (b)
40. (a)
41. (b)
42. (c)
43. (a)
44. (c)
45. (a)

[17]URI: `http://www.interlingua.com`.

Capítulo 9

Evaluación de los sistemas de traducción automática

Este capítulo pretende enunciar y describir muy brevemente algunos de los aspectos relevantes de la evaluación de los sistemas de traducción automática y dar algunas referencias que puedan ser de interés para quien quiera profundizar en este tema.

9.1. Cuestiones básicas

Cuando nos planteamos la evaluación de los sistemas de traducción automática (TA), hay algunas preguntas básicas que hay que responder. Arnold et al. (1994) plantean el problema así:

- ¿Cómo se puede decidir si un sistema de TA es *bueno*?
- ¿Cómo se puede decidir si un sistema de TA es *mejor* que otro?

y añaden la pregunta clave: "¿Qué quiere decir *bueno* o *mejor* en este contexto?" La respuesta a todas estas preguntas es muy difícil, como dice Minnis (1994): "el hecho que no se haya propuesto ningún método de evaluación o de medición estándar es un buen indicador de la magnitud del problema".

Un concepto clave es el de *utilidad*. La traducción automática será *mejor* o *de más calidad* cuanto más *útil* sea para un propósito previsto. La *utilidad* depende de la aplicación. Si la traducción automática se usa para la diseminación, es decir, como base para producir un texto adecuado para ser publicado, será más útil cuanto menos esfuerzo sea necesario para convertirla en adecuada (posteditarla). Pero si la traducción automática se usa tal como está para una aplicación de asimilación, es decir, para comprender un texto escrito en otra lengua, la utilidad aumenta con su inteligibilidad.

9.2. Tipos de evaluación

La naturaleza de la evaluación de un sistema de TA depende de varios factores:

1. *¿Para qué* se hace la evaluación? Hutchins (1996) distingue entre tres tipos básicos de evaluación:
   - la **evaluación de adecuación**, que sirve para "determinar la idoneidad [utilidad] de los sistemas de TA en un contexto operacional específico" —por ejemplo, para decidir si el sistema de TA es útil para traducir el correo comercial de una empresa alimentaria—;
   - la **evaluación diagnóstica**, que sirve para "identificar limitaciones, errores o deficiencias, las cuales pueden ser corregidas o mejoradas" —por ejemplo, defectos en el tratamiento de la concordancia verbal de las oraciones subordinadas—, y
   - la **evaluación del funcionamiento**, "para valorar el estado de desarrollo del sistema o las diferentes realizaciones técnicas" —por ejemplo, si el programa es robusto, rápido, hace un uso racional de la memoria del sistema, etc.

2. *¿Quién* hace la evaluación? La evaluación la pueden hacer:
   a) las personas que presumiblemente usarán el sistema o lo adquirirán para una empresa (evaluación de adecuación) o profesionales externos (*consultores*) contratados al efecto;
   b) los investigadores, equipos de desarrollo, programadores (evaluación diagnóstica), muy especialmente durante el desarrollo de un sistema de TA;
   c) cualquiera de los dos grupos anteriores (evaluación del funcionamiento).

3. *¿Cómo* se hace la evaluación? Cuando se evalúa un sistema de TA se tienen en cuenta:
   a) *La calidad de las traducciones en bruto* producidas por el sistema. Tradicionalmente, la calidad se ha considerado de manera relativamente desconectada de las aplicaciones concretas, y se ha visto como una combinación (en proporciones difíciles de determinar[1]) de varios factores, como por ejemplo: la *inteligibilidad* de los documentos traducidos por parte de los usuarios; la *precisión* o *fidelidad* con que el texto traducido comunica el significado del

[1]Minnis (1994) dice: "La razón por la cual la medición de la calidad es difícil es, por supuesto, el hecho de que la calidad sea un concepto tan polifacético e intangible".

documento original (las cuales tienen que ser juzgadas por parte de personas bilingües conocedoras de la temática de los documentos); la *naturalidad* o *gramaticalidad* del texto; la adecuación del estilo o del registro de los documentos traducidos, etc.

Esta evaluación se suele hacer mediante el uso de colecciones de documentos típicos o representativos (como se suele hacer en las evaluaciones de adecuación) o mediante series de pruebas objetivas (en inglés *test suites*), usadas en las evaluaciones diagnósticas[2] y diseñadas para abarcar conjuntos completos de fenómenos lingüísticos que se manifiestan en la traducción.[3]

Por otro lado, siempre se debe tener en cuenta que los métodos de evaluación de la calidad dependen del uso que se piensa dar al sistema de TA (Arnold et al. 1993); como se ha discutido en el capítulo 6, la noción central es la de *propósito* de la traducción:

- La **evaluación** de un sistema que se usa **para la *diseminación*** de textos se debe hacer estimando de alguna manera el esfuerzo de postedición, puesto que el sistema será tanto más útil cuanto más reducido sea este esfuerzo.

  Una posible medida cuantitativa de la calidad que aproxima (Sager 1993, p. 264) el esfuerzo de postedición por parte de profesionales de la traducción es la *tasa de error por palabra* (o tasa de palabras corregidas). Esta medida se tiene que calcular sobre un conjunto suficientemente grande de textos *representativos* de la tarea de traducción y se calcula como el porcentaje de inserciones, borrados y sustituciones de palabras que son estrictamente necesarios para transformar la traducción automática en bruto en una traducción adecuada al propósito. Esta medida tiene el inconveniente de que da la misma importancia a todas las operaciones de corrección, independientemente de la palabra; ello puede no ser adecuado porque el esfuerzo necesario para corregir todas las palabras no es el mismo.[4]

  Otra medida del esfuerzo de postedición es el tiempo que se tarda en posteditar una traducción automática para hacerla adecuada al propósito previsto. Medir el tiempo de postedición tiene el inconveniente que no todos los posteditores son igualmente eficientes ni tienen la misma experien-

[2] Pero no únicamente, como indica Lewis (1997), puesto que también pueden servir para que los usuarios juzguen la adecuación de la salida producida por el sistema.

[3] Por ejemplo, el reordenamiento de las palabras de los sintagmas nominales cuando se traduce del inglés al español (Mira i Gimènez y Forcada 1998; Forcada 2000).

[4] Por ejemplo, no es lo mismo corregir un artículo que ha sido mal concordado con el sustantivo al que acompaña, que un término de especialidad, cuya corrección puede requerir que nos documentemos con anterioridad.

cia posteditando.

- La **evaluación** de un sistema usado **para la *asimilación*** de información se debe hacer de forma diferente: aquí la utilidad está relacionada más con la inteligibilidad, y se podría determinar directamente a través de cuestionarios de comprensión (Jones et al. 2007) o similares (O'Regan y Forcada 2013; Ageeva et al. 2015); o indirectamente, por ejemplo, estudiando el éxito en la hora de ejecutar una tarea con las instrucciones traducidas automáticamente (Doherty et al. 2012).

b) *La facilidad de uso del sistema de TA mismo*: por ejemplo, "la facilidad con que se pueden crear y actualizar diccionarios, posteditar los textos, controlar el lenguaje de entrada" o "la extensibilidad [del sistema] a pares nuevos de idiomas o a nuevas temáticas" (Hutchins 1996).

Para saber más sobre los problemas de posteditar para determinar la calidad

La determinación de la calidad de una traducción en bruto mediante el cómputo del número de correcciones necesarias no está exenta de problemas:

- Si suponemos que existe una única traducción aceptable del texto origen (lo que es mucho suponer) y la usamos como referencia, existe más de una manera de corregir la traducción en bruto de forma que el resultado sea idéntico al de referencia. Para poder hacer comparaciones, estamos interesados en la corrección producida con el número mínimo de operaciones de inserción, borrado y sustitución de palabras; este número mínimo se puede considerar una *distancia*, y, de hecho, matemáticamente, lo es: se denomina *distancia de edición* (en inglés *edit distance*). La búsqueda de esta manera óptima de corregir puede no ser trivial para una persona, especialmente si los errores aparecen juntos y agrupados.
- Pero es que, además, la traducción de referencia puede no estar disponible; además, en la mayoría de los casos no hay una única traducción aceptable. De nuevo, si queremos comparar, querríamos encontrar la traducción aceptable más próxima a la traducción en bruto, es decir, la que se obtiene con el mínimo de correcciones posibles. Evaluar (corregir) la traducción en bruto comporta por lo tanto hacer una doble búsqueda: la persona que corrige tiene que buscar mentalmente la traducción aceptable más cercana (teniendo en cuenta los criterios que hacen aceptable una traducción, los cuales pueden no ser fáciles de aplicar), pero la *distancia* entre los dos textos también se calcula haciendo una búsqueda mental del número mínimo de correcciones necesarias.

Que sea posible que la evaluación por recuento de correcciones no sea óptima en vista de estos problemas hace que, además, sea especialmente difícil comparar las evaluaciones hechas por personas diferentes. Además, este tipo de evaluación es bastante costosa, puesto que para obtener una medición fiable de la calidad es necesario corregir textos de miles de palabras.

9.2.1. Análisis de costes y beneficios

En el caso concreto de una aplicación de diseminación, finalmente, desde un punto de vista económico, lo que es relevante a la hora de decidir si se adopta o no un sistema de traducción automática es *una comparación de los costes y de los beneficios* de usar este sistema de TA en lugar de usar exclusivamente los servicios de profesionales de la traducción: por ejemplo, si cuesta más (en gastos de personal) la postedición (revisión) de los textos meta producidos por el sistema (añadiendo el coste de usar el sistema de TA) que la traducción completa de los textos origen por parte de profesionales, entonces la adopción del sistema de TA no conviene a una empresa. Para ampliar la primera aproximación mencionada en la página 107,

$$\textbf{coste}\left(\begin{array}{c}\text{traducción automática}\\ + \\ \text{postedición}\end{array}\right) < \textbf{coste}(\text{traducción profesional}),$$

tendríamos que considerar otros factores, es decir, todos los gastos en los que se incurre cuando se adopta la traducción automática seguida de postedición:

- **Costes de *funcionamiento*** (coste efectivo por palabra), que tiene que tener en cuenta:
  - la amortización del traductor automático (en caso de su adquisición),
  - el servicio técnico y el mantenimiento del sistema,
  - la migración (adaptación de los programas que se usan, la adquisición de sistemas informáticos).
- **Costes de *preedición* y de preparación**: hay que preparar y quizás preeditar (véase el apartado 6.5) los textos que se tienen que traducir.
- **Costes de postedición**: depende de la *calidad* del texto en bruto y de la formación de los posteditores, a quienes se les puede pagar por horas de trabajo, por cantidad de texto corregido, etc.
- **Costes de formación**, puesto que los profesionales deben aprender a usar una nueva tecnología:
  - **Formación en el uso del programa de traducción automática**: los profesionales tienen que aprender a usar, configurar y quizás mantener el nuevo software asociado.
  - **Formación en postedición**, la cual tiene que permitir que los profesionales:

- conozcan el comportamiento del programa de traducción automática (por ejemplo, cuáles son los errores típicos que comete);
- aprendan técnicas de corrección, como por ejemplo el uso avanzado del procesador de textos (macroinstrucciones, sustitución de patrones, etc.).

9.3. Sobre la comparación entre traducción automática y traducción humana

Una visión predominante de la evaluación de los sistemas de TA es la llamada *metáfora del traductor humano*, según la cual (Krauwer 1993) la tarea consiste en "determinar hasta qué punto los constructores del sistema han conseguido imitar el comportamiento de un traductor humano". Sager (1993, p. 262) lo formula diciendo que "se ha argumentado que la calidad de los documentos producidos mediante traducción automática se debería evaluar en términos de la identidad con productos humanos".

