Apertium, una plataforma de código abierto para el desarrollo de sistemas de traducción automática

Carme Armentano-Oller, Antonio M. Corbí-Bellot, Mikel L. Forcada,
Mireia Ginestí-Rosell, Marco A. Montava Belda, Sergio Ortiz-Rojas,
Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez y Felipe Sánchez-Martínez

Grup Transducens
Departament de Llenguatges i Sistemes Informàtics
Universitat d'Alacant, E-03071 Alacant

{carmentano,acorbi,mlf,mginesti}@dlsi.ua.es
{amontava,sortiz,japerez,gramirez,fsanchez}@dlsi.ua.es

http://transducens.dlsi.ua.es

Resumen Uno de los principales retos de la informática para las próximas décadas es el desarrollo de sistemas capaces de procesar eficazmente el lenguaje natural (o lenguaje humano). Dentro de este campo, los sistemas de traducción automática, encargados de traducir un texto escrito en un idioma a una versión equivalente en otro idioma, reciben especial atención dado, por ejemplo, el caracter multilingüe de sociedades como la europea. La automatización de dicho proceso es particularmente compleja porque los programas han de enfrentarse a características del lenguaje natural, como la ambigüedad, cuyo tratamiento algorítmico no es factible, de modo que una mera aproximación o automatización parcial del proceso ya se considera un éxito. Los programas de traducción automática han sido tradicionalmente sistemas cerrados, pero en los últimos tiempos la tendencia marcada por el software libre ha llegado también a este campo. En este artículo describimos Apertium, `apertium.org`, una plataforma avanzada de código abierto, con licencia GNU GPL, que, gracias al desacoplamiento que ofrece entre datos y programas permite desarrollar cómodamente nuevos traductores automáticos.
La plataforma Apertium ha sido desarrollada por el grupo de investigación Transducens de la Universitat d'Alacant en el marco de varios proyectos de colaboración con universidades y empresas de España en los que, además de los programas que conforman el motor de traducción, se han confeccionado datos lingüísticos abiertos para la traducción automática catalán–español, gallego–español, portugués–español, francés–catalán, inglés–catalán y occitano–catalán. Tanto la plataforma en la que se integra el motor de traducción como los datos para estos pares de lenguas están disponibles para su descarga en `sf.net/projects/apertium/` y para su evaluación en línea en `xixona.dlsi.ua.es/prototype/`.

Palabras clave: Apertium, traducción automática de código abierto, datos lingüísticos de código abierto, procesamiento del lenguaje natural

1. Introducción

Los sistemas de *traducción automática*, encargados de traducir un texto escrito en un idioma a una versión equivalente en otro idioma, son cada vez más demandados dado el carácter multilingüe de las complejas redes sociales, organizativas o comerciales actuales. La automatización del proceso de traducción es particularmente compleja porque los programas han de enfrentarse a características del lenguaje natural, como la ambigüedad, cuyo tratamiento algorítmico no es factible, de modo que una mera aproximación o automatización parcial del proceso ya se considera un éxito.

Los programas de traducción automática han sido tradicionalmente sistemas cerrados, tanto a nivel de código como de datos (pese a que algunos de ellos pueden utilizarse, con ciertas restricciones, en internet), lo que impide adaptarlos a nuevos pares de lenguas o dominios, integrarlos con otras aplicaciones, o usar sus recursos en proyectos de investigación o desarrollo. La complejidad inherente a este tipo de sistemas, que requiere un esfuerzo considerable de implementación, junto a la propia tendencia del mercado, han sido durante muchos años las principales razones de la falta de alternativas abiertas. Sin embargo, en los últimos tiempos el cambio de enfoque provocado por el software libre está llegando tímidamente también a este campo.

En este artículo describimos Apertium, uno de los principales exponentes de sistemas de traducción automática de código abierto. Apertium[1] es una plataforma avanzada de código abierto, con licencia GNU GPL,[2] que permite prototipar y construir traductores automáticos eficientes sin necesidad de tener que comenzar su desarrollo desde cero. Apertium ha dejado atrás la fase de prototipado y ya es un sistema de alto interés tanto comercial como científico. Por un lado, Apertium está siendo usado por empresas en proyectos reales tales como la edición multilingüe de noticias en prensa; por otro lado, la completa disponibilidad de la plataforma y de los datos asegura la reproducibilidad científica de las investigaciones publicadas en el área de la traducción automática que se basen en Apertium.

El resto del artículo presenta los siguientes contenidos. En la sección 2 se introduce el concepto de traducción automática, sus principales ámbitos de aplicación y algunas ideas acerca de por qué es un clásico problema "difícil" en informática; además, se introducen las principales técnicas con las que se diseñan los sistemas de traducción automática y las aproximaciones actuales basadas en código abierto. En la sección 3 se describe Apertium, una plataforma de código abierto formada por programas y herramientas que permiten desarrollar e implementar sistemas de traducción automática de forma eficiente. Finalmente, las secciones 4 y 5 presentan las oportunidades que se abren a nivel económico y científico, respectivamente, ante la disponibilidad de sistemas abiertos como Apertium. El artículo termina con unas conclusiones a modo de recapitulación.

