

UNIVERSIDAD DE SALAMANCA

FACULTAD DE TRADUCCIÓN Y DOCUMENTACIÓN

GRADO EN TRADUCCIÓN E INTERPRETACIÓN

Trabajo de Fin de Grado

**Evaluación de usabilidad
de la herramienta TAO de
software libre OmegaT**

Lara González Rodríguez

Emilio Rodríguez Vázquez de Aldana

Salamanca, 2023

Tabla de contenido

Introducción.....	6
Motivación.....	6
Objetivos.....	7
Estructura.....	8
Marco teórico.....	10
Características básicas de las herramientas TAO.....	10
Repaso a la evolución de los sistemas de memorias de traducción.....	12
Investigación sobre alternativas a la distancia de edición.....	14
Usabilidad de las herramientas TAO.....	16
Metodología.....	19
Evaluación de la usabilidad de OmegaT.....	21
Compatibilidad e interoperatividad.....	21
Formato.....	23
Flujo de trabajo.....	24
Control de calidad.....	29
Funciones de búsqueda.....	30
Segmentación.....	32
Automatización.....	33
Memorias de traducción y terminología.....	33
Personalización.....	35
Instrucción.....	36
Conclusiones y algunas propuestas de mejora.....	38

Bibliografia.....	40
-------------------	----

Resumen

En la actualidad, las herramientas de traducción asistida por ordenador (TAO) son de vital importancia para nuestra profesión, por lo que consideramos que la disponibilidad de alternativas de *software* libre de calidad es muy beneficioso para los estudiantes y traductores profesionales. Una de las principales maneras de determinar la calidad de un programa informático es midiendo su usabilidad, por lo que hemos decidido determinarla en la herramienta de *software* libre OmegaT. Muchos académicos se han preocupado por la accesibilidad de las herramientas de TAO en general, pero ha habido muy pocos estudios enfocados en herramientas concretas, y ninguno en programas de *software* libre. Hemos evaluado varios aspectos de OmegaT, de los cuales hemos determinado que la gestión de archivos, los recursos de aprendizaje, el formato de texto y la separación y unión manual de segmentos son los más problemáticos. Sus mayores ventajas son su precio, personalización, compatibilidad, búsqueda externa y rendimiento. Aunque este trabajo es meramente preliminar, consideramos que es una herramienta que, aunque tiene mucho por mejorar en términos de usabilidad, se utiliza para trabajo profesional y en contextos educativos en su estado actual. Sin embargo, un estudio más exhaustivo con usuarios es necesario para determinar con más exactitud las ventajas y carencias de esta herramienta.

Palabras clave: herramientas TAO, usabilidad, *software* libre, OmegaT

Abstract

Currently, computer-assisted translation (CAT) tools are vital for our profession. For this reason, we consider the availability of good-quality free (as in freedom) software alternatives to be very beneficial to both students and professional translators. One of the main ways to determine the quality of a computer program is by measuring its usability. Thus, we have decided to study the usability of the free software tool, OmegaT. Many academics have studied the usability of CAT tools, but few have focused on specific programs, and none of them have looked at free software tools. We evaluated multiple aspects of OmegaT, of which file management, learning resources, text formatting and manual splitting and joining of seg-

ments were the most problematic, while price, customization, compatibility, external search and performance were its best features. Even though this work is preliminary, we consider that this tool is ready for use in professional and formative settings, although there is still a lot of room for improvement in terms of usability. However, a more exhaustive study with users is necessary to determine more exactly the advantages and disadvantages of this tool.

Keywords: CAT tools, usability, Free and Open Source Software (FOSS), OmegaT

1 Introducción

1.1 Motivación

En la actualidad, la práctica de la traducción y las herramientas de traducción asistida por ordenador (TAO, en adelante) van casi necesariamente de la mano. Un gran número de traductores utilizan herramientas de TAO y, aunque hay algunos ámbitos en los que su adopción no es mayoritaria, lo cierto es que a día de hoy es prácticamente un requisito para ejercer como traductore. Al ser estos programas una parte tan esencial de nuestra profesión, consideramos que debemos tener la opción de utilizar alternativas de *software* libre. Esto significa, entre otras cosas, poder usar el programa para cualquier propósito que necesitemos, poder estudiar y modificar el código fuente y poder redistribuirlo (Stallman, 2004). Todo ello puede ser útil, por ejemplo, si necesitamos añadir una característica al programa o solucionar un error. No todo el mundo sabe programar, pero, por ejemplo, se podría contratar a un desarrollador por nuestra cuenta para que lo solucione en vez de esperar a que lo haga el propietario del *software*. Esto puede resultar particularmente útil para empresas o instituciones que necesiten características específicas para su trabajo, pues no necesitarían desarrollar una herramienta desde cero, con lo que ahorrarían tiempo y dinero.

Otra ventaja de que el código fuente esté disponible y a la vista de todos son la seguridad y la privacidad. Podemos saber si el programa envía nuestros datos o los documentos con los que trabajamos a un servidor externo o no, porque es posible leer el código fuente, mientras que no se puede saber exactamente cómo funciona un programa privativo (Ídem, p. 255). Esto es especialmente relevante en proyectos con contrato de confidencialidad y al usar memorias de traducción personales muy extensas y de gran valor, que podrían verse comprometidas por una brecha de seguridad.

Este último punto es especialmente importante para los traductores autónomos, que en España suponen el 75,8% (*Libro Blanco de los derechos de autor de las traducciones de libros en el ámbito digital*, 2016, p. 64). Para ellos, el coste de las TAO más populares puede ser restrictivo, especialmente si son jóvenes y están empezando. Si bien no todo el *software*

libre o de código abierto es necesariamente gratuito (por ejemplo, la TAO Swordfish tiene una licencia de código abierto y es de pago), el programa que analizamos en este trabajo sí que lo es. Por ello, consideramos que, además de las libertades que ofrece, puede ser una buena opción para un traductor autónomo novato que aún no tenga demasiados ingresos.

Si bien todo lo anterior es importante, también consideramos que la herramienta que usemos debe ser práctica y funcional. En otros ámbitos hay herramientas de *software* libre que son muy utilizadas, como es el caso de Blender en el ámbito del modelado 3D, pero en la traducción aún dominan con mucha diferencia los programas con licencias privativas. Esto puede ser, en parte, porque las alternativas libres como OmegaT aún no están a la altura de otras como Trados o MemoQ.

Hemos elegido OmegaT para este trabajo porque es una de las herramientas TAO de *software* libre más populares, es multiplataforma y gratuita, y porque tenemos experiencia previa con ella. Además, es sencilla de descargar en todos los sistemas y no requiere ser compilada por los usuarios en su versión gratuita, y su desarrollo y comunidad están activos.

1.2 Objetivos

Algunas de las quejas más frecuentes de los usuarios de TAO son la calidad de la interfaz de usuario, el manejo de formatos, la fluidez de la interfaz (Zaretskaya et al., 2018), la ayuda del programa y la facilidad de aprendizaje (Vargas Sierra, 2019). Hemos decidido centrarnos en las que pertenecen a la usabilidad, porque, además de ser las más numerosas, consideramos que son de las más importantes a la hora de que un usuario decida escoger una herramienta u otra.

