

INTERFAZ PARA EL SISTEMA DE AYUDA A LA TRADUCCIÓN DEL PRHLT

María-Teresa González Cavero¹, Francisco Casacuberta Nolla¹

¹Dept. de Sistemas Informáticos y Computación
 Universidad Politécnica de Valencia
 46071 Valencia, Spain
 {mgonzalez,fcn}@dsic.upv.es

RESUMEN

Hoy en día hay muchos estudios realizados sobre traducción automática, pero no se ha llegado a obtener una calidad en la traducción suficiente como para que los textos traducidos automáticamente no sean revisados por un traductor humano. Por todo ello se han investigado otras técnicas como es la traducción automática asistida por ordenador. En estas técnicas, un traductor humano interactúa con un sistema de traducción automática que le sirve de herramienta de apoyo para ofrecerle dinámicamente una lista de traducciones que mejor complementan la frase que está traduciendo.

Para que el usuario interactúe con el sistema de traducción automática es necesario implementar una interfaz de ayuda a la traducción que sea fácil y cómoda de utilizar. En otros proyectos como el TT2 ya se creó una interfaz de este tipo y en evaluaciones humanas se comprobaba que la ganancia de productividad obtenida se encontraba entre un 15 y un 20%. En este trabajo proponemos otra interfaz basada en la del proyecto TT2. Lo que se pretende con esta interfaz es mejorar algunos aspectos de la interacción con el usuario que existían en la del proyecto TT2.

1. INTRODUCCIÓN

El procesamiento de lenguaje natural es una área de investigación de gran actividad y desarrollo que ha visto incrementada su relevancia y capacidad con los recientes avances de la informática.

Como resultado de la imposibilidad de obtener traducciones de alta calidad con los sistemas de traducción automática actuales, se introdujo una aproximación alternativa [1] que incorpora el conocimiento humano en el marco de la traducción automática. En este marco, llamado traducción asistida por ordenador (CAT¹), un traductor humano interactúa con un sistema de traducción automática que le sirve de herramienta de apoyo para ofrecerle dinámicamente una lista de traducciones que mejor

completan la frase que está traduciendo. De esta manera, el sistema automático aprovecha las correcciones del traductor humano para producir mejores traducciones.

El sistema de traducción asistida por ordenador que utilizaremos está basado en el paradigma de los modelos de estados finitos, más concretamente en transductores estocásticos de estados finitos (SFST²)[2]. Una característica interesante de los SFSTs es que pueden ser aprendidos eficientemente de forma automática a partir de muestras finitas como puede ser un corpus bilingüe [3].

En otros trabajos realizados en proyectos como el TT2 [4] se ha demostrado que tener un sistema de traducción automática asistida (llamados sistemas CAT) por traductores profesionales proveen mejores traducciones y en menor tiempo. En evaluaciones realizadas por traductores humanos profesionales se comprobó que había un aumento del 15-20% de la productividad al utilizar sistemas CAT [5]. Además, las evaluaciones automáticas nos muestran también que el usuario podría de esta manera sólo necesitar el 11'2% de las pulsaciones de teclas que serían necesarias sin utilizar un sistema CAT [6].

Como se puede suponer, este tipo de sistemas necesitará algún tipo de interfaz con el usuario para que éste puede interactuar con el sistema de una manera fácil y cómoda. En proyectos como TT2 [4], ya se creó una interfaz con este fin. En este trabajo describiremos otra de parecidas características. A la hora de implementar la nueva interfaz, fue importante tener presente que el sistema desarrollado debe funcionar bajo condiciones de usabilidad y tiempo de respuesta muy estrictas para que pueda ser de utilidad a los traductores humanos.

En este documento en una primera sección describiremos la traducción automática asistida por ordenador, para luego describir en otra sección la interfaz creada para interactuar con el usuario de un sistema CAT. Por último, algunas conclusiones y trabajos futuros.

2. TRADUCCIÓN ASISTIDA POR ORDENADOR

Las técnicas actuales de traducción automática están todavía lejos de producir traducción de alta calidad. Este inconveniente nos lleva a una aproximación alternativa al

Este trabajo ha sido parcialmente financiado por el proyecto Itefte (CICYT TIC2003-08681-C02-02) y por la Agencia Valenciana de Ciencia y Tecnología (AVCiT) (Grupos I+D+I GRUPOS03/031).