Tanto Krauwer (1993) como Sager (1993) cuestionan esta visión; este último argumenta que "se debe aceptar que no hay ninguna situación que pueda servir como punto de comparación entre la traducción humana y la automática, y que quizás no hay ninguna situación en la cual la traducción humana y la automática sean igualmente adecuadas" (Sager 1993, p. 261) y propone que, en cambio, las traducciones pueden ser comparadas para ver "si satisfacen, y hasta qué punto, las expectativas del usuario final [de los documentos traducidos]", puesto que la traducción es una "actividad de mediación, cuya forma particular está determinada tanto por el texto como por las circunstancias comunicativas que requieren esta mediación" (Sager 1993, p. 261). En concreto, la traducción automática puede ser la más adecuada en algunas circunstancias, en vista de la enorme demanda general existente y, más concretamente, de la demanda de traducciones rápidas y baratas que no pueden ser producidas por profesionales.

Para saber más sobre evaluación: evaluación predictiva

Hay un tipo de evaluación que se puede considerar como un caso particular de la evaluación diagnóstica definida en el apartado 9.2, aunque no se use estrictamente para mejorar el funcionamiento de un sistema, sino sólo para predecir el comportamiento del sistema en situaciones nuevas. Lo denominaremos aquí *evaluación predictiva*, y se aplica principalmente a los sistemas de TA basados en reglas.

Para poder hacer la evaluación predictiva, es crucial que los evaluadores tengan, en primer lugar, un modelo que describa aproximadamente el funcionamiento del sistema de traducción automática (relacionado con la tipología del sistema, es decir, de transferencia morfológica, sintáctica, etc., véase el cap. 8), y, en segundo lugar, un con-

junto de textos o frases de evaluación (en inglés, *test suite*) que les permita obtener detalles concretos sobre los datos lingüísticos (p.ej., las reglas) que usa aquel modelo. Las predicciones serían del estilo de "como parece usar reglas patrón–acción del estilo de "si encuentra un patrón X, hará la acción Y" y en una serie de casos encuentra el patrón X_1 y hace la acción Y_1, podemos predecir que siempre que encuentre este mismo patrón hará la misma acción". Como la mayoría de los sistemas comerciales no nos dan suficiente información sobre la naturaleza del modelo, deberemos tratarlos como una *caja negra*; la intuición de la persona evaluadora, su conocimiento de otros sistemas o de la historia de las empresas involucradas (por ejemplo, en cuanto a la adquisición de tecnología de otras empresas) y su habilidad para elegir ejemplos reveladores le permitirán determinar aspectos básicos del modelo de traducción (normalmente los ejemplos donde el sistema no traduce adecuadamente dan mucha más información que los ejemplos que se traducen adecuadamente). En particular, cualquier evaluación predictiva necesita tener una idea clara sobre el nivel de análisis que se hace en el sistema de TA, puesto que el nivel de análisis es el que más determina la naturaleza de un sistema (véase el apartado 8.3).

Por otro lado, para que la evaluación sea útil los conjuntos de prueba deberían estar diseñados de forma que abarcaran conjuntos completos de fenómenos lingüísticos que se manifiesten con frecuencia relevante en las situaciones reales de traducción que se quieren evaluar, puesto que se quiere predecir el comportamiento del sistema en estas situaciones concretas.

Existe una relación muy estrecha entre las técnicas descritas y la llamada *ingeniería inversa*, o determinación detallada de la estrategia usada por un programa (en este caso, de traducción automática) para reprogramarla en otro.

9.4. Cuestiones y ejercicios

1. ¿Qué característica de un sistema de traducción automática se debe considerar como especialmente importante cuando se evalúa la aplicación del sistema a la *asimilación*?

   a) Sus herramientas de postedición asistida.

   b) Sus herramientas de preedición asistida.

   c) La velocidad de respuesta.

2. ¿Por qué es difícil evaluar la calidad de una traducción automática contando la cantidad mínima de postedición necesaria para hacerla adecuada cuando no hay una traducción de referencia?

   a) No es que sea difícil; sin traducción de referencia es absolutamente imposible.

   b) Porque esta tarea no se puede hacer sin conocer profundamente la estrategia usada por el sistema de traducción automática.

   c) Es relativamente sencillo corregir el texto para que sea adecuado pero es muy difícil hacerlo haciendo el mínimo número de cambios necesarios.

3. Cuando se quiere usar un sistema de traducción automática para la *asimilación* de información, ¿a que daríais *menos* peso en la evaluación?

   a) Facilidad de postedición de la traducción en bruto.

   b) Inteligibilidad de la traducción en bruto.

   c) Velocidad.

4. Elegid la respuesta errónea. A la hora de decidir la adopción de un sistema de traducción automática para la postedición ...

   a) ...tendremos que hacer una evaluación con textos parecidos a los que hay que traducir.

   b) ...los textos a usar en la evaluación deberán tener un número de palabras suficiente que nos permita extrapolar los resultados al resto de textos.

   c) ...deberemos evaluar, mediante cuestionarios o un método equivalente, la inteligibilidad de los textos traducidos en bruto.

5. Estáis evaluando un sistema de traducción automática. Indicad cuál de estas tres magnitudes no usaríais como indicador directo del esfuerzo de postedición.

   a) La inteligibilidad del texto meta en bruto.

   b) El número mínimo de palabras que se debe cambiar en el texto meta en bruto para hacerlo adecuado al propósito previsto, expresado como porcentaje respecto del número total de palabras.

   c) El tiempo necesario que se debe invertir para convertir el texto meta en bruto en un texto adecuado al propósito previsto, expresado en minutos por cada 1.000 palabras.

En cuanto al concepto de *evaluación predictiva*, mirad además los ejercicios 2, 17 y 19 del capítulo 8.

9.5. Soluciones

1. (c)
2. (c)
3. (a)
4. (c)
5. (a)

Capítulo 10

Memorias de traducción

> *Existing translations contain more solutions to more translation problems than any other existing resource* (Isabelle et al. 1993)

10.1. Introducción

Una aproximación a la traducción humana asistida por ordenador (es decir, semiautomática) que está muy relacionada con la traducción directa es la que se usa en las llamadas *memorias de traducción*.[1] La noción básica (Somers y Rutzler 1996; Samuelson-Brown 1996) es la utilidad de tener a mano, cuando se está traduciendo un texto nuevo, ejemplos de frases similares y de las traducciones correspondientes, procedentes de traducciones realizadas con anterioridad. De hecho, ciertos tipos de textos, como por ejemplo documentos técnicos, informes anuales o manuales de instrucciones (los cuales se suelen revisar frecuentemente), a menudo tienen muchas repeticiones. En estos casos, la comparación de versiones diferentes del que es esencialmente el mismo texto y la traducción repetitiva de textos similares es innecesariamente laboriosa. Además, muchas veces el trabajo de traducción comporta un esfuerzo creativo considerable, como por ejemplo cuando se trata de encontrar una equivalencia adecuada a alguna expresión especialmente difícil de traducir; las memorias de traducción permiten no tener que repetir este esfuerzo en el futuro.

El **objetivo** es, por tanto, aprovechar traducciones anteriores para no repetir el esfuerzo cuando se tienen que hacer traducciones nuevas. Debe quedar claro que, para poder hacerlo, los textos originales y las traducciones tienen que estar en formato informatizado (ficheros de texto).

[1] Como veremos más abajo, en vez de traducir palabra por palabra haciendo una simple sustitución de cada palabra origen por la(s) palabra(s) meta correspondientes, las memorias de traducción hacen sustituciones de fragmentos de más de una palabra, y, en lugar de usar un diccionario bilingüe, usan una base de datos de fragmentos de más de una palabra previamente traducidos.

10.2. Bitextos

Supondremos que, como resultado del trabajo anterior de traducción, tenemos pares de textos (E, D) donde E es el *texto izquierdo* (en la *lengua izquierda*) y D es el *texto derecho* (en la *lengua derecha*), y queremos traducir de la lengua izquierda a la lengua derecha.[2] De hecho, cuando dos textos E y D son equivalentes (es decir, traducción uno del otro) diremos que el par (E, D) es un *bitexto* o *texto paralelo*.

Observad este ejemplo donde la lengua izquierda es el catalán y la lengua derecha, el español:

> E: "Tenim textos equivalents i els volem aprofitar per a fer noves traduccions. Quan dos textos són equivalents diem que formen un bitext."
>
> D: "Tenemos textos equivalentes y los queremos aprovechar para hacer nuevas traducciones. Cuando dos textos son equivalentes decimos que forman un bitexto."

Pero los bitextos completos no se pueden aprovechar tal como están porque es muy improbable que nos encarguen de nuevo exactamente la misma tarea de traducción, es decir, que nos encarguen traducir de nuevo un texto E' cuando ya tenemos el bitexto (E', D'). Lo que sí es probable es que algunas partes del texto nuevo E' aparezcan también en la parte izquierda de algunos de los bitextos que ya tenemos. Por eso, necesitamos obtener a partir de ellos bitextos más pequeños, con partes izquierdas que tengan posibilidad de aparecer de nuevo en el futuro.

10.2.1. Segmentación de bitextos

La primera operación consiste en *segmentar* o *dividir* automáticamente cada uno de los dos textos en unidades más pequeñas, que llamaremos *segmentos*, usando algún criterio programable (véase más abajo). La figura 10.1 muestra el resultado de la segmentación de los textos E y D. Como se aprecia en la figura, puede ser que inicialmente el número de segmentos del texto izquierdo E sea diferente del número de segmentos del texto derecho D.

10.2.2. Alineación de bitextos. Unidades de traducción

El resultado de la segmentación todavía no es útil: no queda clara la correspondencia entre los segmentos izquierdos y los segmentos derechos. Necesitamos *revisar la segmentación*, es decir, unir o partir segmentos, en un

[2] Hablamos de *izquierda* y *derecha* porque puede que no sea importante (o que no se sepa) cuál de los dos textos es el original y cuál es la traducción.

E	D
e_1	d_1
e_2	d_2
e_3	d_3
...	...
...	d_M
...	
e_L	

Figura 10.1: Los bitextos E y D, segmentados, respectivamente, en L y M segmentos: $E = e_1 e_2 \dots e_L$ y $D = d_1 d_2 \dots d_M$. En el ejemplo, el texto izquierdo tiene dos segmentos más (es decir, $L = M + 2$).

E	D
e_1	d_1
e_2	d_2
e_3	d_3
...	...
e_N	d_N

Figura 10.2: Los bitextos E y D segmentados y alineados en N unidades de traducción, $(e_1, d_1), (e_2, d_2), \dots (e_N, d_N)$.

lado o en el otro, hasta que tengamos el mismo número de segmentos y sean traducción mutua. A esta operación se la llama *alinear* el bitexto.

Diremos que un bitexto (E, D) está *alineado* si las dos partes tienen el mismo número N de segmentos, $E = e_1 e_2 e_3 \dots e_N$ y $D = d_1 d_2 d_3 \dots d_N$, y sus segmentos son traducción mutua: e_1 es traducción de d_1 (o viceversa), e_2 es traducción de d_2, etc. Es decir, el texto alineado nos proporciona N bitextos (segmentos paralelos) (e_1, d_1), (e_2, d_2), (e_3, d_3), etc., más pequeños (por eso los escribimos en minúscula); estos bitextos se denominan normalmente *unidades de traducción* (UT): véase la figura 10.2.