El objetivo principal de este trabajo, por tanto, será averiguar si el programa OmegaT está a la altura de las demandas de los traductores profesionales en usabilidad y, por lo tanto, si supone una alternativa viable a los sistemas privativos comerciales que hoy en día son los líderes de mercado.

Si bien tocaremos aspectos como el funcionamiento, las memorias de traducción y los formatos de archivos, lo haremos desde la perspectiva del usuario final y de la usabilidad, ya

que consideramos que estos aspectos también son decisivos a la hora de determinar si un programa de TAO es usable o no.

Uno de los focos principales de este trabajo será la facilidad de aprendizaje, por varios motivos. En primer lugar, como ya hemos mencionado, es una de las quejas más frecuentes sobre herramientas TAO (Vargas Sierra, 2019) (Alotaibi, 2020). Además, como la mayoría de traductores son autónomos, suelen aprender herramientas nuevas de manera autodidacta, por lo que la facilidad de aprendizaje es clave para la adopción de programas nuevos. Por último, consideramos que otro grupo que se beneficiaría en gran medida con el uso de programas de *software* libre son los estudiantes que, en la mayoría de los casos, tienen experiencia limitada o nula con herramientas TAO, por lo que su facilidad de aprendizaje es, de nuevo, de vital importancia.

Debido los recursos limitados que tiene un proyecto de software libre como este, creemos que es conveniente poner luz sobre sus principales problemas de usabilidad, de modo que los desarrolladores sepan en qué aspectos enfocarse. Si bien este es un trabajo preliminar, creemos que puede abrir paso a otros estudios más completos sobre esta herramienta que permitan determinar estos problemas con más precisión y, por lo tanto, mejorar la experiencia de usuario y la adopción de OmegaT o de otras herramientas de *software* libre.

1.3 Estructura

Este trabajo está dividido en dos partes: la primera, en la que ponemos en contexto las herramientas TAO y la investigación actual sobre ellas, enfocándonos en los estudios sobre usabilidad, y la segunda, en la que analizamos la usabilidad del programa OmegaT.

En la primera parte, describiremos brevemente las funciones y características básicas de los programas de TAO. Seguidamente, repasaremos tanto la evolución de estas herramientas desde su concepción en los años 60 hasta la actualidad como la investigación sobre sistemas de recuperación de memorias de traducción que utilizan el conocimiento lingüístico y los estudios que se han llevado a cabo a lo largo de los años sobre la usabilidad de las herramientas TAO.

En la segunda parte, se comenzará describiendo la metodología utilizada para analizar la usabilidad de OmegaT y los estudios en los que nos basamos para ello. Posteriormente, analizaremos los aspectos del programa que consideramos más relevantes y ofreceremos sugerencias para mejorarlos si es necesario. Concluiremos con un repaso de los principales puntos fuertes y débiles y mencionaremos las áreas en las que es necesario investigar más.

2 Marco teórico

2.1 Características básicas de las herramientas TAO

Un sistema de memoria de traducción, que aquí tomamos como sinónimo de sistema de traducción asistida por ordenador (TAO), es un programa informático que tiene como elemento principal una memoria de traducción, que es una colección de traducciones guardadas junto a sus textos originales y divididas en unidades fácilmente manejables, normalmente oraciones. A la hora de traducir, estas herramientas buscan coincidencias completas o parciales en las memorias de traducción, con lo que le traductore puede reutilizar traducciones anteriores y ahorrar tiempo (Reinke, 2018).

Además de la memoria, los programas de TAO tienen los siguientes elementos básicos para ayudar a le traductore a llevar a cabo su trabajo:

- Un **entorno de trabajo** o interfaz de usuario, que puede ser personalizable en mayor o menor medida. Se debe distinguir entre programas con interfaz de trabajo propia, entre los que se incluyen OmegaT, Trados Studio y MemoQ, por ejemplo; y programas que se integran en otro (a menudo un procesador de textos), como Anaphraseus y WordFast Classic. El primer tipo es el más común. Dentro de él, también se debe distinguir entre los que colocan el texto traducido bajo el segmento original, como OmegaT, y los que colocan los segmentos originales y traducidos en dos columnas, como Swordfish o Trados. Aunque del editor de traducciones es el elemento principal de la interfaz, los entornos de trabajo también tienen otras ventanas que permiten visualizar las memorias de traducción y las bases de datos terminológicas (Oliver, 2016). También suelen incluir herramientas de gestión de proyectos, que en los programas más actuales tienen funciones de trabajo colaborativo, así como herramientas de alineación y de control de calidad (Alotaibi, 2020).
- También es necesario que el programa sea capaz de consultar **glosarios y diccionarios terminológicos**, que podrán tener más de una entrada por término. El principal

reto que plantea esta característica es el de poder detectar un término con variaciones morfológicas, por ejemplo de número, género o caso (Oliver, 2016).

- La mayoría de sistemas actuales ya integran la conexión con motores de **traducción automática**. El traductor puede optar por que el programa complete con traducción automática los segmentos que no tienen ninguna coincidencia en la memoria (Oliver, 2016 pg. 204-205), o integrándolos de formas diversas. En algunas investigaciones, como (Simard, 2009), se plantean sistemas que combinen la traducción asistida con la automática de manera inteligente. Esto supone que las traducciones automáticas sean consistentes con las almacenadas en las memorias y que haya un sistema que descarte las que no son útiles.
- El tratamiento de formatos es otra característica muy importante de los sistemas de traducción asistida por ordenador. Es necesario que sean capaces de trabajar con la mayor variedad posible de formatos de archivo en los textos de origen y poder reconvertir las traducciones al mismo formato conservando el estilo adecuado (Oliver, 2016). Para ello, los editores de las herramientas TAO tienen métodos diferentes para añadir las etiquetas de los textos originales a las traducciones y añadir marcas de formato nuevas. La capacidad de trabajar con formatos de texto distintos es una clara ventaja para el traductor, que no tendrá que preocuparse de convertir manualmente los ficheros de los encargos que le envíen los clientes, ni en pasarse demasiado tiempo maquetando después de traducir.

Hay dos tipos de memoria de traducción: las basadas en bases de datos y las basadas en textos de referencia. Ambas tienen sus ventajas y sus desventajas.

La mayoría de las memorias de traducción son bases de datos formadas por unidades de traducción en los idiomas de origen y de llegada. Cada unidad de traducción también puede incluir datos como las fechas de modificación, autor, proyecto, cliente, etc. Estas bases de datos son dinámicas, es decir, se les pueden añadir nuevas unidades de traducción durante el proceso de traducción, que pueden reutilizarse inmediatamente. Hay tres maneras de alimentar a una memoria de traducción de este tipo:

- 1 Mientras se traduce, las nuevas unidades de traducción se añaden automáticamente.
- 2 Es posible importar una memoria de traducción ya creada, ya sea en el formato propio de la herramienta TAO o en un formato estándar como TMX.
- 3 Se puede crear una nueva memoria de traducción a partir de textos ya traducidos mediante el uso de una herramienta de alineación.

Un número menor de programas utilizan un modelo de textos de referencia, que tiene la ventaja de facilitar ver la unidad de traducción en su contexto. Sin embargo, es un sistema muy estático que no permite reutilizar unidades de traducción recién añadidas. Para solucionar esto, se creó el *fuzzy index*, que suplementa a los textos de referencia y permite acceder a unidades de traducción inmediatamente después de haberlas creado, de modo que se forma un sistema híbrido.