¹Del inglés *Computer Assisted Translation*

²Del inglés *Stochastic Finite State Transducer*

problema de la traducción que introduce el conocimiento humano en el campo de la traducción automática. En este marco, llamado Traducción Asistida por Ordenador (CAT), traductores humanos interactúan con un sistema de traducción que ofrece, de forma interactiva, la traducción que mejor completa la parte que ya ha sido traducida.

En [7] se propuso el paradigma CAT ejemplificado en el siguiente escenario: inicialmente el traductor humano parte de una frase en la lengua origen que debe traducirse a una frase del lenguaje destino. Dada esta frase en la lengua origen, el sistema de traducción es capaz de proporcionar una traducción en la lengua destino. En la mayoría de los casos esta traducción no es perfecta, por lo que requerirá la intervención del traductor humano para corregir los posibles errores. En el paradigma CAT se supone que el traductor empieza a leer y validar la frase ofrecida por el sistema de traducción de izquierda a derecha hasta cierta palabra, donde decide llevar a cabo una corrección. Por consiguiente, el traductor humano habrá aceptado un prefijo de la frase en la lengua destino y habrá introducido una corrección en forma de nuevo carácter. Este prefijo aceptado junto con el carácter introducido por el traductor humano dan lugar a un prefijo en la lengua destino, que será utilizado para refinar la búsqueda de la siguiente traducción del sistema. Como resultado de esta búsqueda, el sistema ofrecerá el sufijo de la frase de destino que mejor completa el prefijo validado por el traductor. Idealmente, la calidad del sufijo ofrecido por el sistema de traducción debería ir mejorando conforme el prefijo aceptado por el traductor humano es más largo.

Por tanto, el sistema estará compuesto por un módulo de traducción que ofrecerá traducciones de las frases en la lengua origen, y un segundo módulo que a partir del prefijo validado por el traductor humano y de las traducciones ofrecidas por el anterior módulo, define cuál es el mejor sufijo que completa el prefijo del traductor humano.

A continuación describimos formalmente el problema y cómo realizamos el proceso de traducción con esta clase de sistemas.

2.1. Traducción automática con SFSTs

En el marco estadístico de la traducción automática, dada una frase en la lengua origen \mathbf{s} , el objetivo es encontrar una frase en la lengua destino $\hat{\mathbf{t}}$ que cumpla lo siguiente:

$$\hat{\mathbf{t}} = \underset{\mathbf{t}}{\operatorname{argmax}} \operatorname{Pr}(\mathbf{t} | \mathbf{s}) = \underset{\mathbf{t}}{\operatorname{argmax}} \operatorname{Pr}(\mathbf{s} | \mathbf{t}) \cdot \operatorname{Pr}(\mathbf{t}) \quad (1)$$

La ecuación 1 también puede escribirse en forma de probabilidad conjunta que puede ser modelada por un SFST \mathcal{T} :

$$\begin{aligned} \hat{\mathbf{t}} &= \underset{\mathbf{t}}{\operatorname{argmax}} \operatorname{Pr}(\mathbf{s} | \mathbf{t}) \cdot \operatorname{Pr}(\mathbf{t}) \\ &= \underset{\mathbf{t}}{\operatorname{argmax}} \operatorname{Pr}(\mathbf{t}, \mathbf{s}) \\ &\approx \underset{\mathbf{t}}{\operatorname{argmax}} \operatorname{Pr}_{\mathcal{T}}(\mathbf{t}, \mathbf{s}) \end{aligned} \quad (2)$$

Los SFSTs se han utilizado con éxito en muchas tareas de traducción [8]. Además, existen algoritmos de búsqueda eficientes como Viterbi para la búsqueda del mejor prefijo y el algoritmo de enumeración recursivo (REA) para n -best.

Una posible manera de inferir SFSTs es con el método GIATI [9].

2.2. Búsqueda interactiva en un SFST

El concepto de búsqueda interactiva está estrechamente relacionado con el paradigma CAT. Este paradigma introduce un nuevo factor \mathbf{t}_p en la ecuación 1 de traducción automática. \mathbf{t}_p representa el prefijo de una frase traducida obtenida como resultado de la interacción entre un traductor humano y un sistema de traducción automática.