[1] `http://www.apertium.org`
[2] `http://www.gnu.org/copyleft/gpl.html`

2. La traducción automática

La *traducción automática* consiste en la obtención de un texto *equivalente* (esto es, que preserve el contenido) en una *lengua destino* a partir de un texto en una *lengua origen*. Estamos hablando, en cualquier caso, de lenguas *naturales* (como, por ejemplo, el español, el inglés o incluso el esperanto), no de lenguajes informáticos (como, por ejemplo, Java o XML); la traducción entre lenguajes informáticos es una cuestión totalmente diferente (aunque puede compartir algunas técnicas básicas con la traducción automática) que se aborda en el campo del diseño de compiladores.

La traducción automática ha sido objeto de estudio desde los primeros tiempos de la informática en los años cincuenta y a día de hoy es un problema solo parcialmente resuelto. Es una de las áreas que más atención recibe dentro del *procesamiento del lenguaje natural*; este consiste en el estudio de aspectos como la formalización, la generación y la comprensión del lenguaje natural en todas sus formas, para, por ejemplo, permitir la realización de búsquedas en internet utilizando un lenguaje coloquial o la interacción con el ordenador a través de la voz.

2.1. ¿Por qué la traducción automática es difícil?

La ciencia actual no es capaz de expresar de manera formal, mediante reglas, todos los mecanismos que subyacen a los lenguajes naturales ni los procesos mentales involucrados en su uso. Esta imposibilidad para obtener una caracterización precisa, común a otros muchos campos de la inteligencia artificial, es uno de los principales argumentos que pueden darse para explicar la dificultad de escribir un programa de ordenador que traduzca textos. Sin embargo, podría aducirse que un entendimiento de los mecanismos profundos del funcionamiento del lenguaje natural puede no ser necesario para remedar el uso que las personas hacemos de él. Aún así, son muchos los elementos que hacen que la traducción automática siga siendo un problema difícil de resolver. Arnold [3] clasifica estos problemas en los siguientes grupos:

1. *La forma no determina completamente el contenido.* Esto se debe a la presencia de *ambigüedad* a distintos niveles: si decimos “el gato está debajo del coche”, ¿a qué nos estamos refiriendo, a un felino o a una herramienta?; si decimos, “he visto a tu hermano con el telescopio”, ¿quién de los dos es el que tiene un telescopio? Muchas veces es necesario *resolver* estas ambigüedades de cara a obtener una traducción correcta; por ejemplo, ¿cuál es el pronombre personal que ha de aparecer en la traducción al inglés de “tu amigo le dijo la verdad”, el pronombre *him* o el pronombre *her*? Aunque en ocasiones el contexto puede ayudar a la hora de resolver una ambigüedad, es muy complicado hacer que un programa de ordenador sea capaz de analizar siempre el contexto adecuadamente para tomar una decisión. Muchas veces, lo que se necesita para resolver estos casos es lo que llamamos *conocimiento del mundo*, que los humanos poseemos y que es muy difícil sistematizar e introducir en un programa de ordenador.

2. *El contenido no determina completamente la forma.* Existen multitud de formas de expresar en una lengua un contenido dado y para que un ordenador no deba enfrentarse a la complejidad que esto acarrea, es necesario imponer estrategias que reduzcan las posibilidades, aunque sea a costa de perder expresividad.
3. *Distintas lenguas usan estructuras diferentes para expresar las mismas cosas.* Consideremos, la frase "me gustan las manzanas"; su traducción al inglés es "I like apples", donde puede verse cómo el sujeto de la frase en español (*las manzanas*) se ha transformado en un complemento directo en el caso del inglés (*apples*). Aunque este es un ejemplo relativamente sencillo, en general, las estructuras usadas por lenguas distintas pueden ser tan divergentes que hagan que una simple traducción directa sea ininteligible.
4. Finalmente, como se ha comentado al inicio de este apartado, es difícil caracterizar con la precisión necesaria los principios involucrados en el uso del lenguaje. Existen, por otro lado, sistemas de traducción automática que mediante técnicas estadísticas de aprendizaje computacional son capaces de inferir nuevas traducciones a partir de ejemplos, pero la cantidad necesaria de estos ejemplos de traducciones es ingente y no siempre están disponibles para un par de lenguas dado.

Estos problemas se reducen ostensiblemente cuando las lenguas implicadas en la traducción están *emparentadas*. En ese caso las afinidades a nivel morfológico, sintáctico y semántico simplifican el diseño de estos sistemas y permiten llegar fácilmente a traducciones en las que solo un 5 % del texto es incorrecto.

Por otro lado, existen tipos de textos que pueden ser más fácilmente traducidos (cartas comerciales, manuales de intrucciones, textos económicos) y otros cuya traducción automática es a día de hoy totalmente inviable (poesía, por ejemplo).

2.2. Usos de la traducción automática

Las distintas aplicaciones de la traducción automática se pueden dividir en tres grandes grupos: *asimilación*, *comunicación* y *diseminación*. En situaciones de *asimilación*, la traducción resultante se utiliza para hacerse una idea general del contenido del texto original, en este caso la calidad de la traducción no es relevante en tanto el lector pueda satisfacer este deseo. Un uso de la traducción automática cercano al anterior es el que se produce cuando dos o más personas que hablan idiomas distintos se *comunican* (vía chat o correo electrónico, por ejemplo) a través de un sistema de traducción automática. Finalmente, para la *diseminación* se precisa de una traducción de mayor calidad ya que el texto traducido se difunde públicamente. En este último caso, la calidad de la traducción resultante se puede conseguir de diversas formas:

- restringiendo el lenguaje origen, bien reduciendo el dominio lingüístico (por ejemplo, centrándose en el lenguaje de los partes meteorológicos o de las cartas comerciales), bien controlando el tipo de construcciones o palabras que

pueden ser usados (por ejemplo, evitando el uso de oraciones subordinadas o de palabras polisémicas en el texto origen); en el primer caso se habla de traducción de *sublenguajes* y en el segundo caso de traducción de *lenguajes controlados*; lo más habitual es combinar ambas estrategias;
- corrigiendo la traducción automática resultante, lo que se conoce como *postedición*, y que constituye la forma más habitual de obtener traducciones adecuadas para la diseminación.