También se han utilizado estrategias híbridas similares en las memorias almacenadas en bases de datos. Para solucionar el problema del contexto, se crearon las *context matches*, que se aplican si una coincidencia exacta está precedida por otra coincidencia exacta. También hay algunos sistemas de este tipo que incluyen una opción suplementaria de textos de referencia.

2.2 Repaso a la evolución de los sistemas de memorias de traducción

Reinke (2018) realizó un repaso exhaustivo al estado del arte de los programas de TAO, en el que nos basamos aquí para poner en contexto la herramienta que vamos a analizar.

La idea de crear un sistema de traducción asistida por ordenador apareció por primera vez en los años 60, cuando la Comunidad Europea del Carbón y del Acero creó un sistema informático que recuperaba términos extraídos de traducciones humanas previas y los mostraba junto a su contexto (ALPAC, 1966). Si bien esta implementación era muy primitiva, supuso un primer paso muy importante hacia la traducción asistida que conocemos hoy en día.

A principios de los años 70, el servicio de traducción de la *Bundeswehr* alemana desarrolló un sistema que trataba de aprovechar más las traducciones almacenadas. Este sistema buscaba fragmentos del texto que no hubieran cambiado en la memoria para poder reutilizarlos. El objetivo era facilitar las revisiones. Uno de los usos propuestos para este sistema es el de las traducciones de nuevas ediciones de documentos, en las sólo haría falta reescribir las partes que habían cambiado en la versión original (Krollmann, 1971).

El siguiente gran avance que llegó al mundo de la traducción asistida fue el de las coincidencias parciales o *fuzzy matches*. Esta idea fue propuesta por el traductor de la Comisión Europea Peter Arthern a finales de la década de los 70 (Arthern, 1979). Propuso un sistema que devolviera el equivalente más cercano a cada fragmento. Se pretendía disponer de al menos una traducción parcial y correcta gramaticalmente, en la que luego se traducirían de manera manual los fragmentos sin coincidencias en la memoria con ayuda de una base terminológica, aunque también propuso integrar la traducción automática para esta tarea.

No fue hasta los años 80 que estas ideas llegaron a sistemas comerciales. ALPS Inc. fue pionera en implementar la recuperación automática de coincidencias exactas en su *Translation Support System* (TSS) (Seal, 1992). Una década más tarde, sistemas como *IBM Translation Manager* y *Trados Translator's Workbench II* comenzaron a incluir coincidencias parciales (Hutchins, 1998) (Reinke, 2004 p. 36-41).

En las últimas dos décadas, las funciones básicas de las herramientas TAO no han cambiado demasiado (Reinke, 2018, p. 60). La innovación en las herramientas TAO se ha concentrado en el aprovechamiento de subsegmentos y en la integración con la traducción automática.

Hay tres maneras principales de recuperar subsegmentos (Ídem, p. 64). La primera y la más simple es buscar unidades de traducción completas en la lengua de origen e insertarlas en la traducción. De este modo se crea una mezcla de lenguas de origen y de destino que necesita ser adaptada. La segunda manera es recuperar la cadena de caracteres común más larga (LSC, por sus siglas en inglés). La manera más productiva es sugerir automáticamente fragmentos en el lenguaje de llegada mientras se escribe la traducción.

En la actualidad, casi todos los programas de TAO integran uno o varios motores de traducción automática. La mayoría ya usan traducción automática neuronal. Hay dos maneras principales en las que se integran la traducción asistida con la traducción automática: (Ídem, p. 66):

1. Procesamiento por lotes: Todos los fragmentos que no tienen un nivel de coincidencia lo suficientemente alto en la memoria de traducción se traducen automáticamente. Por lo general, el programa marca los fragmentos que se han traducido de manera automática.
2. Procesamiento interactivo: Le traductore ejecuta el sistema de traducción automática cuando lo considera necesario.

En ambos casos, tras el proceso de posesición, o tras la confirmación de le traductore de que el segmento traducido es adecuado, se guarda la traducción en la memoria.

2.3 Investigación sobre alternativas a la distancia de edición

La métrica de similitud básica en la que se basan todos los sistemas de TAO del mercado es la distancia de edición. Este sistema es muy fácil de implementar y es, en la mayoría de los casos, independiente de la lengua (Ranasinghe et al. 2020). Sin embargo, se basa simplemente en la similitud de caracteres y no es capaz de utilizar criterios lingüísticos para calcular coincidencias, por lo que las memorias de traducción no se usan de manera tan eficiente como se podría conseguir con un sistema que tenga en cuenta variaciones morfológicas, sintácticas y semánticas.

Desde hace algunas décadas, los investigadores se han interesado por encontrar un sistema que pueda reemplazar a la distancia de edición y mejorar el funcionamiento de estos sistemas. (Ídem) realiza un repaso a los distintos estudios que se han realizado a lo largo de los años sobre esta materia.

A finales de la década de los 90, Planas y Furuse (1999) añadieron al algoritmo basado en la distancia de edición el cálculo de la similitud entre segmentos lemas e información gramatical. Sin embargo, lo probaron en menos de 150 segmentos.

Hodász y Pohl (2005) también utilizaron lemas e información gramatical. Además, experimentaron con diferentes maneras de alinear automáticamente segmentos suboracionales en origen y destino, aunque los resultados son preliminares.

Pekar y Mitkov (2007) demostraron que la calidad de las coincidencias puede mejorar si se aplican algoritmos que tengan en cuenta la estructura sintáctica de las oraciones. Sin embargo, utilizaron sólo unos pocos segmentos.

En 2011, Utiyama et al. sugirieron un algoritmo que consideraba paráfrasis a hora de buscar coincidencia. No obstante, se enfocó en segmentos con el mismo significado y solo se probó con un traductor (Utiyama et al., 2011).

En 2014, se propuso un algoritmo que buscaba coincidencias mediante árboles sintácticos. Los resultados eran preliminares y el sistema era tan lento que resultaba inviable (Vanlillemeersch y Vandeghinste, 2014). Ese mismo año, Gupta y Orasan (2014) propusieron una variante de la distancia de edición que hacía uso de las paráfrasis y de la programación dinámica. Las pruebas que se realizaron, tanto automáticas como humanas, demuestran que este sistema tiene ventajas sobre la distancia de edición simple (Gupta et al., 2016).

El sistema planteado por Chatzitheodorou (2015) es similar al mencionado anteriormente. Crea paráfrasis para construcciones verbales en todas las unidades de traducción de origen, de modo que es capaz de encontrar más coincidencias parciales y de mayor calidad. Se demostró que usar este sistema facilita y acelera el proceso de traducción.

Por último, Ranasinghe et al. (2020) proponen un sistema de recuperación de coincidencias parciales basado en *Universal Sentence Encoder* (USE), y comparan sus dos versiones: *Transformer* y *Deep Averaging Network* (DAN). En las pruebas realizadas en memorias de traducción, USE tuvo resultados mucho mejores que la distancia de edición en coincidencias parciales con puntuaciones de entre 0.6 y 0.8, por lo que puede mejorar la recuperación de unidades de traducción. Se pidió a hablantes nativos que compararan los resultados, y, en general, USE obtuvo mejores resultados que la distancia de edición. Sin embargo, no hubo tiempo para comparar todos los fragmentos y aún se debe analizar este método más a fondo.