Un ejemplo de esta interacción se puede ver en la Figura 1. En cada iteración, un prefijo (\mathbf{t}_p) de una frase traducida ha sido aceptado por un traductor humano en una iteración anterior y el sistema CAT computa los mejores (o n -best) sufijos como hipótesis ($\hat{\mathbf{t}}_s$) para completar el prefijo.

Dado $\mathbf{t}_p \hat{\mathbf{t}}_s$, el ciclo CAT continúa cuando el usuario establece como aceptado un prefijo nuevo y más largo. Para ello, el traductor humano ha aceptado una parte (a) de $\mathbf{t}_p \hat{\mathbf{t}}_s$ (o más típicamente $\hat{\mathbf{t}}_s$). Después de este punto, el usuario puede teclear alguna tecla (k) con el fin de corregir alguna parte incorrecta. Por lo tanto, el nuevo prefijo está formado por \mathbf{t}_p seguido por la parte aceptada de una de las sugerencias propuestas por el sistema (a), más el texto (k) introducido por el usuario. Ahora este prefijo, $\mathbf{t}_p a k$, se convertirá en el nuevo \mathbf{t}_p , comenzando con ello un nuevo ciclo predictivo del sistema CAT.

Una forma simple de formalizar el proceso de sugerencias de hipótesis en el sistema CAT sería como sigue a continuación [6]. Dada una frase fuente \mathbf{s} y un prefijo validado por el usuario de una frase traducida \mathbf{t}_p , la búsqueda de un sufijo para completar la traducción y que maximice la probabilidad a posteriori sobre todos los sufijos es:

$$\hat{\mathbf{t}}_s = \underset{\mathbf{t}_s}{\operatorname{argmax}} \operatorname{Pr}(\mathbf{t}_s | \mathbf{s}, \mathbf{t}_p). \quad (3)$$

Teniendo en cuenta que $\operatorname{Pr}(\mathbf{t}_p | \mathbf{s})$ no depende de \mathbf{t}_s , podemos escribir:

$$\hat{\mathbf{t}}_s = \underset{\mathbf{t}_s}{\operatorname{argmax}} \operatorname{Pr}(\mathbf{t}_p \mathbf{t}_s | \mathbf{s}). \quad (4)$$

ITER-0	(t_p)	()
ITER-1	(t_s)	(Haga clic para cerrar el diálogo de impresión)
	(a)	(Haga clic)
	(k)	(en)
	(t_p)	(Haga clic en)
ITER-2	(t_s)	(ACEPTAR para cerrar el diálogo de impresión)
	(a)	(ACEPTAR para cerrar el)
	(k)	(cuadro)
	(t_p)	(Haga clic en ACEPTAR para cerrar el cuadro)
FINAL	(t_s)	(de diálogo de impresión)
	(a)	(de diálogo de impresión)
	(k)	(#)
	($t_p \equiv t$)	(Haga clic en <u>ACEPTAR</u> para cerrar el <u>cuadro</u> de diálogo de impresión)

Figura 1. Ejemplo de un sistema de interacción CAT para traducir al Español la frase en Inglés “Click OK to close the print dialog” extraída de un manual de impresoras. Cada paso comienza con un prefijo de salida previamente fijado t_p , para el cual el sistema sugiere un sufijo t_s . Entonces el usuario acepta parte de este sufijo (a) y tecldea algunas teclas (k), con el fin de solucionar la parte equivocada de t_s . Esto produce un nuevo prefijo, compuesto por el prefijo de la anterior iteración y la parte aceptada y tecldeada, (a,k), para ser utilizada como t_p en el siguiente paso. El proceso termina cuando el usuario introduce el caracter especial “#”. En la traducción final, t , todos los textos que han sido tecldeados por el usuario han sido subrayados.

donde $t_p t_s$ es la concatenación de un prefijo dado t_p y un sufijo t_s . Esta distribución conjunta puede ser modelada adecuadamente por SFSTs [10].

La solución a este problema de maximización ha sido realizado en dos fases. En la primera se extrae un grafo de palabras del SFST dada una frase fuente s . En la segunda fase, la búsqueda del mejor o mejores sufijos se lleva a cabo de acuerdo a la aproximación de Viterbi sobre el grafo de palabras de la primera fase dado un prefijo t_p de una traducción.