Así es más probable que cuando nos den un texto nuevo E' tengamos traducciones para algunos de sus fragmentos. El bitexto de más arriba se podría alinear para formar las unidades de traducción

(*Tenemos*, *Tenim)*
(textos equivalents, textos equivalents)
(*i els volem aprofitar*, *y los queremos aprovechar)*
...

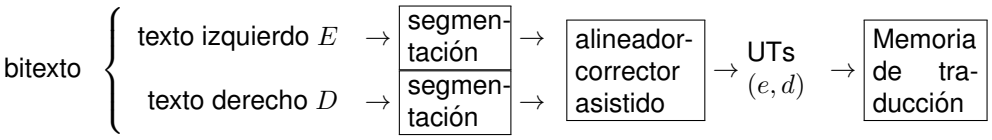


Figura 10.3: Esquema del proceso de *alineamiento* de un bitexto existente para alimentar una memoria de traducción.

(formen un bitext, forman un bitexto)

Como ya se ha dicho más arriba, una operación muy importante para la reutilización o el *reciclaje* de traducciones antiguas es la de *alinear* los textos y las traducciones existentes para identificar fragmentos o unidades *de traducción* que se puedan reutilizar posteriormente. Una *memoria de traducción* es una base de datos en la que cada ficha (registro) contiene una unidad de traducción, y tiene, por lo tanto, como mínimo dos campos: el texto izquierdo y el texto derecho.

La operación de alineación de bitextos existentes es una de las tareas que se puede realizar con la ayuda de un programa de memorias de traducción. La figura 10.3 muestra un esquema del proceso.

Alineamiento automático

El alineamiento automático de textos traducidos no es una tarea sencilla; son necesarios conocimientos previos sobre las lenguas involucradas (por ejemplo, correspondencias entre palabras o alineamientos previamente validados por una persona experta[3]), por eso, la mayoría de los sistemas de memorias de traducción usan mecanismos muy sencillos para segmentar los textos, tanto para alinear bitextos como para dividir un texto izquierdo nuevo en segmentos para buscarlos en la base de datos.[4] Por tanto, los segmentos obtenidos no son en general los *ideales* (véase más abajo). Los mecanismos de segmentación usuales intentan identificar unidades similares a la oración usando la puntuación y el formato como indicadores, con la idea de que el número de oraciones será básicamente el mismo en el texto derecho y el izquierdo. Sin embargo, esto puede fallar dado que, al contrario de lo que podría parecer, no es nada sencillo distinguir cuando un punto representa el final de una oración:

[3] U obtenido mediante métodos estadísticos como los descritos en el apartado 8.5.

[4] Es importante usar el mismo método para segmentar el bitexto antes de alinearlo y para segmentar el nuevo texto izquierdo a traducir.

```
[...] muy  tarde.   Después fueron [...]
[...] por CC.OO.    Los de UGT no [...]
[...] por CC.OO.    y por UGT [...]
[...] para el Sr.   Martínez [...]
```

En los dos primeros casos el punto es el final de la oración pero en los dos últimos no. Hay que definir claramente las reglas de segmentación; los programas de memorias de traducción suelen permitir que el usuario modifique o refine estas reglas. De hecho, hay un formato XML estándar para especificar e intercambiar reglas de segmentación, llamado SRX[5] (*segmentation rules exchange*).

Conviene, además, mencionar otro aspecto adicional importante. Los documentos de texto, además de caracteres y palabras, contienen información de *formato* que hay que considerar en el proceso de segmentación y alineación. Cuando se trata de traducir documentos de texto, el usuario quiere alinear *texto* y en muchos casos probablemente no necesita ver los códigos de formato para validar o corregir una alineación.[6] Además, las unidades de traducción que se obtengan de la alineación tendrían que ser independientes del formato del documento. Los programas más recientes resuelven esto con *filtros* o *conversores* para los formatos de texto más frecuentes, filtros que tratan de ocultar al máximo al usuario las características de formato de los documentos para que puedan concentrarse en las textuales.[7] Los profesionales valoran mucho la capacidad de gestionar el formato de manera sencilla y eficiente, puesto que la preservación del formato de las traducciones es una de las tareas que les hace perder más tiempo.

Alineamiento asistido

Es posible que los dos textos no tengan el mismo número de "oraciones" o que la estrategia para segmentarlos falle por algún motivo y el alineamiento no sea perfecto. La mayoría de los programas de memoria de traducción ofrecen al usuario la posibilidad de validar o modificar (uniendo o dividiendo segmentos en el texto izquierdo o en el texto derecho) el alineamiento automático inicial usando una interfaz sencilla e intuitiva antes de incorporar los segmentos resultantes a la memoria de traducción.

[5] `http://www.unicode.org/uli/pas/srx/srx20.html`

[6] La situación es diferente cuando se están traduciendo los mensajes o textos incluidos en *programas* de ordenador, como parte de las tareas llamadas genéricamente de *localización* (adaptación de programas de ordenador a los usuarios de una región e idioma concretos); en este caso, puede ser muy útil ver todo el programa además de los textos.

[7] A veces, quien traduce tiene que gestionar explícitamente las etiquetas de formato para asegurar una buena traducción: todos los programas ofrecen la posibilidad de *editar etiquetas* manualmente: las etiquetas se agrupan y simplifican para hacer más fácil el trabajo.

Para saber más sobre el alineamiento ideal

Hemos descrito un tipo de alineamiento posible, basado en unidades fundamentalmente equivalente a las oraciones, pero, ¿cuál sería el alineamiento óptimo de un bitexto? Es decir, ¿cuál sería la mejor manera de dividirlo en unidades de traducción? La longitud de las unidades de traducción puede ir desde las palabras hasta las oraciones enteras. La probabilidad de que un fragmento izquierdo (procedente de los bitextos ya existentes) e aparezca en un nuevo texto E' es tanto más grande cuanto más pequeño es el fragmento. Pero si el fragmento es demasiado pequeño es más probable que la traducción presente en la memoria de traducción sea más imprecisa por ser ambigua (pueden aparecer correspondencias múltiples entre las cuales se tendría que elegir: por ejemplo a una parte izquierda e le pueden corresponder dos partes derechas d y d' diferentes en unidades de traducción diferentes). Por otro lado, si los fragmentos son demasiado largos, es más improbable que sean ambiguos, pero es mucho menos probable que se repitan exactamente en textos futuros. Por ejemplo, el fragmento *las decepciones* se puede corresponder en catalán con *les decebes* o *les decepcions* pero el fragmento *las decepciones sufridas* sólo puede aparecer alineado con *les decepcions patides*. El *fragmento ideal* sería el que es lo suficientemente pequeño para aparecer a menudo pero lo suficientemente completo como para tener una traducción constante. Es decir, por un lado, se da un compromiso entre la *cobertura* (fracción de un texto nuevo que podría ser traducido usando los fragmentos alineados) y la *precisión* (corrección de las traducciones resultantes). El tamaño ideal es, por lo tanto, un compromiso entre la cobertura de los fragmentos pequeños y la precisión de los fragmentos más grandes.

10.2.3. La memoria de traducción como base de datos

Una vez alineados los bitextos las unidades de traducción se organizan para que tanto el programa como el usuario puedan acceder a ellas eficientemente; por ejemplo, como una base de datos. Se tiene que tener en cuenta que la utilidad de las memorias de traducción mejora considerablemente con el tamaño del corpus de traducciones usadas para llenarlas; por lo tanto, no es extraño que una memoria de traducción tenga que gestionar una gran cantidad de unidades de traducción. Muchos programas marcan las unidades de traducción con un código que indica la temática o la naturaleza o el nombre del bitexto del cual se han extraído las unidades de traducción, de forma que la temática del nuevo documento sirva para localizar las unidades de traducción más adecuadas en cada caso.

10.3. Traducción con memorias de traducción

La organización de las UT en una base de datos permite, además, *recuperar* de la memoria de traducción, cuando se está traduciendo un texto nuevo E', los segmentos izquierdos, $e'_1, e'_2, \ldots$ y *construir*, partiendo de los segmentos derechos correspondientes, la traducción deseada D'.

La memoria de traducción puede contener unidades de traducción con

segmentos izquierdos idénticos o similares. En caso de encontrar un segmento idéntico (*concordancia exacta*, en inglés *exact match*) para el cual sólo haya una traducción disponible, sólo hay que insertar la traducción directamente.

Pero, como esto sucede pocas veces, puesto que los segmentos son normalmente oraciones y es difícil que se repitan *exactamente*, la mayor parte de los sistemas comerciales usan estrategias para no desaprovechar unidades de traducción que contengan partes izquierdas *similares* a la nueva (las llamadas *concordancias parciales*, o, en inglés, *fuzzy matches*).

Algunos programas, si no encuentran una UT que tenga la parte izquierda idéntica a la observada en el nuevo texto (e'), pero encuentran una, (e, d), cuya parte izquierda e se diferencia en una palabra o en un cierto porcentaje de las palabras de e', presenta como *traducción aproximada* la parte derecha (d) correspondiente a e. Normalmente, los sistemas, cuando encuentran un segmento similar pero no idéntico, destacan gráficamente las diferencias (por ejemplo, con colores); así, el usuario puede hacer las modificaciones necesarias para que la traducción resultante sea correcta. Normalmente, se suele establecer un *umbral* (en inglés *fuzzy match threshold*), de manera que no se presenten propuestas por debajo de una cierta *puntuación de concordancia parcial* mínima (en inglés *fuzzy match score*). Las puntuaciones de concordancia parcial se suelen expresar como porcentajes, donde el 0 % indica falta total de concordancia y el 100 % concordancia exacta, y el umbral se suele establecer por encima del 60 %.

Algunos sistemas incluso son capaces de usar las bases de datos léxicas o terminológicas del usuario para proponer traducciones para las palabras discordantes. Por ejemplo, si en la memoria está la UT (*Connecteu l'ordinador a la impressora*, *Conecte el ordenador a la impresora*) pero el nuevo texto contiene la frase *Connecteu l'ordinador a la xarxa*, el programa puede encontrar las correspondencias (*xarxa*, *red*) e (*impressora*, *impresora*) en una base de datos léxica y usarlas para proponer la traducción correcta (*Conecte el ordenador a la red*). Otros programas usan estrategias propias (no descritas) para construir traducciones usando fragmentos de partes derechas de más de una UT. En general, la utilidad de una memoria de traducción depende en gran parte de la capacidad del sistema para proponer traducciones para segmentos *similares* (y para esto se tienen que definir y usar criterios adecuados de *similitud*).

Hay dos modalidades de uso de las memorias de traducción:

Interactiva: Quien está traduciendo recibe diversas propuestas para cada nuevo segmento e', entre las cuales elige la más adecuada para producir la traducción correspondiente d'. Esta modalidad conlleva el acceso por parte de quien traduce a la memoria de traducción.

Pretraducción: Quien está traduciendo recibe como mucho una única propuesta para cada nuevo segmento e', elegida automáticamente. En

E'	D' en construcción
$e'_1 = e_{234}$ (100 %)	$d'_1 = d_{234}$
$e'_2 \simeq e_{112}$ (87 %)	posteditar $d_{112} \rightarrow d'_2$
$e'_3 \simeq e_{47}$ (93 %)	posteditar $d_{47} \rightarrow d'_3$
e'_4	generar d'_4 desde cero
$e'_5 = e_{51}$ (100 %)	$d'_5 = d_{51}$
...	...

Figura 10.4: La traducción de un bitexto nuevo E' y los tipos de situaciones de concordancia que se pueden dar: concordancia exacta para e'_1 y e'_5, concordancia parcial para e'_2 y e'_3, y ausencia de concordancia razonable para e'_4.

esta modalidad, quien traduce no tiene acceso a la memoria de traducción.