Los métodos de búsqueda de coincidencias basados en el conocimiento lingüístico continúan siendo objeto de estudio y, por el momento, ninguno de ellos ha conseguido llegar a ser implementado en un sistema de traducción comercial.

2.4 Usabilidad de las herramientas TAO

ISO define la usabilidad como «*the capability of a software product to be understood, learnt used and that is attractive to the user when used in specified conditions*¹» (ISO/IEC, 2016) y la divide en tres atributos:

- 1 Efectividad: la precisión y completud con la que los usuarios consiguen objetivos concretos.
- 2 Eficiencia: relación entre la efectividad con la que se consiguen los objetivos y la cantidad de recursos utilizados para ello.
- 3 Satisfacción: la comodidad de uso del programa y las actitudes positivas que expresan los usuarios al utilizarlo

La usabilidad es un atributo esencial para determinar la calidad de un programa informático. En palabras de (Bevan, 1995), «*usability is quality in use*²». Es por ello que es esencial considerarla a la hora de desarrollar un producto de *software*. Dillon (2001 p. 2), enumera tres maneras principales de evaluar un programa informático:

- 1 Con usuarios: se pide a los usuarios que lo prueben y se les encuesta.
- 2 Con expertos: un experto evalúa la usabilidad del producto.
- 3 Con modelos: un experto de infraestructura hiperconvergente utiliza métodos para predecir el desempeño de los usuarios.

El mismo autor aclara que realizar métodos que pongan el foco en los usuarios siempre da los resultados más precisos (Ídem, p. 4). Uno de ellos es SUMI, un método de evaluación

1 «la capacidad de un producto de *software* de ser entendido, aprendido, utilizado y atractivo para el usuario cuando se usa en condiciones especificadas»

2 «la usabilidad es la calidad en uso».

de usabilidad estandarizado internacionalmente que fue presentado en (Kirakowski, 1996). Muchos académicos lo consideran una manera fiable de determinar la satisfacción de los usuarios con una herramienta de *software*, así como la percepción que tienen de la usabilidad de ella. Tiene cuatro escalas: afecto (*affect*), ayuda (*helpfulness*), control (*control*) y aprendizaje (*learnability*).

En los últimos años, varios académicos se han empezado a preocupar por la usabilidad de las herramientas de traducción asistida, pero ninguno se ha enfocado en programas de *software* libre en particular. Vargas Sierra (2019) realiza un repaso a los distintos estudios sobre usabilidad de las herramientas TAO que se han realizado, en el que nos basamos aquí. (Lagoudaki, 2008) identificó la necesidad de incluir varias características adicionales a los programas de TAO de la época, como la gestión de proyectos, el control de calidad y un sistema que facilitase la colaboración. Concluyó que la usabilidad y las necesidades de los usuarios a menudo se pasan por alto a la hora de diseñar estas herramientas. Asare (2011) empleó métodos etnográficos para examinar sus opiniones sobre la usabilidad de las herramientas. Los participantes mostraron actitudes positivas hacia las tecnologías de la traducción, pero también expresaron descontento acerca de las interfaces de usuario, las características complicadas y la falta de flexibilidad. En (LeBlanc, 2013), los encuestados expresaron quejas similares, centradas sobre todo en el diseño y la concepción de las herramientas. Por otro lado, también afirmaron que los sistemas de memorias de traducción mejoraban la consistencia y reducían las labores repetitivas. Bergh et al. (2015) realizaron un estudio empírico sobre la usabilidad de los sistemas de TAO para el proyecto de investigación Smart Computer-Aided Translation Environment (SCATE). Utilizaron varios métodos de recolección de datos: una encuesta, entrevistas semi-estructuradas y nueve consultas contextuales. El estudio aporta varias recomendaciones que podrían mejorar el proceso de trabajo de los traductores. La encuesta a gran escala *ErgoTrans* (O'Brien et al., 2017) se enfocó en identificar los elementos negativos de las herramientas de traducción. Los resultados muestran que más de la mitad de ellos tienen que ver con la interfaz de usuario y con las características de los programas. Vargas Sierra (2019) realizó una encuesta SUMI a 95 estudiantes de traducción de dos años distintos sobre el programa Trados Studio. La escala global se encontraba en la media, pero

«aprendizaje» estaba por debajo. Solo una de las escalas puntuó por encima de la media. Se concluyó que es necesario hacer más hincapié en el diseño de esta herramienta, y se aportaron varias soluciones a los problemas que se encontraron. Por último, (Alotaibi, 2020) realizó otra encuesta SUMI a 42 traductores que trabajan con el árabe. En este caso, la calificación global se situó por encima de la media; las escalas que puntuaron más alto fueron las de «afecto» y «eficiencia», mientras que «ayuda» y «aprendizaje» fueron de nuevo las peor calificadas. La mayoría de participantes indicó que le recomendaría esta herramienta a otras personas, y que la herramienta era eficiente, rápida y cómoda de navegar. Por otro lado, se quejaron de que había demasiadas características y que la interfaz debía ser más fácil de usar.

3 Metodología

El objetivo de este trabajo es evaluar la usabilidad de OmegaT. Para ello, vamos a partir de los resultados de la encuesta realizada por O'Brien et al. (2017) sobre características «irritantes» de las herramientas TAO. En ella, se identificaron varios elementos como problemáticos, que corresponden a las siguientes categorías: Interfaz de usuario (28%), características de la herramienta TAO (25%), rendimiento técnico (17%), apariencia del texto (12%), compatibilidad (10%) y aspectos organizativos (8%) (Ídem, p. 155).

En la misma encuesta, también se descubrió que había varias características que faltaban en la mayoría de los sistemas de memorias de traducción. Son las siguientes (Ídem, p. 158-160):

- Compatibilidad/interoperatividad (16%): la capacidad de trabajar con todo tipo de archivos, como presentaciones de diapositiva, PDF y Excel, así como con otros programas como los reconocimiento de voz y traducción automática.
- Formato (13%): formato de texto, posibilidad de resaltar y cambio automático de elementos como comillas, que cambian entre lenguas.
- Flujo de trabajo (12%): se aportaron ideas como que la interfaz se pareciera más a la de un procesador de textos tradicional, que se pudiera trabajar con tres lenguas a la vez, mostrar un porcentaje del texto traducido, calculadora automática de precios, etc.
- Control de calidad (12%): correctores ortográficos, gramaticales y de puntuación.
- Funciones de búsqueda (12%): acceso directo a diccionarios en línea, búsqueda universal en memorias de traducción y textos y buscar y sustituir en varios textos a la vez.
- Segmentación (10%): separar y unir segmentos y herramienta de alineado de textos.
- Automatización (9%): autoguardado, autocorrección, sugerencias y actualizaciones automáticas.

- Memorias de traducción/terminología (8%): memorias de traducción que utilicen aprendizaje profundo.
- Vista previa (4%): capacidad de ver la disposición del texto mientras se traduce.
- Personalización (4%): capacidad de cambiar las barras de herramientas, atajos y botones.

También queremos examinar la documentación y la ayuda integrada en el programa, así como los recursos disponibles en línea, ya que las investigaciones realizadas por Vargas Sierra (2019) y Alotaibi (2020) concluyeron que eran de los puntos débiles más frecuentes de las herramientas de TAO. Si bien no podemos examinar la facilidad de aprendizaje al completo, ya que esto se encuentra fuera del alcance de este trabajo, trataremos de tener una visión general de la cantidad y calidad de los recursos de ayuda disponibles para aprender a utilizar este programa, lo que puede servir como una primera aproximación a esta cuestión.