En cualquier caso, cuanto mejor sea el sistema de traducción automática menor será el esfuerzo humano necesario para obtener una traducción adecuada. A día de hoy, la *traducción de gran calidad completamente automatizada* es todavía una quimera, pero esto no invalida el uso de la *traducción automática* en tareas de diseminación, ya que en ciertos casos sale más rentable posteditar un texto traducido automáticamente que traducirlo manualmente desde cero. En el caso de la asimilación, por otro lado, muchos internautas utilizan sistemas de traducción automática para desenvolverse adecuadamente en servicios que no están disponbiles en los idiomas que conocen; por ejemplo, para realizar compras en internet.

Evidentemente, en un entorno como el actual cada vez más globalizado y marcado por un vertiginoso ritmo de producción y consumo de información, la demanda de sistemas de traducción automática se ha disparado a la vez que surgen nuevas especializades dentro de la profesión de traductor, como es la del *posteditor*. Por otro lado, el interés de algunas administraciones y el valor que en la nueva economía adquieren las minorías lingüísticas está provocando la aparición paulatina de sistemas de traducción que involucran lenguas minoritarias; considérese, por ejemplo, la reciente publicación de un sistema de traducción automática basado en Apertium entre el catalán y el aranés (lengua cooficial en Cataluña, usada en el pequeño territorio del Valle de Arán) [2].

2.3. Técnicas de traducción automática

A la hora de construir sistemas de traducción automática se suelen seguir dos grandes enfoques: el de los sistemas *basados en reglas* y el de los sistemas basados en *corpus* (colecciones de documentos); los ejemplos más destacados de sistemas de traducción automática basados en corpus son los sistemas *estadísticos* y los sistemas *basados en ejemplos*. Por descontado, también son posibles enfoques híbridos.

En un sistema *basado en reglas* los datos lingüísticos (básicamente diccionarios monolingües, diccionarios bilingües y reglas de transferencia, que recogen las transformaciones estructurales necesarias para pasar de una lengua a otra) se recopilan manualmente y se codifican adecuadamente para que puedan ser usados por el motor de traducción. Los sistemas *basados en corpus* utilizan cadenas o patrones de traducción inferidos automáticamente a partir de las regularidades observadas en *corpus paralelos*, esto es, corpus que contienen documentos en la lengua origen y su correspondiente traducción en lengua destino.

Los sistemas basados en reglas más habituales son los sistemas de *traducción automática por transferencia*, que funcionan mediante tres fases bien diferenciadas: análisis, transferencia y generación.

- La fase de *análisis* produce a partir de la frase en lengua origen una representación intermedia dependiente de la lengua origen.
- La fase de *transferencia* convierte la representación intermedia anterior en una nueva representación intermedia que, esta vez, es dependiente de la lengua destino, pero no de la de origen.
- La fase de *generación* produce una frase en lengua destino a partir de la representación intermedia obtenida en la fase anterior.

2.4. Traducción automática de código abierto

Un traductor automático de código abierto[3] puede ser usado, copiado, estudiado, modificado y redistribuido con la única restricción de que el código fuente ha de estar siempre disponible. Para que el sistema sea de código abierto basta con acompañarlo con una licencia compatible con estos principios (tal como la licencia GNU GPL, la licencia Mozilla, las licencias de tipo BSD, etc.); si no se explicitan estos principios con una licencia, se aplican las restricciones habituales de los programas comerciales.

Las ideas del código abierto se pueden aplicar a elementos distintos del software siempre que esté claro qué es el *código fuente* del producto final. Por ejemplo, en muchos sistemas de traducción automática la forma original "textual" de los diccionarios se convierte (o se *compila*) en una forma "binaria" mucho más eficiente pero más complicada de editar; en este caso, las entradas del diccionario en forma textual se pueden considerar como el código fuente del diccionario.

Independientemente del enfoque, de entre los comentados en el apartado 2.3, que utilice un sistema de traducción automática, en todos los sistemas se pueden distinguir una serie de componentes comunes (aunque en función del sistema particular algunos componentes pueden no estar presentes o estar integrados en otros):

- el motor de traducción,
- los datos lingüísticos en forma "textual",
- los *compiladores* (en un sentido general) que transforman los datos anteriores en una forma "binaria" más eficiente usada por el motor, y
- las herramientas de mantenimiento de los datos lingüísticos.

[3] Para los propósitos de este trabajo consideraremos *código abierto* (término acuñado por la Open Source Initiative, `http://www.opensource.org`) y *software libre* (término acuñado por la Free Software Foundation, `http://www.fsf.org`) como términos equivalentes; del mismo modo, usaremos como términos opuestos a estos los de *código cerrado* o *software comercial*. La terminología en el campo es bastante ambigua: por ejemplo, algunos programas comerciales tienen su código fuente publicado, pero no pueden considerarse como de código abierto, según la definición de la Open Source Initiative, debido a determinadas cláusulas restricitivas de la licencia.