Tanto en un caso como en el otro, pueden pasar tres cosas, tal y como se ve en el ejemplo de la figura 10.4:

- Que se dé una concordancia exacta, como en el caso de los segmentos e'_1 y e'_5 y por tanto, solo haya que comprobar que la propuesta de la memoria se pueda usar como su traducción.
- Que se dé una concordancia parcial, como en el caso de los segmentos e'_2 (similar a e_{112}) y e'_3 (similar a e_{47}), y haya que posteditar, respectivamente, las propuestas d_{112} y d_{47} de la memoria de traducción para producir d'_2 y d'_3.
- Que no se dé ninguna concordancia razonable, como en el caso del segmento e'_4, y haya que generar la traducción d'_4 desde cero.

Tanto en el caso interactivo como en el de la pretraducción, quién traduce ha de acabar la traducción. El proceso se muestra en la figura 10.5.

10.3.1. Ampliación de la memoria

Una vez hecha la pretraducción o la selección interactiva de propuestas de traducción para cada uno de los segmentos del texto izquierdo E', el programa las muestra alineadas con los segmentos del texto izquierdo para que la persona usuaria las pueda posteditar y producir el texto derecho correcto D'. Las nuevas unidades de traducción (e', d') que se hayan producido en el proceso se pueden añadir a la memoria de traducción. La repetición del ciclo pretraducción–corrección–adición de nuevas UT a la

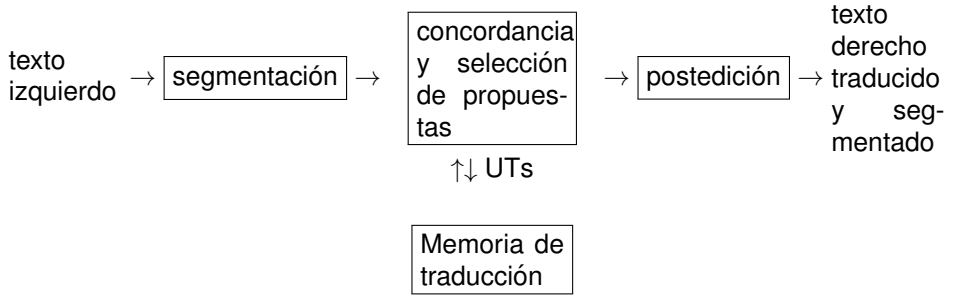


Figura 10.5: Esquema del proceso de pretraducción de un nuevo texto izquierdo utilizando una memoria de traducción.

memoria enriquece la memoria de traducción y permite obtener cada vez pretraducciones más completas y correctas.

10.4. Productos

En la actualidad, hay muchos programas de memorias de traducción de propósito general disponibles comercialmente (*SDL Trados* de SDL International, *Déjà Vu* de Atril, *IBM Translation Manager*, *Transit* de Star, etc., por nombrar algunos de los más conocidos). También hay una alternativa interesante libre/de código fuente abierto y distribución gratuita, llamada OmegaT.[8] Estos programas de traducción asistida son muy populares entre las personas que se dedican profesionalmente a la traducción, mucho más populares que los programas de traducción automática. Esto puede ser debido, por un lado, a que los profesionales tienen así la impresión de que el programa no traduce y los relega a un mero papel de correctores, sino que *organiza* y hace más eficiente el trabajo de traducción profesional y, por otro lado, a que las memorias de traducción *conservan el estilo* y las *decisiones terminológicas* de traducciones anteriores, que pueden variar de un equipo a otro, mientras que los sistemas de traducción automática suelen basarse en selecciones terminológicas y de estilo de propósito general, aunque sea dentro de una temática concreta.

Hay que tener en cuenta que la traducción automática puede constituir una alternativa aceptable en aquellos segmentos donde la memoria de traducción no puede hacer una propuesta razonable, y, de hecho, muchos programas de traducción asistida por ordenador combinan el acceso a memorias de traducción con el acceso a la traducción automática. La pun-

[8] `http://www.omegat.org`

tuación de concordancia parcial por debajo de la cual las propuestas de la memoria de traducción empiezan a ser menos útiles que la traducción automática depende de la cobertura de la memoria de traducción para el tipo de texto, de la lengua origen y la lengua meta, y, por supuesto, del sistema concreto de traducción automática: para lenguas muy similares como por ejemplo el español y el catalán, podría pasar que sólo las concordancias mejores que, digamos, el 95 % fueran aceptables, mientras que si las lenguas son más distantes, como el español y el turco, podría pasar que la traducción automática sólo fuera útil para puntuaciones de concordancia muy bajas.

Hay tecnologías de traducción automática que se asemejan mucho al funcionamiento de las memorias de traducción, como por ejemplo la traducción automática estadística (véase el apartado 8.5). Un ejemplo clásico es la edición bilingüe (español–catalán) de *El Periódico de Cataluña* (véase el epígrafe A.2.3), que se prepara diariamente con un método completamente automático que funciona en muchos aspectos de manera similar a una memoria de traducción.

10.5. El intercambio de memorias de traducción

10.5.1. El formato de intercambio TMX

Frecuentemente, los traductores forman equipos que colaboran a la hora de producir las traducciones; cuando usan memorias de traducción, es posible que haya traductores que prefieran un programa y otros que prefieran otro. ¿Quiere decir esto que no podrán compartir las memorias de traducción que hayan ido construyendo? Por suerte, no. En agosto de 1998 se aprobó la versión 1.1 de un formato estándar llamado TMX (*Translation Memory eXchange*, "intercambio de memorias de traducción"); casi todos los programes gestores de memorias de traducción pueden escribir y leer memorias en este formato. El formato TMX sigue las especificaciones XML (veáse el apartado 4.4); es decir, las memorias TMX son un tipo de documento XML, definido, por lo tanto, por una DTD concreta.[9]

La figura 10.6 muestra parte de una memoria de traducción en el formato TMX. Se muestran sólo dos unidades de traducción (`tu`) dentro del cuerpo (`body`), cada una con su identificador único (`tuid`). Cada unidad de traducción contiene dos variantes (`tuv`), cada una en una lengua (`xml:lang="..."`), además de un elemento `prop` que contiene la clave (`key`) para buscar la unidad de traducción en la base de datos. Antes del cuerpo, el elemento raíz `tmx` contiene una cabecera (`header`) con información sobre la creación y las características de la memoria.