Vamos a comprobar cuáles de estos elementos están presentes en OmegaT y trataremos de predecir las impresiones que tendría un traductor al usar OmegaT. También ofreceremos sugerencias para mejorar el programa y que se adapte mejor a las necesidades de sus usuarios.

Para este trabajo, hemos probado OmegaT durante seis meses consecutivos mientras cursábamos unas prácticas de traducción. Ya habíamos usado el programa anteriormente, pero nunca durante de manera constante y durante un período largo de tiempo, y ha sido necesario volver a aprender cómo usar tanto características que ya conocíamos como características nuevas del programa. La versión probada ha sido la 5.7.1, que era la más reciente cuando se empezó a escribir este trabajo.

4 Evaluación de la usabilidad de OmegaT

4.1 Compatibilidad e interoperatividad

OmegaT es compatible con un número muy alto de archivos, y es de los programas TAO con más compatibilidad del mercado. Los diferentes formatos con los que es compatible se pueden ver en la siguiente tabla (Oliver, 2016, p. 29):

Herramientas	Formatos
OmegaT	Plain text, HTML, XHTML, StarOffice, OpenOffice.org, OpenDocument (ODF), MS Office Open XML, Help & Manual, HTML Help Compiler (HCC), LaTeX, DokuWiki, QuarkXPress CopyFlow Gold, DocBook, Android Resource, Java Properties, Typo3 LocManager, Mozilla DTD, Windows RC, WiX, ResX, INI files, XLIFF, PO, SubRip Subtitles, SVG Images.
Open Language Tools	XLIFF, HTML, XHTML, XML, DocBook SGML, ASCII, StarOffice, OpenOffice, ODF, PO, .properties, .java (ResourceBundle), .msg, .tmsg (catgets).
SDL Trados	Cuatro entornos de traducción: TagEditor, MSWord Interface, SDLX y SDL Trados Studio 2011. Cuenta además con filtros para traducir con TagEditor: Word, Excel, PowerPoint, OpenOffice, InDesign, QuarkXPress, PageMaker, Interleaf, Framemaker, HTML, SGML, XML, SVG, etc., e in-

Herramientas	Formatos
	cluye SDL MultiTerm para la gestión terminológica, así como el Project Management Dashboard para el seguimiento y la realización de tareas automatizadas.
Virtaal	XLIFF, PO y MO, TMX, TBX, Wordfast TM, Qt.ts, y muchos otros disponibles con Translate Toolkit.
WordFast Classic	MS Word, Excel, PowerPoint (para Windows y Mac), así como documentos marcados.

Tabla 1: Formatos procesados por diversos sistemas de traducción asistida

Como se puede ver, la compatibilidad de OmegaT está a la par con la del líder de mercado, Trados Studio. Es posible ampliar aún más esa lista añadiendo el complemento de filtros de Okapi, que añade varios formatos más y mejora algunos que están disponibles en el programa base, de modo que le usuario puede abrir la mayoría de los ficheros que se utilizan de manera habitual en la profesión. Es posible utilizar memorias y ficheros traducibles generados en otras herramientas de traducción, y, en teoría, también sería posible traducir proyectos de Trados Studio si se desempaquetaran y se utilizaran los archivos individuales, aunque no hemos podido comprobar esto en la práctica, y es posible que características como el control de cambios no funcionen. Los glosarios generados por otras herramientas tendrían que convertirse a TSV y es posible que no se pudieran usar todos los campos, porque OmegaT solo soporta tres campos (ver sección «memorias de traducción y terminología»).

La compatibilidad con archivos de Word (.docx) es, en general, bastante buena y no suele dar problemas. Sin embargo, a veces aparecen demasiadas etiquetas superfluas en el editor de traducciones y hay que limpiar el documento manualmente en un procesador de textos. La experiencia que tenemos con esta situación es limitada, porque sólo hemos probado

OmegaT con textos de una cliente, y es posible que sea un error del formato del texto original, así que no podemos afirmar con certeza la frecuencia de este problema. Sin embargo, hemos encontrado usuarios con el mismo problema en varios foros de internet, así que no es un caso aislado. En cualquier caso, aunque sea un error del formato original, sería útil que el programa incorporase una función que detectase estas etiquetas superfluas y las ignorase, ya que ahorraría mucho tiempo y esfuerzo a los traductores.

Otro formato con el que tiene problemas es el PDF. OmegaT puede traducir este tipo de fichero si tiene texto insertado, pero tiene problemas para distinguir cuándo terminan las oraciones y los párrafos, y simplemente corta la unidad de traducción cuando ocurre un cambio de línea en el documento original. Algunos documentos que tienen varias columnas son inutilizables, ya que suele juntar partes de dos columnas distintas. Esto no es solo un problema de OmegaT y otros programas TAO, sino de los conversores de PDF a texto en general. Se suele recomendar a los traductores pedir a sus clientes una copia del original en un documento de texto para poder trabajar correctamente, pero si no existe la posibilidad, será, desafortunadamente, más rápido traducir manualmente con un procesador de textos que trabajar con el archivo PDF en OmegaT. Debemos esperar a que mejore la tecnología de reconocimiento óptico de caracteres (OCR, por sus siglas en inglés) para poder traducir directamente de documentos PDF.

En cuanto a la interoperatividad, OmegaT integra una amplia variedad de motores de traducción automática que incluye algunos de los más punteros como DeepL o IBM Watson. Su uso es bastante sencillo y se puede cambiar entre distintos motores en cualquier momento. La mayoría de los motores disponibles requieren usar una clave proporcionada por la empresa desarrolladora. OmegaT no incluye ninguna característica de reconocimiento de voz. Sería útil incorporar una en el futuro o desarrollar un complemento que permitiese usar esta función, porque mejoraría en gran medida la accesibilidad del programa.

4.2 Formato

OmegaT tiene un sistema de inserción de etiquetas bastante sencillo. Se pueden teclear manualmente, pero para ahorrar tiempo, OmegaT tiene varias funciones que facilitan la tarea.

En primer lugar, al hacer clic derecho sobre el texto, el menú muestra una lista de las etiquetas que faltan y se pueden insertar haciendo clic en ellas. Otra opción que ahorra aún más tiempo es utilizar el atajo de teclado Ctrl+T, que inserta la siguiente etiqueta ausente. Es un sistema menos intuitivo que el que emplea, por ejemplo, Trados, que permite insertar la etiqueta haciendo clic en ella en el segmento original, por lo que puede resultar más difícil de aprender. Sin embargo, los menús están organizados de manera bastante clara y, una vez aprendido, el atajo de teclado resulta mucho más rápido que el método con el ratón en la mayoría de los casos.

Además, tiene una herramienta de control de calidad que avisa al traductor si quedan etiquetas por insertar cuando pasa al siguiente segmento. Trataremos esta herramienta más a fondo en una sección posterior.