3. INTERFAZ CAT

En esta sección describiremos cómo hemos comunicado la interfaz con el sistema de traducción asistida por ordenador (motor de traducción), las funciones que fueron necesarias para esta comunicación, las partes que la componen y por último el funcionamiento que tiene y las facilidades que proporciona al usuario.

3.1. Comunicación entre la interfaz y el motor de traducción

La interfaz fue desarrollada en Java, mientras que el motor de traducción en C++. Por ello, es necesario comunicar estos dos programas de alguna manera. La forma más directa de integrar código nativo en programas escritos en Java es a partir de JNI (Java Native Interface)[11]. Hay otras como los sockets que nos permitirían tener motores de traducción en código nativo sin la necesidad que estuvieran implementados en C. En este trabajo nos centraremos exclusivamente a la integración de código nativo a través de JNI y dejaremos para posteriores versiones con más de un motor de traducción la comunicación por sockets.

Las funciones que son necesarias en este proceso de comunicación son las que se especifican a continuación:

- *init*: Realiza la carga del traductor en el motor de traducción.
- *setSource*: Se le indica la frase fuente al motor de traducción.
- *setTarget*: Se le indica la traducción aceptada por el usuario hasta el momento al motor de traducción.
- *suggest*: Soluciones alternativas que el motor de traducción proporciona como sugerencia de traducción dada una frase fuente y la traducción aceptada hasta el momento.

Estas funciones estarán tanto en Java como en C. En Java como funciones de una clase que cargará una librería dinámica que contendrá la parte de C. Y en C como un .h y .c específicos para que las funciones en C sean “entendibles” por el Java.

3.2. Componentes de la interfaz

Las partes que componen la interfaz y que se pueden observar en la Figura 2 son las siguientes:

- *Menús*: desde donde se pueden realizar tareas como cambiar de traductor, abrir/cerrar un fichero, guardar las traducciones a fichero, etc.
- *Frases Anteriores*: son las frases fuente junto con su traducción (en el caso que la tenga) que se encuentran antes (de la frase fuente que se está traduciendo en este momento) en el fichero que se ha cargado o las que se han guardado antes en el caso que no se realicen traducciones desde fichero.
- *Frases Posteriores*: son las frases fuente junto con su traducción (en el caso que la tenga) que se encuentran después (de la frase fuente que se está traduciendo en este momento) en el fichero que se ha cargado o las que se han guardado después en el caso que no se realicen traducciones desde fichero.