[9] `http://www.ttt.org/oscarstandards/tmx/tmx14b.html`

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<!DOCTYPE tmx SYSTEM 'tmx13.dtd'>
<tmx version='1.3'>
 <header creationtool='Waikoloa'
                  creationtoolversion='1.00'
                  datatype='plaintext'
                  segtype='paragraph'
                  adminlang='EN-US'
                  srclang='EN-US'
                  o-tmf='okLiteTM'>
 </header>
 <body>
 <!-- ... -->
   <tu tuid='511'>
   <prop type='tmkey'>a thesaurus error occurred word
   is ending the current session</prop>
   <tuv xml:lang='EN-US'>
    <seg>A thesaurus error occurred. Word is ending
    the current session.</seg>
   </tuv>
   <tuv xml:lang='FR-FR'>
    <seg>Une erreur s'est produite pendant l'exécution
    du dictionnaire des synonymes. Word met fin à la
    session en cours.</seg>
   </tuv>
  </tu>
  <tu tuid='512'>
   <prop type='tmkey'>a thumbnail preview is not
    available for this file</prop>
   <tuv xml:lang='EN-US'>
    <seg>A thumbnail preview is not available for
         this file.</seg>
   </tuv>
   <tuv xml:lang='FR-FR'>
    <seg>Il n'y a pas d'aperçu disponible pour
         cette image.</seg>
   </tuv>
  </tu>
  <!-- ... -->
</body>
</tmx>
```

Figura 10.6: Ejemplo de memoria de traducción en TMX; sólo se muestran dos unidades de traducción.

10.5.2. Otros problemas

Incluso cuando ya se ha resuelto este problema técnico, el intercambio de memorias de traducción entre traductores o equipos de traducción diferentes no está exento de problemas. Por un lado, se pueden producir incoherencias terminológicas y de estilo entre los fragmentos procedentes de grupos diferentes; las decisiones en caso de conflicto comportan mecanismos complejos de reconocimiento de autoridad o de prestigio, que pueden ser difíciles de consensuar. Por otro lado, la organización, el mantenimiento y la explotación de grandes memorias de traducción distribuidas (en las diversas máquinas de una red) está lejos de ser trivial. Por ejemplo, en el caso del español y el catalán, una gran memoria de traducción alimentada con las traducciones hechas sólo en el ámbito de las administraciones autonómicas y locales ahorraría grandes cantidades de tiempo y dinero a la hora de mantener la documentación bilingüe de estas instituciones, pero todavía no se ha sustanciado un recurso de este tipo, a pesar de la gran cantidad de voces que han expresado la necesidad y la conveniencia.

10.6. Cuestiones y ejercicios

1. Indica cuál de estas afirmaciones es cierta:

   a) Las memorias de traducción son básicamente sistemas de traducción directa y, por lo tanto, la unidad básica de traducción que usan es la palabra.

   b) Las memorias de traducción usan información sobre las categorías léxicas de las palabras para decidir los alineamientos.

   c) Para no tener que traducir un texto nuevo desde cero con un sistema de ayuda a la traducción basado en memorias de traducción, es necesario que haya textos originales y traducidos que hayan sido alineados.

2. Una memoria de traducción se puede ver como una base de datos ...

   a) ... donde cada registro es una lengua y cada campo una unidad de traducción.

   b) ... donde cada registro es una oración y cada campo una palabra.

   c) ... donde cada registro es una unidad de traducción y la variante en cada lengua se guarda en un campo diferente.

3. ¿Qué característica de los textos que se tienen que traducir hace que el uso de memorias de traducción sea la solución adecuada?

a) Que los textos estén escritos con un léxico monosémico, es decir, preciso y no ambiguo.

b) La repetitividad.

c) La similitud entre las lenguas origen y meta.

4. En cuanto al funcionamiento, ¿a qué sistemas de traducción automática se asemejan más las memorias de traducción?

   a) A los sistemas de traducción automática directa o palabra por palabra.

   b) A los sistemas de traducción automática por interlingua.

   c) A los sistemas de traducción automática por transferencia.

5. Las memorias de traducción comerciales actuales segmentan y alinean los textos ...

   a) ... haciendo un balance óptimo entre cobertura y precisión.

   b) ... usando la información sintáctica como pista.

   c) ... usando reglas que analizan la puntuación y el formato.

6. Indica cuál de estas afirmaciones es falsa:

   a) Los resultados producidos por una memoria de traducción no necesitan revisión, ya que se basan en traducciones correctas realizadas anteriormente por profesionales.

   b) Las memorias de traducción organizan los fragmentos y las traducciones correspondientes en bases de datos similares a las bases de datos léxicas o terminológicas.

   c) Una memoria de traducción es un sistema de traducción directa con equivalencias entre fragmentos de texto extraídas de textos anteriormente traducidos.

7. Las memorias de traducción usan los signos de puntuación para segmentar las oraciones antes de alinear los textos. En particular, la aparición de un punto (“`.`”) es muchas veces un buen indicador del final de una oración, pero no siempre. ¿Cuándo no? Dad al menos *cuatro* excepciones diferentes a esta regla y describid, en cada caso, una regla sencilla que permita decidir con seguridad razonable cuando nos encontramos en cada una de estas excepciones, usando el mínimo posible de análisis lingüístico del texto.

8. La mayoría de las memorias de traducción comerciales dividen los bitextos en unidades de traducción ...

a) ... aproximadamente equivalentes a una oración, usando reglas sencillas y relativamente independientes de la lengua para segmentar (dividir) cada texto en oraciones.

b) ... en palabras y pequeñas unidades multipalabra (entre dos y cuatro palabras) de gran repetitividad.

c) ... equivalentes a una oración, usando un análisis lingüístico detallado del texto para determinar la extensión de cada oración.

9. ¿Qué tenemos que hacer con los bitextos existentes para poder reutilizar la información que contienen para hacer nuevas traducciones con una memoria de traducción?

   a) Pasarlos a XML.

   b) Segmentar cada uno de los dos textos en oraciones.

   c) Segmentar los dos textos y alinearlos.

10. Estamos traduciendo un segmento con un programa de traducción asistida (como por ejemplo OmegaT) y el programa nos da una serie de propuestas que vienen de la memoria de traducción. ¿Cuál de estos indicadores indica mejor el esfuerzo que tendremos que hacer para acabar de traducir el segmento?

    a) El porcentaje de concordancia parcial de la mejor propuesta.

    b) El número de propuestas.

    c) La longitud en palabras de la mejor propuesta (cuanto más larga, mejor).

11. La mayoría de los programas de traducción asistida basados en memorias de traducción no dan una de las tres informaciones siguientes:

    a) El porcentaje de coincidencia entre el nuevo segmento a traducir y el segmento origen de la unidad de traducción propuesta.

    b) Las palabras que hay que cambiar en el segmento meta de la unidad de traducción propuesta.

    c) Las palabras del segmento origen de la unidad de traducción propuesta que se diferencian de las del nuevo segmento a traducir.

12. ¿Cuál de estas características de un trabajo de traducción lo hace más tratable con memorias de traducción?

    a) La repetitividad de los textos.

    b) La proximidad entre las lenguas origen y meta.

c) Que los textos origen y meta estén codificados en ISO-8859-1 (*Latin-1*).

13. Para poder explotar las traducciones presentes en un bitexto (o texto paralelo) en un programa de ayuda a la traducción basado en memorias de traducción ...

    a) ...basta con incluir los bitextos como fuente de unidades de traducción para su uso.

    b) ...es necesario alinear los segmentos previamente; esta operación se puede hacer automáticamente, aunque puede requerir supervisión.

    c) ...primero hay que convertir el bitexto al formato binario que use el programa de ayuda a la traducción que estemos utilizando.

14. En general, el uso de un programa gestor de memorias de traducción hace que el proceso de traducción sea más eficiente cuando ...

    a) ...los textos a traducir son cortos.

    b) ...los textos a traducir son muy repetitivos.

    c) ...la lengua origen y meta de la traducción pertenecen a la misma familia de lenguas.

15. ¿Cómo podemos generar una memoria de traducción a partir de un texto y de su traducción?

    a) Hay que segmentar y alinear los segmentos.

    b) Hay que segmentar el texto, pero el alineamiento no es necesario porque ya sabemos que un texto es traducción del otro.

    c) No se puede, las memorias de traducción se crean al ir traduciendo los segmentos.

16. ¿Podemos usar una memoria de traducción con segmentos en inglés y español para traducir del alemán al español?

    a) Sí, si tratan del mismo tema.

    b) Sí, si la longitud de los segmentos es la misma.

    c) No.

10.7. Soluciones

1. (c)

2. (c)
3. (b)
4. (a)
5. (c)
6. (a)
7. El punto aparece en muchas construcciones que no indican el final de una oración. He aquí unos ejemplos y cómo detectarlos para no segmentar.

   a) Puntos de miles (1.259), millones (1.032.200), decimales anglosajones (3.4), números de teléfono franceses (01.10.23.87.49), fechas (20.01.1999), etc. *Solución:* Si detectamos [cifra] "." [cifra] (sin espacios), no segmentamos.

   b) Puntos en medio de siglas: CC(.)OO., EE(.)UU., etc. *Solución:* si detectamos [mayúscula] "." [mayúscula] (sin blancos), no segmentamos.

   c) Puntos al final de abreviatura de cortesía: Sr(.), Dra(.), Excm(.), etc. u otros que preceden nombres propios (Avda., Pça.) o números (Tel.). *Estrategia de solución:* si detectamos "." [espacio] [minúscula], no segmentamos; si detectamos "." [espacio] [número], no segmentamos; si detectamos "." [espacio] [mayúscula], ¿qué hacemos?: depende de la abreviatura (la decisión es impracticable sin vocabularios de nombres propios).

   d) Puntos al final de siglas: CC.OO(.), O.N.U(.). *Solución:* si detectamos [mayúsculas] "." [mayúsculas] ".", no segmentar.

   e) Puntos en URIs y direcciones de correo electrónico. *Solución:* la mejor sería una regla general (patrón o expresión regular) que detectara estas entidades y evitara segmentarlas. Posibles escapatorias: si detectamos [minúscula] "." [minúscula] (sin espacios), no segmentamos.

   f) Puntos suspensivos ("..."). *Solución:* no segmentar nunca si se encuentra "...".

8. (a)
9. (c)
10. (a)
11. (b)

12. (a)
13. (b)
14. (b)
15. (a)
16. (c)

Apéndice A

Traducción automática español–catalán

La inclusión de este apéndice en el libro tiene tres objetivos:

1. Estudiar con algo más de detalle los problemas que plantea la traducción automática entre dos lenguas emparentadas. Podríais pensar que, siendo el español y el catalán tan similares sintácticamente, los problemas serían poco importantes: el capítulo intenta convenceros de que las cosas no son tan sencillas como podrían parecer a primera vista.

2. Ilustrar, con un par de lenguas concreto, algunos de los conceptos tratados en los capítulos anteriores.

3. Proporcionar una breve descripción de las experiencias de traducción automática español–catalán existentes.

A.1. Problemática de la traducción automática español–catalán

A.1.1. Introducción

Las aplicaciones potencialmente más interesantes de la TA español–catalán se enmarcan dentro de la llamada *normalización lingüística*, es decir, el esfuerzo de las sociedades de habla catalana para promover su uso normal en todos los ámbitos; un ejemplo actual lo constituyen los servidores de Internet de instituciones públicas y de empresas privadas de los territorios de habla catalana, donde la presencia del catalán es todavía minoritaria. Cuando la lengua original de los documentos es el español, se podría usar un sistema de TA para generar borradores de documentos en catalán (o,

incluso, documentos prácticamente correctos si los documentos en español están escritos en un lenguaje controlado).

En el caso concreto del español y el catalán, la proximidad lingüística entre las dos lenguas hace que sea abordable el diseño de sistemas de traducción automática que generen textos de un nivel de corrección tal que resulte más rentable revisar el resultado en bruto producido por el programa que hacer la traducción completa (véase la p. 6.5.1).