La mayor desventaja de OmegaT en este respecto es que es imposible agregar al texto meta formato que no esté presente en el texto original. Otros programas de traducción permiten añadir formato adicional básico al texto (cursiva, negritas, tachado, subrayado, versalitas, etc.), ya que a veces hace falta cambiarlo en el texto meta por extranjerismos u otras convenciones de la lengua. Por ejemplo, si estamos traduciendo un texto del inglés al español y nos encontramos con la palabra «*software*», en inglés aparecería en redonda, pero en español se convierte en un extranjerismo, así que sería necesario ponerlo en cursiva. La ausencia de esta característica significa que se deberá hacer un trabajo adicional de formato en un programa externo, con lo que los cambios no se guardarán en la memoria de traducción a menos que decidamos alinear los textos manualmente al final del proceso. Además, da pie a errores, porque es muy fácil olvidarse de formatear todo lo necesario.

Por otro lado, tiene una opción para pasar todo el texto seleccionado a mayúscula, minúscula, tipo título (la primera letra de todas las palabras en mayúscula, no sirve para todas las lenguas y necesita edición) y tipo oración (la primera letra de la oración en mayúscula), que pueden ser útiles en algunos casos. También permite insertar caracteres especiales de Unicode desde el menú de la aplicación.

4.3 Flujo de trabajo

En general, OmegaT es un programa relativamente rápido y con un consumo de recursos medio, aunque los tiempos de apertura son un poco largos, y, si se cargan diccionarios muy extensos, la aplicación se ralentiza considerablemente. Por suerte, este no parece ser el caso con memorias de traducción de gran tamaño. La aplicación es algo más lenta que otras con características similares por ser un programa escrito en Java, que utiliza más recursos que los programas escritos en otros lenguajes de programación por tener que cargar una máquina virtual además del programa. A pesar de todo, a la hora de traducir, la aplicación responde bien en equipos con características medias (la mayoría de las pruebas se han realizado en un ordenador con un procesador Intel i5 4210M), e incluso puede funcionar en equipos antiguos con prestaciones muy limitadas (también se ha probado muy brevemente en un equipo con un procesador Intel Atom N455), siempre y cuando no se carguen diccionarios demasiado grandes. El mayor problema de rendimiento son los tiempos de carga, pero una vez cargado completamente, el programa responde con rapidez y sin ningún problema, incluso en el equipo más modesto que hemos probado y en proyectos con varias memorias de traducción.

La interfaz de OmegaT tiene algunas diferencias con respecto a la de otros sistemas de memorias de traducción más populares, por lo que el programa puede resultar confuso para una persona que solo haya utilizado estas otras herramientas. En primer lugar, el editor de traducciones muestra la traducción debajo del original, en vez de en tabla, como es más habitual. Esta disposición tiene ventajas y desventajas; El mayor inconveniente es que la separación visual de los segmentos de origen y destino no es tan clara como en una tabla, y en ocasiones esto puede llevar a confusiones. La mayor ventaja es que se puede hacer que la interfaz se parezca más a la de un procesador de texto convencional, sobre todo si se ocultan los segmentos de origen ya traducidos y se activan las marcas de delimitación de párrafos. Esto puede ser beneficioso sobre todo a la hora de revisar, ya que a algunas personas les resulta más fácil leer las traducciones si se parecen algo más al texto final. De hecho, algunas encuestadas en (O'Brien et al., 2017, p. 159) afirmaron que les gustaría que su herramienta TAO tuviera esta característica. Otro punto positivo de que los segmentos estén unos encima de los otros es que ocupan menos espacio horizontal, por lo que es más fácil colocar OmegaT al lado de otro programa como un navegador o un lector de PDF, lo que agiliza la tarea de refe-

rencia. Quizás en el futuro se podría añadir una opción para cambiar entre ambas disposiciones para las personas que prefieren el editor en tabla. Esto ayudaría a muchos traductores a migrar desde herramientas comerciales como Trados Studio o MemoQ.

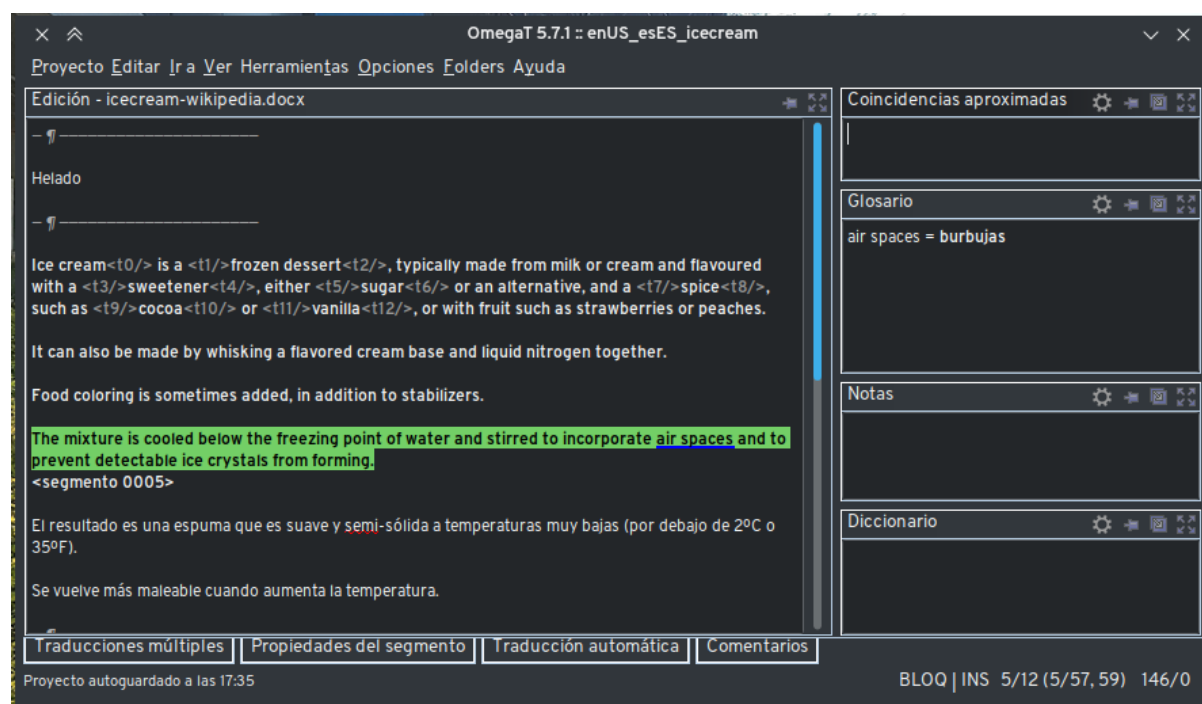


Figura 1: La interfaz principal de OmegaT con el tema GTK+

El programa también tiene capacidades muy básicas de gestión de proyectos. La barra de estado muestra la cantidad de texto traducido en el documento actual y en el proyecto y tiene una ventana de estadísticas que puede ser más o menos útil. Sin embargo, le faltan otras como la inclusión de los datos de le cliente en las propiedades del proyecto, gestión de fechas de entrega y calculadoras de precio, mencionadas en la encuesta de *ErgoTrans* (Ídem), que serían de mucha utilidad.