Para que un sistema de traducción automática se pueda considerar *abierto*, es necesario que el código fuente del motor de traducción y de los datos lingüísticos se distribuya con licencias adecuadas; debe tenerse en cuenta que es mucho más probable que los usuarios de un sistema de traducción automática de código abierto realicen modificaciones en los datos lingüísticos que que lo hagan sobre el código fuente del motor de traducción. Por otro lado, si se realizan determinados cambios en el código fuente del motor, puede ser necesario también modificar los compiladores, por lo que su código también debe ser abierto. Finalmente, si se alteran algunas características de los datos lingüísticos puede ser necesario cambiar las herramientas de mantenimiento. En definitiva, los cuatro componentes deben idealmente ser abiertos para que el sistema de traducción automática pueda considerarse estrictamente como tal.

Aunque los ejemplos de sistemas de traducción automática de código abierto no son abundantes, en los últimos años han aparecido varios de ellos. En este artículo describimos Apertium, un sistema de traducción automática totalmente abierto, según la definición anterior.[4] En el terreno de los sistemas comerciales, la mayoría de los traductores automáticos han estado tradicionalmente basados en reglas[5], pero cada vez son más los sistemas basados en corpus[6].

3. Apertium, una plataforma de código abierto para la traducción automática

OpenTrad Apertium[7] (Apertium para abreviar) es una plataforma de código abierto que incluye las herramientas y programas necesarios para construir y ejecutar sistemas de traducción automática basados en reglas, aunque alguno de sus módulos son de tipo estadístico o híbrido; los traductores construidos con Apertium pueden llegar a traducir decenas de miles de palabras por segundo.[8] Originalmente, Apertium se diseñó como un sistema de transferencia superficial orientado a lenguas emparentadas (como catalán–español, portugués–español, checo–eslovaco, sueco–danés, bahasa indonesia–bahasa malasia, etc.), pero desde

[4] Otros sistemas total o parcialmente abiertos son OpenLogos, la versión de código abierto del sistema comercial basado en reglas Logos, `http://logos-os.dfki.de`, o el sistema de traducción estadística Moses, `http://www.statmt.org/moses/`; en el pasado se realizaron algunos intentos de desarrollar sistemas de traducción automática de código abierto, que quedaron incompletos, como GPLTrans, `http://www.translator.cx`, Traduki, `http://traduki.sourceforge.net`, o Linguaphile, `http://linguaphile.sourceforge.net`.

[5] Por ejemplo, Reverso, `http://www.reverso.net`, o Babylon, `http://www.babylon.com`.

[6] Por ejemplo, AutomaticTrans, `http://www.automatictrans.es`, o Language Weaver, `http://www.languageweaver.com`.

[7] La plataforma OpenTrad, `http://www.opentrad.org`, está formada por dos arquitecturas de código abierto: Apertium y Matxin, `http://matxin.sourceforge.net/`; esta última utiliza un sistema de transferencia sintáctica más profundo que el de Apertium.

[8] En un ordenador personal de sobremesa.

su versión 2, publicada en diciembre de 2006, el motor de traducción usa un sistema de transferencia más avanzado que permite manejar rasgos lingüísticos presentes en lenguas no tan emparentadas (como español–inglés).

Apertium es el resultado de diversos proyectos de subvención pública en los que han participado hasta el momento diferentes universidades (Universitat d'Alacant, Universidade de Vigo, Universitat Politècnica de Catalunya, Euskal Herriko Unibertsitatea y Universitat Pompeu Fabra) y empresas (Eleka Ingeniaritza Linguistikoa, Imaxin Software, Elhuyar Fundazioa y Prompsit Language Engineering). La plataforma Apertium se basa en la experiencia y conocimientos adquiridos por el grupo Transducens[9] de la Universitat d'Alacant en el desarrollo del sistema catalán–español interNOSTRUM[10] y del traductor portugués–español Tradutor Universia[11], que no son, sin embargo, de código abierto.

Apertium está escrito principalmente en C++ y puede compilarse y ejecutarse sobre el sistema operativo Linux,[12] aunque su adaptación a otros sistemas operativos no debería plantear, teóricamente, muchos problemas; en cualquier caso, el programa puede instalarse fácilmente en un servidor de aplicaciones de internet para ser accedido de forma remota. Hasta la fecha se han desarrollado datos lingüísticos funcionales para los siguientes pares de lenguas (en ambos sentidos de traducción): catalán–español, gallego–español, portugués–español, aranés–catalán, catalán–francés y catalán–inglés;[13] además, los autores de este artículo tienen conocimiento de desarrollos iniciales para rumano–español y sueco–danés. Para algunos de los pares de lenguas mencionados, las tasas de error[14] se sitúan entre el 5 % y el 10 % con textos periodísticos; estos resultados aún pueden mejorarse fácilmente aumentando la cobertura de los diccionarios o de las reglas de transferencia.

El motor de traducción de Apertium, las herramientas auxiliares, la documentación correspondiente y la mayoría de los datos lingüísticos desarrollados hasta la fecha para Apertium pueden descargarse desde la web del proyecto en `http://apertium.sourceforge.net`. Existe, además, una web donde pueden usarse en línea los distintos traductores automáticos disponibles para Apertium: `http://xixona.dlsi.ua.es/prototype/`.