En este capítulo se presentan algunos de los problemas más importantes con los que se puede encontrar quién quiera diseñar un sistema de traducción automática para traducir textos del español al catalán. A la vista de la notable similitud lingüística existente entre las dos lenguas, se podría pensar que la tarea de traducción automática podría, en la mayoría de los casos, ser tan sencilla como sustituir una a una las palabras en español por sus equivalentes en catalán. De hecho, el modelo de traducción automática *palabra por palabra* (definido en la pág. 154, y que no se debe confundir con lo que se suele denominar *traducción literal*) es el modelo de referencia que usaremos en este capítulo: los tres grupos de *problemas* que se presentan en este capítulo son algunos —no todos— de los que no resuelve el modelo palabra por palabra: la segmentación del texto origen, la homografía y las divergencias sintácticas.

A.1.2. Segmentación del texto origen

La segmentación de un texto en palabras suele ser normalmente muy sencilla: el programa puede usar los blancos, los tabuladores, los finales de línea o los signos de puntuación como fronteras entre palabras. Pero a veces no es tan fácil: por ejemplo, el español une muchas veces varias palabras en una sola palabra, sin que se puedan distinguir las palabras componentes; en catalán, salvo contracciones como *al*, *pels*, y *del*, siempre queda alguna indicación de esta unión, como por ejemplo un apóstrofo o un guion. Por ejemplo, en español, los pronombres enclíticos se unen al imperativo, al infinitivo y al gerundio, y muchas veces hacen que cambie la forma (véase el epígrafe 8.3.1). Por suerte, estos problemas se pueden resolver de manera sencilla usando analizadores morfológicos como los que se describen en el epígrafe 8.3.2.

A.1.3. Homografía

La homografía puede producir ambigüedad léxica categorial o incluso aparecer entre palabras de la misma categoría léxica. La homografía aparece cuando una palabra (denominada usualmente *homógrafa*) tiene más de un análisis morfológico posible (véase el apartado 7.2.1). El español —como las otras lenguas románicas— tiene muchos homógrafos. Una de las fuentes más importantes de homografía es la coincidencia entre algunas

terminaciones de la flexión verbal y algunas terminaciones de la flexión nominal y adjetival (*-a , -as*, *-o , -e*, *-es*), puesto que involucra categorías léxicas abiertas con muchos miembros.[1] Pero hay, además, otras fuentes menos productivas de ambigüedad, como por ejemplo la coincidencia de algunas de las terminaciones del presente de indicativo de los verbos en *-ar* con las del presente de subjuntivo de los verbos en *-er* e *ir* y a la inversa. Finalmente, hay algunas homografías fortuitas (algunas particularmente frecuentes, como por ejemplo *para*, preposición y verbo; *una*, determinante y verbo, y *como*, adverbio relativo, preposición y verbo).

Para ilustrar este hecho, se presenta un ensayo de clasificación —no exhaustiva— de los homógrafos españoles:

1. Homografía verbo conjugado–sustantivo:
   - *a*) En *-a :*
     - Pres. ind., 3.ª pers. sing. (1.ª conj.) / sust. fem. sing.: *casa*, *pinta*, *sala*, *toma*, *entrega*, *osa*.
     - Pres. subj., 1.ª y 3.ª pers. sing. (2.ª y 3.ª conj.) / sust. fem. sing.: *bata*, *tema*, *meta*
     - Otros: *era* (verbo *ser*, 1.ª y 3.ª pers. sing. pretérito imperf. y sust. fem. sing.).
   - *b*) En *-as*:
     - Pres. ind. 2.ª pers. sing. (1.ª conj.) / sust. fem. pl.: *casas*, *salas*, *tomas*, *entregas*, *osas*;
     - Pres. ind. 2.ª pers. sing (2.ª y 3.ª conj.) / sust. fem. pl.: *batas*, *temas*, *metas*.
     - Otros: *eras*.
   - *c*) En *-e*:
     - Pres. subj., 1.ª y 3.ª pers. sing. (1.ª conj.) / sust. masc. y fem. sing.: *cante*, *deje*, *sobre*, *pose*, *apunte*
     - Pres. ind., 3.ª pers. sing. (1.ª conj.) / sust. masc. y fem. sing.: *vale*.
     - Otros: *traje* (verb *traer*, 1.ª pers. sing. pretérito indefinido y sust. masc. sing.)
   - *d*) En *-es*:
     - Pres. subj., 2.ª pers. sing. (1.ª conj.) / sust. masc. y fem. pl.: *sales* (verb *salar*), *ases* (verbo *asar*), *cantes*, *dejes*, *sobres*, *poses*, *apuntes*
     - Pres. ind., 2.ª pers. sing. (1.ª conj.) / sust. masc. y fem. pl.: *vales*, *sales* (verb *salir*), *ases* (verb *asir*).

[1] A veces, las palabras homógrafas comparten una semántica relacionada, como *ahorro*, y otras veces no, como *oso*.

e) En *-o :*

- 1.ª pers. del presente de indicativo / sust. masc. sing.: *oso*, *remiendo*, *riego*, *mando*, *canto*, *cardo*, *recibo*, *abono*, *saldo*;
- otros: *vino*.

f) En *-os*: *marchamos* (1.ª pers. pl. presente y pretérito perfecto simple de indicativo y subst. masc. pl.).

g) Otras terminaciones: *sal* (verbo *salir*) *mentís*, *pagaré*.

2. Homografía verbo conjugado–adjetivo:

a) En *-a*:

- Pres. ind., 3.ª pers. sing. (1.ª conj.) / adj. fem. sing.: *pinta*, *monda*, *baja*, *linda*.
- Pres. subj., 1.ª y 3.ª pers. sing. (2.ª y 3.ª conj.) / adj. fem. sing.: *viva*.

b) En *-as*:

- Pres. ind. 2.ª pers. sing. (1.ª conj.) / adj. fem. pl.: *pintas*, *bajas*, *mondas*, *lindas*;
- Pres. ind. 2.ª pers. sing (2.ª y 3.ª conj.) / adj. fem. pl.: *vivas*.

c) En *-e*:

- Pres. subj. 1.ª y 3.ª pers. sing. (1.ª conj.) / adj. masc. y fem. sing.: *leve*, *ausente*, *presente*.

d) En *-es*:

- Pres. subj. 2.ª pers. sing. (1.ª conj.) / adj. masc. y fem. sing.: *leves*, *ausentes*, *presentes*.

e) En *-o*:

- 1.ª pers. del presente de indicativo / adj. masc. sing.: *pinto*, *mondo*, *bajo*, *lindo*, *vivo*.

3. Homografía verbo conjugado–verbo conjugado (muy difícil de resolver):

a) Entre verbos de la 1.ª conj. y verbos de la 2.ª o 3.ª conj.:

- *sentir/sentar*: *siento*, *sientes*, *siente*, *sienten*, *sienta*, *sientas*, *sientan*.
- *mentir/mentar*: como *sentir/sentar*
- *vendar/vender*: *vendo*, *venda*, *vendas*, *vendamos*, *vendáis*, *vendan*, *vende*, *vendes*, *vendemos*, *vendéis*, *venden*.
- *salir/salar*: *sales*, *sale*, *salen*
- *asir/asar*: como *salir/salar*

- *poder/podar*: *podamos*, *podáis*, *podemos*, *podéis*.
- *vengar/venir*: *vengo*, *vengas*, *venga*, *vengamos*, *vengáis*, *vengan*.

b) Entre la 1.ª pers. pl. del presente de indicativo y del pretérito indefinido de los verbos regulares de la 1.ª y 3.ª conj.: *amamos*, *cantamos*, *conseguimos*, etc.

c) Otros casos: *amase*, *amasen*, *amases* (*amar*, *amasar*); *fui*, *fuiste*, ... (*ir* i *ser*), *ven* (*ver* i *venir*), etc.

4. Homógrafos verbo conjugado–preposición: *bajo*, *cabe*, *entre*, *para*, *sobre*.
5. Homógrafos adjetivo–preposición: *bajo*.
6. Homógrafos sustantivo–preposición: *ante*, *sobre*.
7. Homógrafos verbo conjugado–determinante: *uno*, *una*, *unas* (*unir*)
8. Homógrafos verbo conjugado–adverbio: *así* (*asir*), *fuera* (*ser*, *ir*), *arriba* (*arribar*), *adelante* (*adelantar*), *cerca* (*cercar*).
9. Homógrafos adjetivo–adverbio: *mucho*, *poco*, *fuerte*...
10. Homógrafos sustantivo–adverbio: *antes*, *tanto*, *mal*, *bien*...
11. Homógrafos adjetivo–sustantivo: *complejo*, *impreso*, *derecho*...
12. Homógrafos determinante–pronombre *la*, *los*, *las*, *lo* (en "lo que", "lo grande")
13. Otros homógrafos: *como* (conjunción y forma de *comer*), *ora* (conjunción y forma de *orar*), *bien* (conjunción, sustantivo y adverbio)

A.1.4. Divergencias de traducción

Imaginemos que hemos podido segmentar el texto español y que hemos resuelto correctamente las ambigüedades léxicas; si todavía decidimos hacer la traducción palabra por palabra, nos encontraremos que hay ciertas construcciones para las cuales la traducción no es correcta, puesto que las palabras catalanas no se corresponden palabra por palabra con las españolas. Veamos cuáles son algunos de los problemas:

Concordancia de género y número: A veces el género y el número de una palabra varían del español al catalán. La dificultad para un sistema de traducción automática aparece a la hora de propagar el género y el número del núcleo de un sintagma a los modificadores que tengan que concordar con él: *su único amparo* → *la seua única empara*; *un buen*

postre → *unes bones postres*. Los problemas aumentan si la concordancia se debe producir entre sintagmas distantes: *el calor producido por el motor ha resultado ser nefasto* → *la calor produïda pel motor ha resultat ser nefasta*. Un problema similar lo presenta el establecimiento (opcional) de la concordancia del participio, inexistente en español en situaciones como por ejemplo *todavía no la hemos estudiado con profundidad* → *encara no l'hem estudiada amb profunditat*.

El artículo neutro: El español posee el llamado *artículo neutro*, que no tiene correspondencia en catalán estándar (*lo que me dijiste* → *el que em vas dir*); presentan especial dificultad las construcciones usadas para expresar la abstracción o la intensidad: *recibirá el informe lo más pronto posible* → *recibirá el informe el més aviat possible*; *me asusta lo grande que es* → *m'espanta com és de gran*.

Los posesivos: A veces, el catalán usa artículos determinados y construcciones con el pronombre débil *en* donde el español usa posesivos: *cuando hagas cosas así debes valorar sus consecuencias* → *quan faces coses així n'has de valorar les conseqüències*.

Los relativos: El principal problema aparece cuando se quieren traducir oraciones que contienen el relativo posesivo *cuyo*, inexistente en catalán, donde lo más sencillo es usar una construcción con *qual*, que, además, presenta un esquema de concordancia diferente (*qual* debe concordar con el antecedente, mientras *cuyo* concuerda con el nombre que lo sigue): *el contribuyente cuyos informes hemos solicitado llegará tarde* → *el contribuent els informes del qual hem sol·licitat arribarà tard* (véase el final del apartado 8.3.3).

Los pronombres débiles: Los principales problemas se encuentran: en la traducción de *lo*, ya que puede corresponder en catalán a alguna forma del pronombre masculino singular *lo* o a alguna forma del pronombre neutro *ho*; en la traducción de *se*, el cual corresponde normalmente al reflexivo catalán *se* pero en las combinaciones españolas *se la*, *se lo*, etc. puede corresponder a veces a alguna forma de *li* o *els*, y en el hecho que el español no tiene equivalentes de los pronombres catalanes adverbiales *en* y *hi* (*me* ∅ *dio uno* → *me'n va donar un*); ∅ *había dos salidas* → *hi havia dues eixides*; *no* ∅ ∅ *dejó una* → *no n'hi va deixar cap*).

Régimen preposicional: Hay diferencias notables entre los regímenes preposicionales español y catalán: las preposiciones españolas delante de *que* completivo no aparecen en catalán (*el hecho de que me hable* → *el fet que em parle*); algunas preposiciones no son posibles en catalán delante de infinitivo (*el juego consiste en ganar...* → *el joc consisteix a guanyar...*), etc.

A.2. Experiencias de TA español–catalán

En esta sección se describen brevemente cinco experiencias de traducción automática del español al catalán: SALT, el traductor español–catalán de Lucy Software, el traductor de *El Periódico de Cataluña* y Automatictrans, interNOSTRUM y, con más detalle, Apertium.

A.2.1. SALT, de la Generalitat Valenciana

El programa SALT (la versión actual es la 4.0) lleva el nombre del antiguo *Servicio de Asesoramiento Lingüístico y Traducción* de la Conselleria de Cultura, Educación y Ciencia (ahora Conselleria de Educación, Investigación, Cultura y Deporte) de la Generalitat Valenciana; se trata de un programa que se ejecuta en los sistemas operativos Windows, GNU/Linux y MacOS. El desarrollo del programa lo inició a finales de los noventa un equipo de programadores dirigido por Rafael Pinter bajo la dirección lingüística de Josep Lacreu, en aquel momento responsable de este servicio. Inicialmente, la disponibilidad del programa fue más bien reducida y su lanzamiento se retrasó por las discusiones en cuanto al estándar de valenciano que debía producir; actualmente se puede descargar gratuitamente de varios servidores de Internet[2] y también lo distribuyen los servicios de normalización lingüística de algunas universidades. SALT 4.0 se ejecuta como una extensión de los procesadores de textos LibreOffice y Openoffice.org, traduce textos en español a la variante valenciana del catalán y está concebido también como una ayuda a las personas que quieren empezar a generar documentos en valenciano (entre otras herramientas, incluye diccionarios y guías de consulta completísimas). La Academia Valenciana de la Lengua declaró *oficiales* "los contenidos" del programa SALT 2 (acuerdo de 20 de mayo del 2002). [3]