Además, cambiar de documento dentro del proyecto no es tan intuitivo como en otros programas, porque no hay ningún botón ni pestaña que facilite el acceso a la lista de archivos, y hay que acceder mediante el menú. Sin embargo, se puede utilizar un atajo de teclado, que

por defecto es Ctrl+L, para abrir la ventana que contiene esta lista. Un problema mayor es la interfaz del explorador de archivos en las últimas versiones, que es muy complicada y confusa. En primer lugar, coloca las carpetas y los archivos de un mismo directorio en dos paneles separados lo que, aunque es posible que ayude a encontrar lo que se busca en directorios con muchos elementos, es una disposición muy poco convencional y que puede confundir a le usuarié. Además, no existe un botón para bajar de nivel en el sistema de archivos o para acceder al directorio raíz, sino que la herramienta nos ofrece dos opciones: una lista desplegable que incluye todos los niveles anteriores al actual, y el directorio «../», que lleva al nivel directamente anterior. La primera opción puede ayudarnos a subir varios niveles de manera rápida, pero no es demasiado obvio para une usuarié novate, y la segunda es una convención muy usada en informática, pero con la que la mayoría de usuaries no están familiarizadas. Si bien ambas opciones tienen ciertas ventajas, sería conveniente añadir un botón en la barra de herramientas para subir de nivel, ya que es el método más habitual para realizar esta tarea. También podría volver a usarse el explorador de archivos de las versiones anteriores, que no tenía estos problemas.

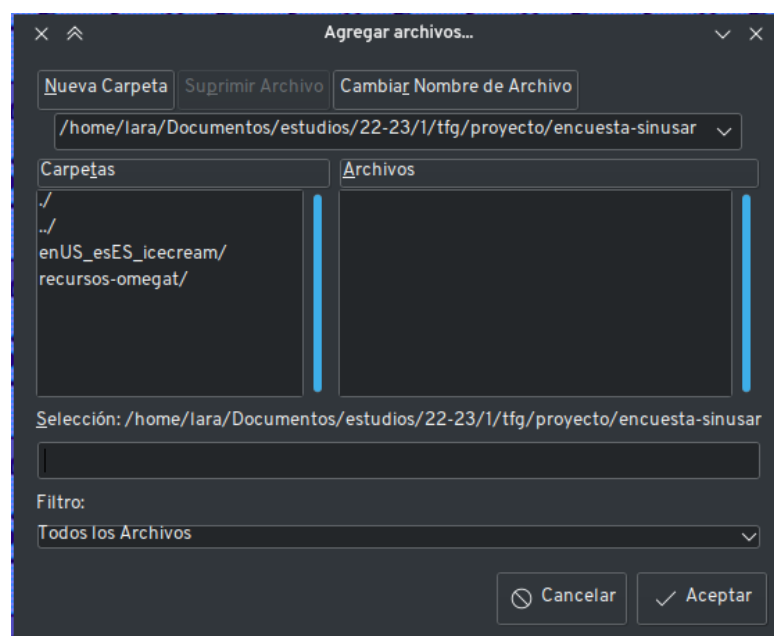


Figura 2: El gestor de archivos de OmegaT

El proceso de creación de un proyecto también tiene algunas partes que no son demasiado claras. Por ejemplo, al escribir el nombre del proyecto en el explorador de archivos, no queda claro si la carpeta del proyecto se crea automáticamente o no. Sería conveniente que el asistente de creación de proyectos guardase todo en un directorio preestablecido y creara carpetas automáticamente con el nombre del proyecto especificado por el usuario.

Algo positivo que tiene la gestión de proyectos de OmegaT es que es muy fácil añadir archivos extra, ya que el menú «Archivo» tiene un elemento dedicado a esta función.

Por último, el programa no tiene ningún tipo de barra de herramientas. Todo se controla mediante los paneles de las distintas funciones, el menú y los atajos del teclado. Sería útil añadir una barra de herramientas personalizable por el usuario, porque hay muchas personas que prefieren tener una. Las barras de herramientas con iconos agilizan el aprendizaje del programa por ser más visuales, y son más rápidas que el menú para funciones frecuentes si se utiliza el ratón. Podría darse la opción de desactivarla para los usuarios que lo prefieran.

4.4 Control de calidad

El programa cuenta con correctores gramaticales, ortográficos y de puntuación que por lo general funcionan bastante bien. El corrector que falla más a menudo es el gramatical, ya que no utiliza técnicas de aprendizaje profundo. Una función muy útil del corrector ortográfico es que está separado por dialectos, por lo que avisa a los traductores si la palabra que ha escrito no se usa en la variante asignada en el proyecto. También se pueden añadir palabras nuevas al diccionario del sistema e ignorar errores ortográficos en todo el proyecto, como es habitual en este tipo de programas.

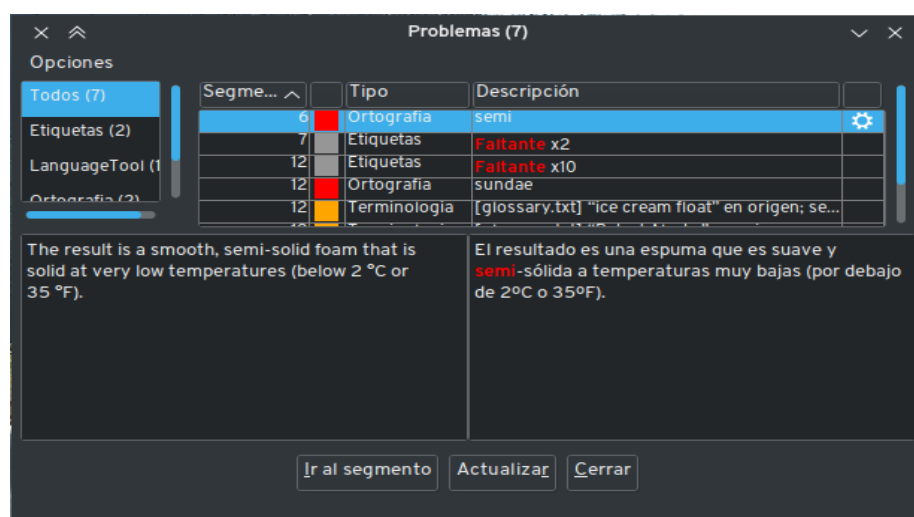


Figura 3: Solucionador de problemas de OmegaT

Tiene un comprobador de problemas que incluye errores de terminología, ortografía, MT, puntuación y formato. Es menos completo que el de otras herramientas TAO como Trados Studio, ya que no permite establecer niveles de importancia, pero también es más sencilla de utilizar. Al exportar el proyecto, notifica al usuario si aún quedan errores por corregir, y, como hemos comentado anteriormente, también notifica si quedan etiquetas por agregar en el momento de cambiar de segmento. Como es frecuente en este tipo de herramientas, permite ignorar los errores.

Puede mostrar en uno de los paneles los comentarios que se han insertado en formatos especiales para la traducción como PO. Si es un comentario del procesador de textos, aparecen después del texto principal, en el panel del editor. Además, le traductore puede insertar notas para uso en el propio programa en otro de los paneles. Esto es muy útil a la hora de corregir, y es una característica bastante habitual en las herramientas TAO.

4.5 Funciones de búsqueda

OmegaT tiene una función de búsqueda global muy completa, que permite buscar en memorias internas, externas y glosarios. Se puede acceder a ella mediante la opción del menú «Buscar en el proyecto». La herramienta busca en todos los archivos del proyecto pero, a nuestro conocimiento, no permite elegir un sólo archivo en el que buscar dentro del proyecto. Si se incluyera esta opción, la herramienta de búsqueda sería mucho más flexible. También permite buscar de manera recursiva en archivos de un directorio concreto, externo al proyecto, que es otra de las características mencionadas en (Ídem).