A continuación se describe brevemente algunas de las características de Apertium; el lector interesado en más detalles puede consultar la documentación de la plataforma o alguno de los artículos publicados sobre ella [1].

[9] `http://transducens.dlsi.ua.es`
[10] `http://www.internostrum.com`
[11] `http://tradutor.universia.net`
[12] Para algunas de las distribuciones de Linux basadas en Debian existen paquetes precompilados presentes en los repositorios oficiales.
[13] Algunos de estos datos se distribuyen con licencia GNU GPL y otros con licencia Creative Commons, `http://www.creativecommons.org`.
[14] Calculadas como el procentaje de palabras que deben ser añadidas, eliminadas o modificadas para que la traducción resultante sea correcta.

3.1. El motor de traducción

Los traductores que pueden desarrollarse con la plataforma Apertium son sistemas de traducción automática por transferencia consistentes en una serie de módulos [7] dispuestos en cascada (véase la figura 1):

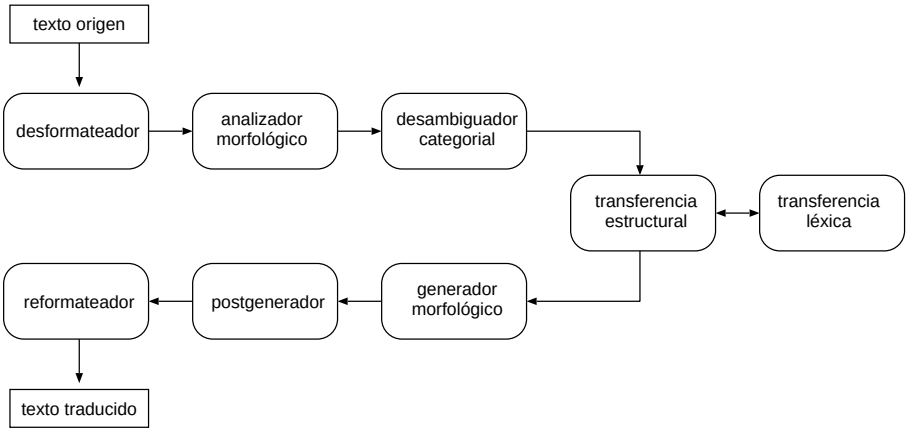

Figura 1. Los módulos de un sistema de traducción automática construido con la plataforma Apertium.

El desformateador, que separa el texto a traducir de la información de formato del documento (XML, HTML, etc.). La información de formato es *encapsulada* de manera que los módulos restantes la tratan como blancos entre palabras. Por ejemplo, ante el texto HTML en español:

```
es <em>una señal</em>
```

el desformatador encapsula las etiquetas HTML entre corchetes y proporciona como salida:

```
es [<em>]una señal[</em>]
```

Las secuencias de caracteres entre corchetes son tratadas por el resto de módulos como simples espacios en blanco entre palabras.

El analizador morfológico, que segmenta el texto en *formas superficiales* (las unidades léxicas tal como se presentan en los textos) y entrega para cada forma superficial una o más *formas léxicas* consistentes en un *lema* (la forma base comúnmente usada para las entradas de los diccionarios clásicos), la *categoría léxica* (nombre, verbo, preposición, etc.) y la información de flexión morfológica (número, género, persona, tiempo, etc.). La división de un

texto en formas superficiales presenta aspectos complejos debido a la existencia, por un lado, de contracciones (*del*, *teniéndolo*, *vámonos*) y, por otro, de unidades léxicas de más de una palabra (*a pesar de*, *echó de menos*). El analizador morfológico permite analizar estas formas superficiales complejas y tratarlas adecuadamente para que sean procesadas por los módulos posteriores. En el caso de las contracciones, el sistema lee una única forma superficial y da como salida una secuencia de dos o más formas léxicas (por ejemplo, la contracción *del* sería analizada como dos formas léxicas, una para la preposición *de* y otra para el artículo *el*). Las unidades léxicas de más de una palabra (*multipalabras*) son tratadas como formas léxicas individuales y, según su naturaleza, reciben un tratamiento específico.
Al recibir como entrada el texto de ejemplo proveniente del módulo anterior, el analizador morfológico proporcionaría como salida:

```
^es/ser<vbser><pri><p3><sg>$[ <em>]
^una/un<det><ind><f><sg>/unir<vblex><prs><1><sg>/unir
<vblex><prs><3><sg>$
^señal/señal<n><f><sg>$[</em>]
```

donde cada forma superficial es analizada como una o más formas léxicas. Así, *es* es analizada como una forma superficial con lema *ser*, mientras que *una* recibe tres análisis: lema *un*, determinante indefinido femenino singular; lema *unir*, verbo en presente de subjuntivo, primera persona del singular, y lema *unir*, verbo en presente de subjuntivo, tercera persona del singular.

El desambiguador léxico categorial, elige, usando un modelo estadístico (*modelo oculto de Markov* [4]), uno de los análisis de una palabra ambigua de acuerdo con su contexto; en el ejemplo utilizado, la palabra ambigua sería la forma superficial *una*, que puede analizarse de tres maneras diferentes. Las formas superficiales ambiguas por tener más de un lema, más de una categoría o representar más de una flexión son muy comunes (en las lenguas románicas, en torno a una de cada tres palabras) y son una fuente muy importante de errores de traducción en caso de elegir el equivalente incorrecto. El modelo estadístico se entrena sobre corpus suficientemente representativos de textos en lengua origen, aunque también puede emplearse información de la lengua destino de la traducción durante el entrenamiento [8].
El resultado tras la desambiguación léxica categorial del texto de ejemplo proporcionado por el analizador morfológico sería:

```
^ser<vbser><pri><p3><sg>$[ <em>]^un<det><ind><f><sg>$
^señal<n><f><sg>$[</em>]
```

en el que se ha elegido la forma léxica correcta, determinante, para la palabra *una*.