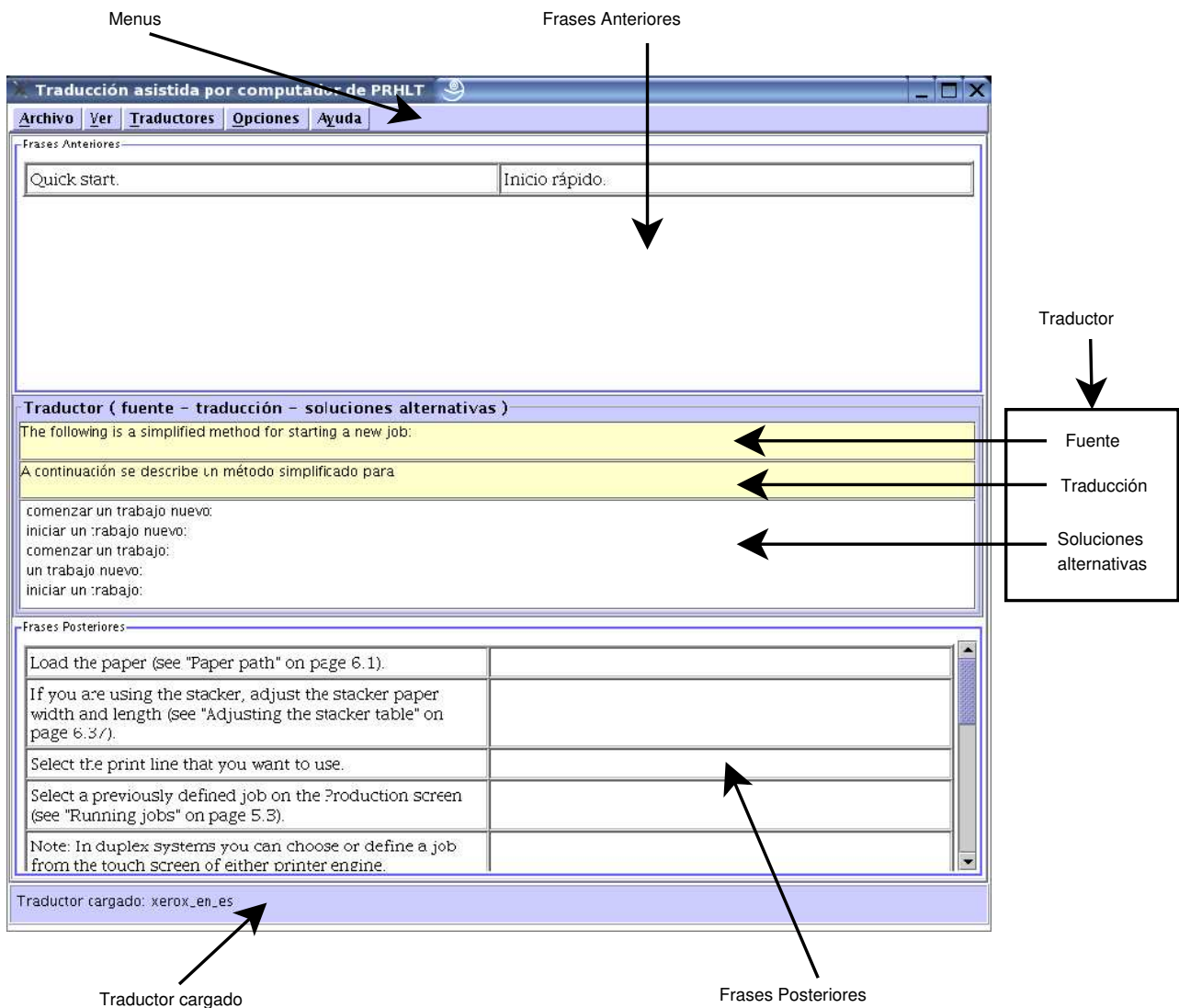


Figura 2. Componentes de la interfaz

- **Traductor:** es la parte en la interfaz en donde se visualiza siempre el proceso de traducción.
 - *Frase fuente:* es la frase a traducir.
 - *Frase traducida:* es tal y como está en un momento dado la traducción de la frase fuente (prefijo aceptado).
 - *Soluciones alternativas:* son las traducciones (sufijos) que se proporcionan al usuario como posibles traducciones de la frase fuente.
- **Traductor cargado:** es un campo que nos muestra en cada momento el traductor que utiliza el motor de traducción para proporcionar las soluciones alternativas.

3.3. Funcionamiento

El proceso de traducción que se lleva a cabo sigue los siguientes pasos:

1. Inicializar (cargar) el traductor³.
2. Pasar al motor de traducción la frase fuente.
3. Pasar al motor de traducción la frase traducida aceptada hasta el momento⁴.
4. Pedir al motor de traducción las soluciones alternativas para la traducción de una frase fuente dada teniendo en cuenta la frase traducida hasta el momento⁵.

³Este paso sólo se realiza al comienzo de la aplicación y cuando el usuario seleccione un cambio de traductor.

⁴Al principio se le pasa la cadena vacía, ya que no se tiene traducción.

⁵Devuelve 5 opciones.

5. Interacción con el usuario⁶.

- Elección como traducción una de las soluciones alternativas.
- Elección de parte de una traducción.
- Modificación de la traducción.

6. Volver a paso 3 después de cada una de las interacciones anteriores realizadas por el usuario.

En la Figura 3 se puede ver un ejemplo de este proceso.

Algunas de las funcionalidades se especifican a continuación.

3.3.1. Carga de varios traductores

La interfaz permite realizar traducciones de diferentes tareas y en diversas direcciones. Esto se lleva a cabo cargando el traductor correspondiente en cada caso. Por ejemplo, se tiene un traductor de manuales de impresoras del inglés al castellano, otro sobre partes meteorológicos del euskera al castellano y viceversa, etc. En la interfaz se mostrará en la parte inferior el traductor que está cargado para que el usuario sepa en cada momento sobre qué tarea puede realizar traducciones.