A.2.2. El traductor español–catalán de Lucy Software

El sistema de traducción automática español–catalán de Lucy Software, originalmente desarrollado por la empresa Incyta de Cornellà en colaboración con la Universitat Autònoma de Barcelona es un sistema de trans-

[2]como por ejemplo `http://www.ceice.gva.es/polin/val/salt/apolin_salt4.htm` y `https://www.softcatala.org/wiki/rebost:Salt`

[3]En el año 2000, la empresa Autotrad de Valencia lanzó el programa Ara. El gerente de la empresa era Rafael Pinter, responsable informático de SALT. Ara era básicamente una versión bastante mejorada de la primera versión de SALT, con una apariencia muy similar pero con algunas diferencias: p.e., producía textos en catalán oriental estándar, podía dialogar con la persona usuaria en español y en catalán, y permitía programar tareas de traducción que se ejecutaban sin necesidad de que la persona usuaria las atendiera. El coste (en 2004) era de 45 euros por licencia. El sitio web de la empresa[4] no parece funcionar correctamente, y es posible que el programa ya no se esté comercializando.

ferencia sintáctica estándar (véase el apartado 8.3.3), heredero del sistema METAL de la empresa Siemens. Su desarrollo fue pasando de una empresa a otra: en la actualidad lo desarrolla la empresa Lucy Software y lo distribuye la empresa Incyta, S.L. El programa se puede usar en Internet[5] y los resultados son de gran calidad.

A.2.3. El traductor de *El Periódico de Cataluña* y AutomaticTrans

Una experiencia interesante (Fité 2006) de traducción español–catalán para la diseminación es la edición bilingüe del diario *El Periódico de Cataluña*;[6] el texto original —en español la mayor parte de las veces— se traduce usando un sistema de traducción automática basado en corpus combinado con técnicas similares a las *memorias de traducción* (véase el capítulo 10) y después es revisado por los redactores de cierre del mismo periódico antes de ser publicado.

Un programa similar (y, según la información de la que disponemos, de origen común) al usado por *El Periódico de Cataluña* se denominaba antes AutomaticTrans y ahora probablemente lo comercializa la empresa AT Language Solutions.[7]

A.2.4. interNOSTRUM

Un equipo de investigadores de la Universitat d'Alacant, financiado por la extinta Caja de Ahorros del Mediterráneo y por la misma Universidad, desarrolló entre 1998 y 2006 bajo la dirección de uno de los autores de este libro un sistema de traducción automática español–catalán llamado interNOSTRUM (Canals-Marote et al. 2001a,b). El objetivo del proyecto era desarrollar un sistema de traducción automática del español a las variantes estándares del catalán y el sistema inverso correspondiente. Durante el último decenio, ha sido uno de los sistemas de traducción automática más usados en Internet.

La versión actual de interNOSTRUM (que estuvo accesible de forma gratuita a través de la URI `http://www.internostrum.com` y que en noviembre de 2016 todavía estaba accesible a través de la dirección `http://torsimany.ua.es/index.php`) genera, casi instantáneamente, borradores de traducciones al catalán listas para ser corregidas (posteditadas).

interNOSTRUM traduce textos en formatos ANSI, HTML y RTF del español al catalán oriental y a la inversa y permite la navegación traducida por Internet (es decir, permite la traducción instantánea de los documentos que se vayan visitando sin tener que invocar explícitamente el traductor).

[5] `http://www.lucysoftware.com/catala/traduccio-automatica/kwik-translator-/`

[6] Disponible por Internet: `http://www.elperiodico.es`.

[7] `https://www.at-languagesolutions.com/`

El traductor estaba escrito para ejecutarse sobre el sistema operativo GNU/Linux y es todavía accesible, como ya se ha dicho, a través de un servidor de Internet.[8] Se trata de un sistema de transferencia morfológica avance como el descrito en el apartado 8.3.1; el diseño del sistema es muy similar al del sistema Apertium que se describe más abajo en la sección A.2.5, del cual es precursor.

A.2.5. Apertium

Apertium[9] (Forcada et al. 2011), iniciado en la Universitat d'Alacant en 2004, es una plataforma de software libre o de código fuente abierto que permite construir sistemas de traducción automática de transferencia morfológica avanzada (como los del apartado 8.3.1); que sea libre o de código fuente abierto quiere decir que se puede descargar y copiar libremente pero también que programadores y lingüistas pueden modificar el software, los diccionarios, las reglas, etc. y distribuir versiones modificadas, dado que además del ejecutable del software que necesitamos para usarlo, se distribuye el código fuente, es decir, la forma del software que permite a los expertos modificarlo.

El primer sistema que se construyó sobre la plataforma Apertium fue el sistema español–catalán (en la actualidad hay disponibles en Apertium más de 40 sistemas de traducción automática diferentes).

Apertium se puede usar gratuitamente en línea a través de muchas *webs*[10] pero también se puede instalar localmente. La versión completa —por ejemplo para montar un servidor para un entorno de producción— se instala sobre ordenadores con sistema operativo GNU/Linux, pero hay otras muchas versiones que funcionan sin necesidad de conexión a Internet, como por ejemplo:

- Una aplicación para el sistema operativo Android, *Apertium offline translator*;[11]
- Una aplicación de sobremesa, *apertium-caffeine*[12] para GNU/Linux, Windows o MacOS (requiere que se haya instalado Java);
- Una extensión para el programa de traducción asistida OmegaT, denominada *apertium-omegat*.[13]

[8] A pesar de que ya no se mantiene: también hay disponible una versión para servidores basados en el sistema operativo Windows.

[9] `www.apertium.org`

[10] Por ejemplo: `http://www.apertium.org`, `http://apertium.ua.es`, `http://apertium.uoc.edu`, `http://politraductor.upv.es` y `http://aplica.prompsit.com`.

[11] `https://play.google.com/store/apps/details?id=org.apertium.android`

[12] `http://wiki.apertium.org/wiki/Apertium-Caffeine`

[13] `http://wiki.apertium.org/wiki/Apertium-OmegaT`

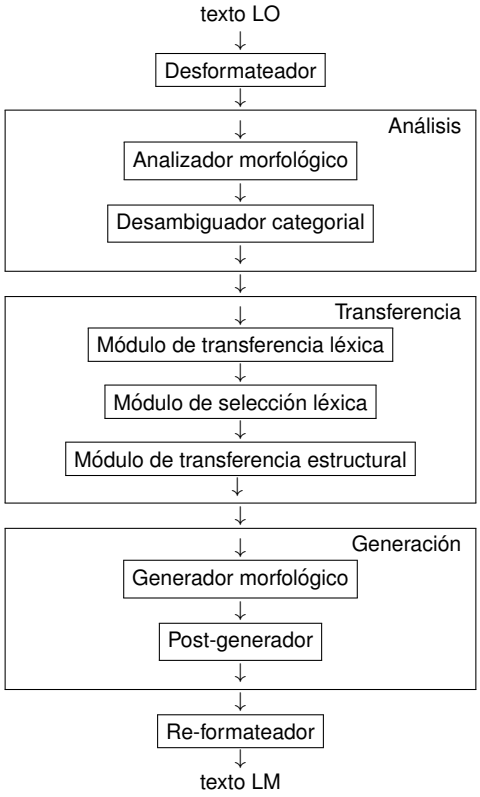


Figura A.1: Arquitectura más común de los sistemas de traducción automática basados en Apertium.

El diseño de Apertium Como se ha dicho más arriba, los sistemas de traducción automática basados en la arquitectura Apertium son todos sistemas de transferencia morfológica avanzada. La arquitectura de un sistema de traducción concreto basado en Apertium es flexible: dependiendo de la lengua origen y la lengua meta, se pueden seleccionar módulos diferentes. La figura A.1 representa la configuración más común de un sistema de traducción automática basado en Apertium: la construcción del texto de entrada se va haciendo etapa por etapa, como en la cadena de montaje de una fábrica de automóviles. El sistema español–catalán sigue la arquitectura de la figura A.1 pero la versión disponible en 2016 no tiene (todavía) módulo de selección léxica.

Apertium separa efectivamente el *motor de traducción* (que consiste en módulos genéricos, comunes a todas los pares lengua origen–lengua meta) y los *datos lingüísticos* específicos de un par lengua origen–lengua meta, tal como se discute en la p. 156. Esto permite que las personas expertas que

desarrollan datos lingüísticos (diccionarios, reglas) para un sistema determinado no tengan que preocuparse de cómo está programado el motor de traducción.

Los siguientes párrafos describen algunos de los módulos con más detalle.

Subprogramas basados en técnicas de estados finitos: Los módulos de *análisis morfológico*, *transferencia léxica*, *generación morfológica* y *postgeneración* están basados en *transductores de estados finitos*, similares a los descritos en el cuadro "para saber más" del epígrafe 8.3.2. Esta tecnología permite velocidades de procesamiento del orden de 10.000 palabras por segundo en equipos estándares, velocidades que prácticamente no dependen del tamaño de los diccionarios. Los *transductores de estados finitos* usados en Apertium leen la entrada símbolo a símbolo; cada vez que se lee una letra cambian de estado y van produciendo, también letra a letra, una o más salidas.

Analizador morfológico: El analizador morfológico se genera automáticamente a partir de un *diccionario morfológico* de la lengua origen (LO), el cual contiene los lemas, los paradigmas de flexión y las conexiones entre ellos. La entrada son las formas superficiales del texto y la salida, formas léxicas consistentes en lema, categoría léxica e información de flexión.

Transferencia léxica : El subprograma de consulta del diccionario bilingüe se genera automáticamente a partir de un fichero que contiene las correspondencias bilingües. La entrada es la forma léxica de la LO y la salida, la forma léxica o formas léxicas correspondientes en la lengua meta (LM).

Generación morfológica: El generador morfológico hace la operación inversa al analizador morfológico pero con formas de la LM y se genera automáticamente a partir de un diccionario morfológico de la LM.

Postgeneración: Las formas superficiales que están implicadas en procesos de apostrofación y guionado (pronombres átonos, artículos, algunas preposiciones, etc.) activan este subprograma, que normalmente se encuentra inactivo. El postgenerador se genera a partir de reglas sencillas de apostrofación, guionado y combinación de pronombres átonos.

Cómo ya se ha discutido más arriba, la división de un texto en palabras presenta algunos aspectos no triviales; se mencionan dos: las *locuciones* (o *giros*) y los pronombres enclíticos.

Locuciones y giros: Hay numerosas locuciones y giros que se pueden tratar como *unidades léxicas multipalabra* y que se van incorporando a los diccionarios morfológicos de las dos lenguas y al diccionario bilingüe:

- *con cargo a* → *a càrrec de*
- *por adelantado* → *per endavant*, *a la bestreta*
- *el abajo firmante* → *el sotasignat*
- **echar** *de menos* → **trobar** *a faltar*

En el último ejemplo, el giro no es invariable sino que tiene un elemento que se flexiona (en negritas).

Pronombres enclíticos: El subprograma de análisis morfológico también es capaz de resolver las combinaciones de verbos y pronombres débiles enclíticos en español, las cuales presentan variaciones ortográficas como por ejemplo cambios de acentuación o pérdida de consonantes:

- *dámelo* = *da* + *me* + *lo* → *dóna* + *me* + *lo* = *dóna-me'l*
- *pongámonos* = *pongamos* + *nos* → *posem* + *nos* = *posem-nos*.

El sistema Apertium trata estos dos problemas con el analizador morfológico, el cual es capaz de decidir cuando un grupo de palabras se tiene que tratar conjuntamente o por separado.

El módulo de desambiguación léxica categorial: Este programa se encarga de decidir, cuando el analizador morfológico entrega, para una palabra homógrafa, más de una forma léxica, cuál es la forma léxica más adecuada en el contexto. Los desambiguadores léxicos categoriales combinan (véase el apartado 7.2.4) reglas de base lingüística que permiten eliminar algunas formas léxicas y modelos estadísticos, entrenados sobre un corpus de referencia, que asignan una probabilidad a cada posible desambiguación de la frase que contiene palabras con ambigüedad categorial: la desambiguación más probable (la más verosímil) es la elegida.

El módulo de transferencia estructural: A pesar del gran parecido entre el español y el catalán, hay divergencias gramaticales considerables (véase la sección A.1.4):

- perífrasis modales: *tienen que firmar* → *han de firmar*;
- cambios de género y número: *la deuda contraída* → *el deute contret* (masc.);

- caída de preposiciones: *la intención de que el cliente* → *la intenció ∅ que el client*;
- construcciones relativas: *la cuenta cuyo titular es* → *el compte el titular del qual és*.