El módulo de transferencia léxica, que gestiona un diccionario bilingüe y es invocado por el módulo de transferencia estructural, lee cada forma léxica en lengua origen y entrega la forma léxica correspondiente en lengua destino. El diccionario contiene actualmente un único equivalente para cada

forma léxica en la lengua destino, pero próximamente la plataforma permitirá indicar más de una (polisemia) y un nuevo módulo se encargará de elegir el equivalente adecuado. Las multipalabras son traducidas como una unidad. En el ejemplo utilizado, cada una de las formas léxicas se traduciría al catalán de la siguiente manera:

```
ser<vbser> ⟶ ser<vbser>
un<det> ⟶ un<det>
señal<n><f> ⟶ senyal<n><m>
```

El módulo de transferencia estructural, que detecta y trata patrones de palabras (sintagmas) que exigen un tratamiento especial por causa de las divergencias gramaticales entre las lenguas (cambios de género y número, reordenamientos, cambios preposicionales, etc.). Este módulo lee de un archivo las reglas que describen la acción que debe realizarse para cada patrón. Ante la frase de ejemplo, el patrón formado por `^un<det><ind><f><sg>$ ^señal<n><f><sg>$` sería detectado por una regla determinante–nombre, que en este caso modificaría el género del determinante para que concuerde con el nombre; el resultado sería:

```
^ser<vbser><pri><p3><sg>$[ <em>]^un<det><ind><m><sg>$
^senyal<n><m><sg>$[</em>]
```

El generador morfológico , que genera a partir de la forma léxica en lengua destino una forma superficial flexionada adecuadamente. El resultado para la frase de ejemplo sería:

```
és[ <em>]un senyal[</em>]
```

El postgenerador, que realiza algunas operaciones ortográficas en lengua destino tales como contracciones y apostrofaciones. En la frase de ejemplo utilizada no hay que realizar ninguna contracción ni apostrofación.

El reformateador, que reintegra la información de formato original al texto traducido; el resultado para la frase de ejemplo sería la correcta conversión del texto a formato HTML:

```
és <em>un senyal</em>
```

Como se ha podido observar, los módulos del sistema se comunican entre sí mediante flujos de texto, lo que permite su evaluación independiente; de hecho, algunos módulos pueden ser utilizados de forma aislada en otras aplicaciones de procesamiento del lenguaje natural.

3.2. Los datos lingüísticos

Los datos lingüísticos de Apertium están completamente desacoplados del motor de traducción y se codifican mediante formatos basados en XML[15]; esto

[15] `http://www.w3.org/XML/`

permite su interoperabilidad (es decir, la posibilidad de utilizar los datos XML en entornos diferentes) y facilita la transformación y el mantenimiento. En gran medida, el éxito de un motor de traducción automática de código abierto depende de que otros colectivos acepten los formatos que utiliza; este es, de hecho, el mecanismo por el cual aparecen los estándares *de facto*. La aceptación puede verse facilitada por el uso de un formato interoperable basado en XML, así como por la disponibilidad de herramientas para gestionar los datos lingüísticos. También es cierto que, a su vez, la aceptación de los formatos depende considerablemente del éxito del sistema de traducción en sí. Los formatos XML para los diferentes tipos de datos lingüísticos de Apertium se definen mediante definiciones de tipo de documento (DTD) en XML diseñadas específicamente.

3.3. Los compiladores

La plataforma Apertium contiene *compiladores* que convierten los datos lingüísticos a la forma "binaria" eficiente que es usada por cada módulo. Los cuatro módulos de procesamiento léxico (el analizador morfológico, el módulo de transferencia léxica, el generador morfológico y el postgenerador) utilizan un único compilador, que genera una representación basada en *transductores de estados finitos* [5]. El módulo de transferencia estructural utiliza una representación de las reglas de transferencia en forma de *máquina de estados finitos*.

Estos compiladores no solo necesitan unos pocos segundos para compilar diccionarios con miles de entradas (lo que simplifica la labor de desarrollo, ya que el efecto en el sistema de un cambio en una regla de transferencia o en una entrada del diccionario se puede evaluar casi instantáneamente), sino que, además, las estructuras compiladas resultantes permiten que el motor pueda traducir a velocidades del orden de decenas de miles de palabras por segundo.

4. Modelos de negocio basados en Apertium

La plataforma Apertium, junto con la documentación y las herramientas de gestión de datos lingüísticos ya disponibles, ofrece buenas oportunidades para investigadores, traductores profesionales y empresas que trabajen en ingeniería lingüística u ofrezcan servicios lingüísticos.

Los usuarios individuales, los investigadores y las comunidades lingüísticas (al igual que las empresas privadas) pueden unir sus esfuerzos y contribuir al desarrollo de las tecnologías y los datos de Apertium; cualquier persona que tenga las habilidades informáticas y lingüísticas necesarias puede adaptar o ampliar Apertium para producir nuevos o mejores sistemas de traducción automática, incluso para nuevos pares lingüísticos.