3.3.2. Abrir ficheros

Hay dos maneras de realizar traducciones. La primera es introduciendo directamente la frase fuente en el campo correspondiente (modo directo) y la otra es cargar un fichero de frases y seleccionar una para traducir (modo desde fichero). Este proceso se realiza de la misma manera que en otras aplicaciones. Junto con la interfaz hay unos ficheros de prueba para cada traductor, para probarlos sólo hay que seleccionar el que corresponda en cada caso.

3.3.3. Aumento/disminución de la letra

A los campos afectados durante el proceso de traducción se les puede aumentar o disminuir la letra al tamaño que le sea cómodo al usuario. Dicha función se encuentra en el menú Ver y permite aumentar, disminuir o volver al tamaño original de letra.

3.3.4. Modos de traducción: de fichero/directo

Como ya se ha comentado en apartados anteriores, se permite realizar traducciones o bien de un modo directo o bien de un modo desde fichero. El proceso de traducción es el mismo para ambos casos. Lo único que varía es la manera de seleccionar una frase a traducir. En el primer caso se introduce en el campo de frase fuente y cuando se quiere cambiar de fuente se selecciona la opción Nueva Fuente del menú Archivo. En el otro caso (de fichero) se selecciona una de las frases anteriores o posteriores.

⁶Puede realizar una o varias de estas opciones.

3.3.5. Guardar traducciones

En el modo directo, primero hay que decirle al sistema que esa traducción la queremos guardar (Archivo/Guardar traducción) y luego decirle que la guarde en fichero (Archivo/Guardar traducciones como...). En el modo desde fichero, lo único que hay que hacer es que la guarde a fichero.

3.3.6. Cambio de foco

Estas opciones del menú se crearon para facilitar el proceso de traducción al usuario. Estas opciones tienen asociadas unas teclas rápidas para agilizar el cambio de foco.

3.3.7. Ayuda teclas rápidas

Se han asociado una serie de teclas rápidas a ciertas utilidades de la interfaz para facilitar su utilización. Éstas se pueden consultar en la ayuda en la interfaz.

4. CONCLUSIONES Y TRABAJO FUTURO

Se ha creado una interfaz para la ayuda a la traducción basada en la ya existente en el proyecto TT2 [4]. Esta nueva interfaz, a diferencia de la anterior permite cambiar el traductor en tiempo de ejecución, permitiendo de esta manera realizar traducciones sobre diferentes tareas sin necesidad de salir del programa.

Es una interfaz fácil de utilizar y en la que el proceso de traducción está siempre en el mismo punto de la interfaz. Ésto hace que el usuario siempre sepa dónde le van a aparecer las traducciones y las opciones que se le proponen.

Otra funcionalidad extra es la de poder aumentar y disminuir la letra de los campos utilizados durante la traducción dependiendo del gusto del usuario o para permitir una mejor visualización en demos en donde se tienen que visualizar desde bastante distancia.

Además también se permite realizar traducciones y guardarlas sin éstas tener que estar en un fichero, es decir, el usuario puede introducir una frase de la tarea del traductor cargado en cada momento y poder obtener una traducción de ella y su posterior almacenamiento en disco. Esta interfaz también permite dejar frases sin terminar de traducir y continuar con él en otro momento.

En trabajos futuros se pretende introducir nuevas funcionalidades como sería el caso de permitir cambiar el motor de traducción o permitir cargar diferentes textos planos y ella misma que se encargue de segmentarlos en frases para poder realizar el proceso de traducción de una manera más rápida. En estos momentos se permite cambiar de traductor, pero el motor de traducción es siempre el desarrollado en el marco del proyecto TT2 y las frases desde fichero se le tienen que pasar ya segmentadas. Además, se pretende crear una interfaz