Estas divergencias se deben tratar con las reglas gramaticales oportunas, muy similares a las que se discuten en el apartado 8.3.1: la solución se basa en la detección y el tratamiento de secuencias predefinidas de categorías léxicas (denominadas *patrones*), es decir, un tipo de sintagmas rudimentarios, como por ejemplo **art–nom** o **art–nom–adj**. Las secuencias consideradas por el módulo forman el *catálogo* de patrones. El funcionamiento del subprograma se basa en un esquema patrón–acción:

- Lee el texto (analizado y ya desambiguado) de izquierda a derecha, categoría léxica a categoría léxica.
- Busca, en la posición actual de la frase, el patrón más largo que concuerda con un patrón de su catálogo (por ejemplo, si en la posición actual se lee "una señal inequívoca…", elige **art–nom–adj** en vez de **art–nom**).
- Opera sobre este patrón (propagación de género y número, reordenamiento, cambios léxicos) siguiendo las reglas asociadas a él.
- Continúa inmediatamente detrás del patrón tratado (no vuelve a visitar las palabras sobre las cuales ha operado).

Cuando no se detecta ningún patrón en la posición actual, se traduce literalmente una palabra y se vuelve a iniciar el proceso. Los fenómenos "a la larga" como la concordancia sujeto–predicado son algo más difíciles de tratar; se usan variables de *estado*, una especie de *memoria* que recuerda cierta información a lo largo del proceso.

El subprograma de tratamiento de patrones se genera automáticamente a partir de un fichero de reglas que especifica los patrones y las acciones asociadas. Este es muy probablemente el subprograma más lento (unos pocos miles de palabras por segundo).

A.3. Cuestiones y ejercicios

Estos ejercicios pueden servir para repasar los conceptos tratados en este apéndice.

1. ¿Cuál de estas tres tareas es más difícil en un sistema de traducción automática español–catalán?

   a) Decidir la traducción del pronombre español *se* (puede ser *se*, *li* o *els*).
   b) Detectar las formas de *tener que* y traducirlas por *haver de*.
   c) Hacer el análisis morfológico de verbos seguidos de enclíticos como por ejemplo *estudiémonoslos* o *dándoselo*.

2. Indica cuál de estas tres es la fuente más importante de homografía (ambigüedad léxica categorial) del español:

   a) Las coincidencias de algunas formas de algunos nombres y de algunos adjetivos con ciertas formas conjugadas de algunos verbos.
   b) Las coincidencias de algunas formas de nombres con preposiciones.
   c) Las coincidencias de algunas formas de nombres con adverbios.

3. El catalán no tiene ninguna construcción equivalente al *cuyo* español. En traducción automática del español al catalán, una alternativa interesante es poner primero el sintagma nominal que sigue al *cuyo* y después, una forma de *del qual* que concuerde con el antecedente. ¿Se puede hacer siempre correctamente esta operación en un sistema de traducción automática que no haga análisis sintáctico?

   a) Sí, basta con hacer el análisis morfológico.
   b) No, porque hay que determinar bien la longitud del sintagma nominal que sigue a *cuyo* para poder poner *del qual* en la posición correcta.
   c) No, porque *cuyo* no tiene un equivalente morfológico en español.

A.4. Soluciones

1. (a)
2. (a)
3. (b)

Bibliografía

AECMA (2007). AECMA Simplified English. http://www.simplifiedenglish-aecma.org/Simplified_English.htm.

Ageeva, E., Forcada, M., Tyers, F., Pérez-Ortiz, y J.A. (2015). Evaluating machine translation for assimilation via a gap-filling task. En *Proceedings of EAMT 2015, The Eigtheenth Annual Conference of the European Association for Machine Translation (Antalya, May 11-13, 2015)*, pages 137–144.

Alcaraz Varó, E. y Martinez Linares, M. (1997). *Diccionario de lingüística moderna*. Ariel, Barcelona.

Almqvist, I. y Sågvall Hein, A. (1996). Defining ScaniaSwedish — a controlled language for truck maintenance. En *CLAW 96, Proceedings of the First International Workshop on Controlled Language Applications (Leuven)*, pages 159–164.

Arnold, D. (2003). En Somers, H., editor, *Computers and Translation: A translator's guide*, chapter Why translation is difficult for computers, pages 119–142. John Benjamins, Amsterdam i Philadelphia.

Arnold, D., Balkan, L., Meijer, S., Humphreys, R., y Sadler, L. (1994). *Machine Translation: An Introductory Guide*. NCC Blackwell, Oxford. Available as http://clwww.essex.ac.uk/~doug/MTbook/.

Arnold, D., Sadler, L., y Humphreys, R. (1993). Evaluation: an assessment. *Machine Translation*, 8:1–24.

Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., y Roossin, P. S. (1990). A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Canals-Marote, R., Esteve-Guillen, A., Garrido-Alenda, A., Guardiola-Savall, M., Iturraspe-Bellver, A., Montserrat-Buendia, S., Ortiz-Rojas, S., Pastor-Pina, H., Perez-Antón, P., y Forcada, M. (2001a). El sistema de traducción automática castellano-catalán interNOSTRUM. *Procesamiento del Lenguaje Natural*, 27:151–156. XVII Congreso de la Sociedad Española de Procesamiento del Lenguaje Natural, Jaén, España, 12–14.09.2001.

Canals-Marote, R., Esteve-Guillen, A., Garrido-Alenda, A., Guardiola-Savall, M., Iturraspe-Bellver, A., Montserrat-Buendia, S., Ortiz-Rojas, S., Pastor-Pina, H., Perez-Antón, P., y Forcada, M. (2001b). The Spanish-Catalan machine translation system interNOSTRUM. En *Proceedings of MT Summit VIII: Machine Translation in the Information Age*, pages 73–76. Santiago de Compostela, Spain, 18–22.09.2001.

Carl, M., Iomdin, L. L., Pease, C., y Streiter, O. (2001). Towards a dynamic linkage of example-based and rule-based machine translation. *Machine Translation*, 15(3):223–257.

Chomsky, N. (1996). *The minimalist program*. MIT Press, Cambridge, Massachusetts.

Doherty, S., Kenny, D., y Way, A. (2012). A user-based usability assessment of raw machine translated technical instructions. En *The 10th Biennial Conference of the Association for Machine Translations in the Americas (AMTA 2012)*, San Diego, California, USA. 28 oct. – 1 nov. 2012.

Don, J., Kerstens, J., Ruys, E., y Zwarts, J. (1996). Lexicon of linguistics. `http://www-uilots.let.ruu.nl/~Hans.Leidekker/lexicon/ll.html`.

Douglas, S. y Hurst, M. (1996). Controlled language support for perkins approved clear english (pace). En *Proceedings of the First International Workshop on Controlled Language Applications*, volume 93, page 105. Citeseer.

Fité, R. (2006). El periódico, una experiencia en traducció automàtica. *Tradumàtica*.

Flamand, J. (1983). *Écrire et traduire: sur la voie de la création*. Editions du Vermillion, Ottawa.

Forcada, M. (2000). Learning machine translation strategies using commercial systems: discovering word-reordering rules. En *Proceedings of MT 2000 (Exeter, November 2000)*.

Forcada, M. y Pérez-Ortiz, J. (2009). *Informática Aplicada a la Traducció: notes de classe amb exercicis i problemes resolts*. Alacant.

Forcada, M. L., Ginestí-Rosell, M., Nordfalk, J., O'Regan, J., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Sánchez-Martínez, F., Ramírez-Sánchez, G., y Tyers, F. M. (2011). Apertium: a free/open-source platform for rule-based machine translation. *Machine translation*, 25(2):127–144.

Hovy, E. (1993). How MT works. *Byte*, (gener):167–176.

Huijsen, W.-O. (1998). Controlled language–an introduction. En *Proceedings of CLAW*, volume 98, pages 1–15.

Hutchins, J. (1995). Machine translation: a brief history. En Koerner, E. y R.E.Asher, editors, *The concise history of the language sciences: from the Sumerians to the cognitivists*, pages 431–445. Pergamon.

Hutchins, J. (1996). Evaluation of machine translation and translation tools. En Cole, R., editor, *Survey of the State of the Art in Human Language Technology*. (disponible per internet: `http://www.cse.ogi.edu/CSLU/HLTsurvey/HLTsurvey.html`).

Hutchins, J. (2001). Machine translation over fifty years. *Histoire Epistémologie Langage*, 23(1):7–31.

Hutchins, W. y Somers, H. (1992). *An introduction to machine translation*. Academic Press. (hi ha una traduucció al castellà, *Introducción a la traducción automática*, editada por Visor en 1995).

Ide, N. y Veronis, J. (1998). Introduction to the special issue on word sense disambiguation: the state of the art. *Computational Linguistics*, 24:1–40.

Isabelle, P., Dymetman, M., Foster, G., Jutras, J., Macklovitch, E., Perrault, F., Ren, X., y Simard., M. (1993). Translation analysis and translation automation. En *Proceedings of the 1993 conference of the Centre for Advanced Studies on Collaborative research:, vol. 2, «Distributed computing»*, pages 1133–1147. IBM Press.

Jakobson, R. (1966). On the linguistic aspects of translation. En Brower, R., editor, *On translation*. Oxford Univ. Press, Oxford.

Jones, D., Herzog, M., Ibrahim, H., Jairam, A., Shen, W., Gibson, E., y Emonts, M. (2007). ILR-based MT comprehension test with multi-level questions. En *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 77–80. Association for Computational Linguistics.

Koehn, P. (2010). *Statistical Machine Translation*. Cambridge University Press.

Kohl, J. R. (2008). *The Global English Style Guide: Writing Clear, Translatable Documentation for a Global Market*. SAS Institute.

Krauwer, S. (1993). Evaluation of MT systems: a programmatic view. *Machine Translation*, 8.

Lappin, S. y Leass, H. J. (1994). An algorithm for pronominal anaphora resolution. *Computational linguistics*, 20(4):535–561.

Lewis, D. (1997). MT evaluation: science or art? *Machine Translation Review*, 6:25–36.

Lyovin, A. (1997). *Languages of the world*. Oxford Univ. Press, Oxford.

Masterman, M. (1967). *Machine Translation*, chapter Mechanical pidgin translation: An estimate of the research value of "word-for-word" translation into a pidgin language, rather than into the full normal form of an output language. North Holland.

Minnis, S. (1994). A simple and practical method for evaluation machine translation quality. *Machine Translation*, 9:133–149.

Mira i Gimènez, M. y Forcada, M. L. (1998). Understanding PC-based machine translation systems for evaluation, teaching and reverse engineering: the treatment of noun phrases in Power Translator. *Machine Translation Review (British Computer Society)*, 7:20–27. (available at `http://www.dlsi.ua.es/~mlf/mtr98.ps.Z`).

Newton, J. (1992). The Perkins experience. En *Computers in Translation: a practical appraisal*. Routledge, Londres.

Nida, E. (1966). Principles of translation exemplified by Bible translation. En Brower, R., editor, *On translation*. Oxford Univ. Press, Oxford.

O'Regan, J. y Forcada, M. L. (2013). Peeking through the language barrier: the development of a free/open-source gisting system for basque to english based on `apertium.org`. *Procesamiento del Lenguaje Natural*, 51:15–22.

O'Brien, S. (2003). Controlling controlled english. an analysis of several controlled language rule sets. *Proceedings of EAMT-CLAW*, 3:105–114.

Radford, A., Atkinson, M., Britain, D., Clahsen, H., y Spencer, A. (1999). *Linguistics: an introduction*. Cambridge Univ. Press, Cambridge.

Radford, A., Atkinson, M., Britain, D., Clahsen, H., y Spencer, A. (2009). *Linguistics: an introduction, 2nd. ed.* Cambridge Univ. Press, Cambridge.

Ramos, J. R. (1992). *Introducció a la sintaxi*. Tàndem, València.

Sager, J. C. (1993). *Language engineering and translation: consequences of automation*. Benjamins, Amsterdam.

Samuelson-Brown, G. (1996). New technology for translators. En Owens, R., editor, *The translator's handbook*. Aslib, Londres, 3a. edició.

Schwitten, R. (2007). Controlled natural languages. `http://www.ics.mq.edu.au/ rolfs/controlled-natural-languages/`.

Somers, H. y Rutzler, C. (1996). Machine translation. En Owens, R., editor, *The translator's handbook*. Aslib, Londres, 3a. edició.

Tellier, I. (2000). Semantic-driven emergence of syntax: the principle of compositionality upside-down. En *Proc. 3rd Conference on the The Evolution of Language*, pages 220–224, Paris.

Tuson, J. (1999). *¿Com és que ens entenem? (si és que ens entenem)*. Empúries, Barcelona.

Vandooren, F. (1993). Divergences de traduction et architectures de transfert. En P., B. y Clas, A., editors, *La traductique*. Presses Univ. Montréal, Montréal.

Wojcik, R. y Hoard, J. (1996). Controlled languages in industry. En Cole, R., editor, *Survey of the State of the Art in Human Language Technology*. (disponible per internet: `http://www.cse.ogi.edu/CSLU/HLTsurvey/HLTsurvey.html`).