El desarrollo en código abierto de software y datos garantiza el acceso libre e ilimitado a los recursos creados. Además, no es necesario crear estos recursos desde cero, puesto que el punto de partida de nuevos proyectos puede arrancar donde se quedaron proyectos anteriores.

Como el código y los datos de Apertium pueden reutilizarse, los investigadores y desarrolladores pueden centrarse en mejorarlos. Nuestra experiencia nos dice que reutilizar datos de pares de lenguas existentes acelera el desarrollo de datos para nuevos pares; por ejemplo, las reglas para la concordancia de género y número son básicamente las mismas para todos los pares de lenguas emparentadas que nuestro equipo ha desarrollado. Además, los diccionarios monolingües pueden utilizarse en más de un par de lenguas con pequeñas modificaciones.

4.1. Servicios ofertables en torno a Apertium

El software de código abierto permite a las empresas trabajar con nuevos modelos de negocio. En concreto, una empresa podría ofrecer un amplio rango de servicios en torno a Apertium [6], tales como:

- instalación y mantenimiento de servidores de traducción automática basados en Apertium;
- mantenimiento y ampliación de los datos lingüísticos;
- adaptación de los datos lingüísticos a dominios particulares o a variedades dialectales;
- adaptación de los datos lingüísticos para reducir el esfuerzo de postedición en los documentos de un cliente particular;
- construcción de datos lingüísticos para nuevos pares de lenguas;
- integración de Apertium en sistemas de gestión documental multilingües;
- desarrollo de extensiones, nuevos módulos y herramientas auxiliares para Apertium;
- traducción corregida por traductores humanos especializados en la postedición de Apertium; el conocimiento del sistema de traducción automática permite reducir el tiempo necesario para realizar postediciones de calidad y esto puede permitir, en función de la calidad del traductor automático, que estas traducciones se ofrezcan a precios inferiores a los que ofrece el mercado para traducciones humanas “desde cero”.

Además, algunos de los servicios anteriores también pueden ser ofrecidos por traductores profesionales autónomos que conozcan con suficiente detalle el funcionamiento del traductor, lo que abre nuevas oportunidades de trabajo para este colectivo.

Las empresas que han participado en algunos de los proyectos en los que se ha desarrollado Apertium[16] han comenzado a explotar las posibilidades de negocio que surgen en torno al traductor, y pese a que la primera versión estable de Apertium fue publicada a finales de 2005, ya existen casos reales de comercialización. Así, las ediciones en internet de dos de los principales periódicos de Galicia,[17] publicados originalmente en español, cuentan desde hace meses con

[16] Eleka Ingeniaritza Linguistikoa, `http://www.eleka.net`, Imaxin Software, `http://www.imaxin.com`, Elhuyar Fundazioa, `http://www.elhuyar.org`, y Prompsit Language Engineering, `http://www.prompsit.com`.

[17] La Voz de Galicia, `http://www.lavozdegalicia.es`, y El Correo Gallego, `http://www.elcorreogallego.es`.

una versión diaria en gallego basada en la tecnología de Apertium. Algunos organismos públicos, por otro lado, han comenzado a estudiar la posibilidad de utilizar Apertium como plataforma de traducción automática de cara a ofrecer servicios de traducción de alta velocidad por internet.

Un argumento esgrimido con cierta frecuencia es que este mercado sería más rentable si los sistemas fueran de código cerrado; al mismo tiempo, es habitual que los potenciales clientes sean reticentes a tecnologías abiertas como Apertium. Sin embargo, pueden enumerarse ciertas ventajas sobre los sistemas cerrados; una de las más significativas es que los clientes de sistemas de traducción automática de software abierto no ven a la empresa proveedora como una empresa de la que *dependen* tecnológicamente, sino, más bien, como *socios* tecnológicos, ya que la puerta siempre está abierta para, llegado el caso, contratar los servicios con otra empresa que los ofrezca. Si tanto los programas como los datos son de código abierto, esto no solo es posible, sino que genera una interesante competencia.

Además, para instituciones, entidades públicas y grandes empresas es incluso más interesante la contribución social que pueden ejercer al hacer modificaciones abiertas, mejorar los datos lingüísticos o añadir nuevas funcionalidades al software de código abierto desarrollado específicamente para ellas.

Esto les da una imagen muy positiva ante sus clientes y usuarios, puesto que no solo ofrecen un mejor servicio, sino que benefician a toda la comunidad.

5. Modelos de investigación basados en Apertium

El software de código abierto garantiza la reproducibilidad de los experimentos de investigación. Cuando se publican resultados de evaluación de un sistema de traducción automática al que se han icorporado nuevas técnicas, si los experimentos realizados no pueden ser reproducidos, los autores obligan a los lectores a confiar en sus afirmaciones. Si el software y los datos lingüísticos son de código abierto es posible reproducir los experimentos fácilmente, realizar nuevos experimentos y compararlos entre sí. La *reproducibilidad* es muy importante en la ciencia: es necesaria para poder comprobar o verificar los resultados experimentales y constituye una de las bases del progreso científico, ya que si un experimento concreto no puede repetirse, no se considera que proporcione una *prueba científica* sólida.