ITER-0	Frase traducida	()
ITER-1	Soluciones alternativas	(A continuación se describe un método muy simple para comenzar un trabajo nuevo.) (A continuación se describe un método muy simple para comenzar un trabajo.) (El continuación se describe un método muy simple para comenzar un trabajo nuevo.) (La continuación se describe un método muy simple para comenzar un trabajo nuevo.) (Continuación se describe un método muy simple para comenzar un trabajo nuevo.)
	Prefijo seleccionado	(A continuación se describe un método)
ITER-2	Soluciones alternativas	(muy simple para comenzar un trabajo nuevo.) (muy simple para comenzar un trabajo.) (muy simple para comenzar un.) (muy simple para comenzar un trabajo nuevo.) (muy simple para comenzar un trabajo nuevo)
	Modificación	(simplificado para)
FINAL	Soluciones alternativas	(comenzar un trabajo nuevo.) (iniciar un trabajo nuevo.) (comenzar un trabajo.) (un trabajo nuevo.) (iniciar un trabajo.)
	Frase traducida	(A continuación se describe un método <u>simplificado para</u> comenzar un trabajo nuevo.)

Figura 3. Ejemplo de interacción de la interfaz con el usuario para traducir del castellano la frase en inglés “The following is a simplified method for starting a new job:” extraída de un manual de impresoras. En cada paso se especifican las soluciones alternativas proporcionadas por el sistema y las operaciones de selección y modificación que realiza el usuario. El proceso termina cuando el usuario cambia de frase. En la traducción final, última *Frase traducida*, todos los textos que han sido tecleados por el usuario han sido subrayados.

que sea accesible desde la web para mayor difusión del sistema.

5. BIBLIOGRAFÍA

- [1] J. Civera, J. M. Vilar, E. Cubel, A. L. Lagarda, S. Barrachina, E. Vidal, F. Casacuberta, D. Picó, y J. González, “From machine translation to computer assisted translation using finite-state models,” in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP04)*, Barcelona, 2004.
- [2] E. Vidal y F. Casacuberta, “Learning finite-state models for machine translation,” in *Grammatical Inference: Algorithms and Applications. Proceedings of the 7th International Colloquium ICGI 2004*, vol. 3264 of *Lecture Notes in Artificial Intelligence*, pp. 16–27. Springer, Athens, Greece, October 2004, Invited conference.
- [3] Enrique Vidal, Francisco Casacuberta, y Pedro García, “Grammatical inference and automatic speech recognition,” in *New Advantages and Trends in Speech Recognition and Coding*, A. Rubio, Ed., vol. 147 of *NATO-ASI Series F: Computer and Systems Sciences*, pp. 174–191. Springer-Verlag, 1995.
- [4] Atos Origin, Instituto Tecnológico de Informática, RWTH Aachen, RALI Laboratory, Celer Soluciones and Societé Gamma and Xerox Research Centre Europe, “TransType2 - Computer Assisted Translation. Project Technical Annex.” 2001.
- [5] E. Macklovitch, “TransType2: The Last Word,” in *Proc. of LREC’06*, 2006.
- [6] J. Civera, A.L. Lagarda, E. Cubel, F. Casacuberta, E. Vidal, J.M. Vilar, y S. Barrachina, “A Computer-Assisted Translation Tool based on Finite-State Technology,” in *Proc. of EAMT’06*, 2006, pp. 33–40.
- [7] Philippe Langlais, George Foster, y Guy Lapalme, “Unit completion for a computer-aided translation typing system,” *Machine Translation*, vol. 15, no. 4, pp. 267–294, 2000.
- [8] F. Casacuberta, H.Ñey, F. J. Och, E. Vidal, J. M. Vilar, S. Barrachina, I. Garcia-Varea, D. Llorens, C. Martinez, S. Molau, F.Ñevado, M. Pastor, D. Pico, y A. Sanchis, “Some approaches to statistical and finite-state speech-to-speech translation,” *Computer Speech and Language*, vol. 18, pp. 25–47, 2004.
- [9] F. Casacuberta y E. Vidal, “Machine translation with inferred stochastic finite-state transducers,” *Computational Linguistics*, vol. 30, no. 2, pp. 205–225, 2004.
- [10] J. Civera et al., “A syntactic pattern recognition approach to computer assisted translation,” in *Advances in Statistical, Structural and Syntactical Pattern Recognition*, A. Fred, T. Caelli, A. Campilho, R. P.W. Duin, y D. de Ridder, Eds., *Lecture Notes in Computer Science*, pp. 207–215. Springer-Verlag, 2004.
- [11] Sheng Liang, *The Java Native Interface: Programmer’s Guide and Specification*, Addison Wesley Longman, 1999, ISBN: 0201325772.