El software de código abierto fomenta la interacción colaborativa entre grupos de investigación que trabajen en la misma área. Tanto las administraciones públicas como las universidades deberían apoyar y financiar este tipo de trabajo colaborativo, que tiene un beneficio especial para grupos de investigación con pocos recursos. La perspectiva del código abierto estimula y facilita la presencia de grupos de investigación en programas locales, nacionales y europeos de investigación, desarrollo e innovación tecnológica.

El código abierto también puede facilitar la transferencia de nuevos avances de la comunidad investigadora a empresas interesadas en su aplicación. Este papel que pueden jugar los productos de código abierto debería ser tenido en cuenta

por los organismos públicos que promocionan la cooperación entre investigadores y empresas en su entorno socioeconómico.

El software de código abierto plantea retos comerciales a los investigadores que trabajen en nuevos métodos y técnicas. De hecho, el número de *spin-offs* de base tecnológica (nos referimos aquí a empresas creadas por investigadores como resultado de una actividad investigadora concreta) ha aumentado en los últimos años. Estas empresas no sólo tienen un producto a ofrecer y un catálogo de servicios relacionados, sino también el *know-how* desarrollado durante el trabajo investigador de sus miembros; además, al estar situados en un punto intermedio, pueden ofrecer mejores servicios en colaboración con universidades y empresas. Este es, por ejemplo, el caso de Prompsit Language Engineering,[18] una empresa *spin-off* creada por investigadores del grupo Transducens de la Universitat d'Alacant, que ofrece servicios alrededor de la plataforma Apertium.

6. Conclusiones

El software libre plantea nuevos retos tanto para las empresas como para los centros de investigación. Aunque la filosofía del código abierto acaba de aterrizar en el campo de la traducción automática, un campo concentrado hasta ahora exclusivamente en productos comerciales, ya comienza a vislumbrarse el enorme potencial de las tecnologías abiertas en el terreno de la traducción. Es posible, por tanto, un nuevo modelo de negocio orientado a la oferta y demanda de servicios en torno a los motores de traducción y los datos lingüísticos, más que a los programas y datos en sí mismos. Además, con el desarrollo de lo que se ha dado en llamar la web 2.0, aparecerán nuevas aplicaciones en internet que establecerán redes sociales para el desarrollo colaborativo de mejores sistemas de traducción automática que funcionen sobre sistemas abiertos. En este artículo hemos presentado Apertium, una plataforma completamente funcional de traducción automática desarrollada en la Universitat d'Alacant, que es a día de hoy uno de los principales exponentes de código abierto dentro del área.

Agradecimientos

Este trabajo ha sido parcialmente subvencionado por el Ministerio de Industria, Comercio y Turismo a través de los proyectos FIT-340101-2004-3, FIT-340001-2005-2 y FIT-350401-2006-5, por el Ministerio de Educación y Ciencia a través de los proyectos TIC2003-08681-C02-01 y TIN2006-15071-C03-01, y por la Generalitat de Catalunya a través del proyecto DURSI1-05I. Felipe Sánchez-Martínez disfruta de la ayuda para la formación de personal investigador BES-2004-4711, financiada por el Fondo Social Europeo y el Ministerio de Educación y Ciencia.

[18] `http://www.prompsit.com`

Referencias

1. Armentano-Oller, C., Carrasco, R. C., Corbí-Bellot, A. M., Forcada, M. L., Ginestí-Rosell, M., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Ramírez-Sánchez, G., Sánchez-Martínez, F., Scalco, M. A.: Open-source Portuguese-Spanish machine translation. En: Computational Processing of the Portuguese Language: 7th Workshop on Computational Processing of Written and Spoken Portuguese, PROPOR 2006. Lecture Notes in Artificial Intelligence 3960. Springer-Verlag (2006) 50–59
2. Armentano-Oller, C., Forcada, M. L.: Open-source machine translation between small languages: Catalan and Aranese Occitan. En: Strategies for developing machine translation for minority languages: 5th SALTMIL workshop on Minority Languages (2006) 51–54
3. Arnold, D.: Why machine translation is difficult for computers. In Somers, H. L. (coordinador): Computers and Translation: a translator's guide. John Benjamins, Amsterdam (2003) 119–142.
4. Cutting, D., Kupiec, J., Pealersen, J., Sibun, P.: A practical part-of-speech tagger. En: Proceedings of the Third Conference on Applied Natural Language Processing (1992) 133–140.
5. Garrido, A., Iturraspe, A., Montserrat, S., Pastor, H., Forcada, M. L.: A compiler for morphological analysers and generators based on finite-state transducers. En: Procesamiento del Lenguaje Natural, 25 (1999) 93–98.
6. Ramírez-Sánchez, G., Sánchez-Martínez, F., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Forcada, M. L.: Opentrad Apertium open-source machine translation system: an opportunity for business and research. En: Proceedings of the Twenty-Eighth International Conference on Translating and the Computer (2006)
7. Ginestí Rosell, M. (coordinadora): Documentación del sistema de código abierto Opentrad Apertium de traducción automática de transferencia sintáctica superficial. `http://apertium.svn.sourceforge.net/viewvc/apertium/` (febrero 2007).
8. Sánchez-Martínez, F., Pérez-Ortiz, J. A., Forcada, M. L.: Speeding up target language driven part-of-speech tagger training for machine translation. En: Proceedings of the 5th Mexican International Conference on Artificial Intelligence, MICAI 2006. Lecture Notes in Artificial Intelligence 4293. Springer-Verlag (2006) 844–854