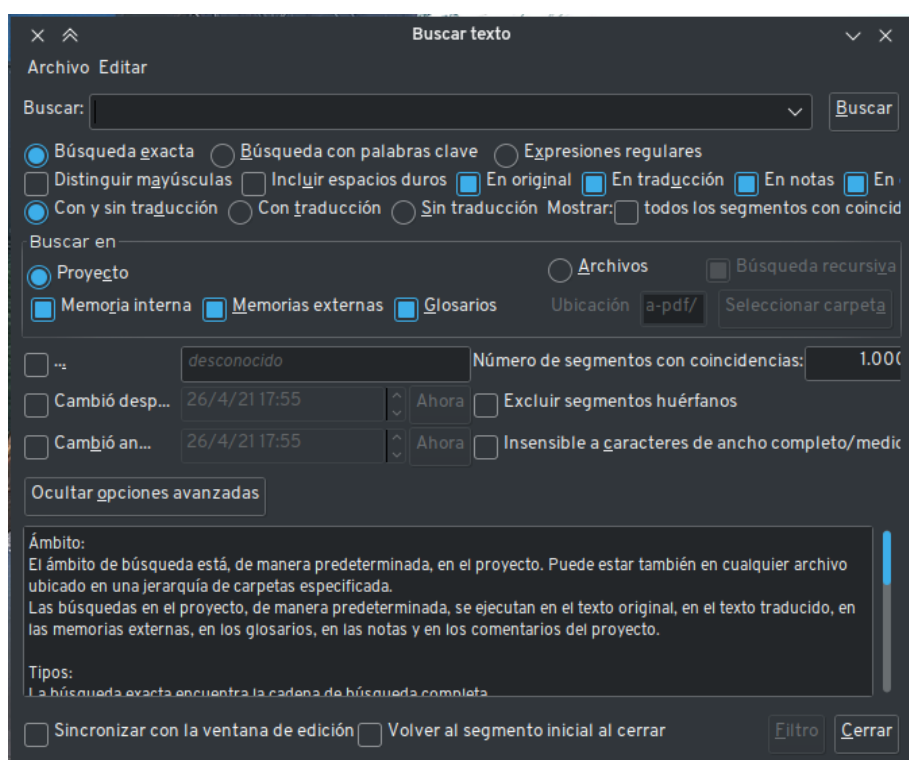


Figura 4: Ventana de búsqueda global

La función de búsqueda en diccionarios es bastante completa, aunque en la versión que hemos probado solo se podía utilizar con diccionarios descargados. En la página web de OmegaT hay un enlace a una colección de diccionarios de distintas lenguas y tipos, pero no hay ninguno lo suficientemente reciente o fiable. Otra desventaja es que si el diccionario es muy extenso, añadirlo ralentiza considerablemente el programa. Sin embargo, la herramienta en sí funciona bastante bien. Por defecto, el panel de diccionario muestra las entradas de todas las palabras de la unidad de traducción, pero cuando se selecciona una, la busca automáticamente.

Otra función más interesante son los atajos que la aplicación permite crear para buscar en diccionarios en línea en un programa externo. Sin embargo, hay que configurar la orden de búsqueda manualmente y no existe documentación en internet, así que es muy complicado de utilizar. Como la búsqueda se hace en un programa externo, como, por ejemplo, un navegador, puede entorpecer el flujo de trabajo. Sería útil si hubiera un panel con un navegador básico integrado, como lo tiene, por ejemplo, la herramienta CafeTran Espresso. Se podría añadir incluso una función con la que insertar automáticamente una palabra del navegador en la traducción. Sin embargo, como la disposición de OmegaT es tan personalizable, una manera de mitigar este inconveniente es limitar el número de paneles que están visibles en la herramienta TAO y ponerla justo al lado de la ventana del navegador, de modo que cada ventana ocupe la mitad de la pantalla. Se puede cambiar entre ambas ventanas rápidamente con Alt+Tab. No obstante, esto debería ser sólo una medida temporal y la inclusión de un navegador integrado facilitaría mucho el trabajo.

Existe una extensión que permite añadir un navegador integrado a OmegaT, pero, como veremos más adelante, instalar extensiones no es tan sencillo como en otros programas de TAO. Además, hay que añadir los sitios web a los que quieres que pueda acceder la extensión de manera manual mediante *scripts*, así que no es una herramienta adecuada para usuarios con pocas capacidades tecnológicas. Por último, la extensión depende de JavaFX, que no viene incluido en muchas instalaciones de Java, lo que añade aún más complejidad al proceso.

Por suerte, mientras realizábamos este trabajo, OmegaT ha lanzado la beta de la versión 6.0, que incluye la posibilidad de utilizar diccionarios en línea como el de Oxford mediante una clave de API, al igual que se podía hacer con los traductores automáticos. Esto es una gran ventaja y, si funciona correctamente, podría mejorar enormemente la usabilidad del programa y evitar que le usuarie tenga que cambiar de ventana para realizar consultas con tanta frecuencia. Esta es una característica que muchos programas comerciales no tienen, por lo que consideramos que OmegaT ahora tiene una ventaja en este respecto. Aunque por limitaciones de tiempo no hemos podido probar esta función aún, consideramos que su implementación es un paso en la dirección correcta.

Por desgracia, no se incluye ningún tipo de función de vista previa. Esta característica sería realmente útil para comprobar que no hay problemas en la disposición y para revisar el texto, ya que es más fácil atender al contexto si la traducción se presenta en párrafos. Sin embargo, no consideramos que la ausencia de esta característica sea tan grave como la de otras, porque la demanda es bastante baja (Ídem).

4.6 Segmentación

Uno de los mayores inconvenientes de OmegaT es que no permite juntar ni separar segmentos manualmente, una función que otras herramientas como Trados sí que tienen. Es una función que debería añadirse, porque, aunque las reglas de segmentación funcionan bien la mayor parte del tiempo, en casi todos los proyectos nos encontramos con casos en los que no segmentaban correctamente y era necesario hacer cambios. Un ejemplo muy frecuente de esto son las siglas «*b. C.*» en inglés, que se dividen en dos unidades de traducción. La solución que propone el programa actualmente son las reglas «QuickFix» que, como su nombre indica, son arreglos temporales añadidos al motor de segmentación de manera manual por le usuarie. Esta manera de abordar el problema, aunque es posible que ahorre tiempo a la larga si el mismo error aparece en varias ocasiones, es enrevesada y mucho más complicada para le traductore que simplemente juntar dos segmentos de manera manual.

4.7 Automatización

El programa automatiza por defecto algunas tareas para hacerle la vida más sencilla a los usuarios. Por ejemplo, los proyectos se guardan automáticamente cada pocos minutos, lo que puede evitar la pérdida de datos en caso de un apagón o de un despiste. No se actualiza automáticamente, pero notifica al usuario si hay una versión nueva del programa y le dirige a la página de descarga en el navegador del sistema. Esto se podría mejorar si se actualizara de manera completamente automática desde el propio programa. En versiones de GNU/Linux cuyo gestor de paquetes tiene repositorios que mantienen el programa al día esto no es un problema, porque OmegaT se actualiza automáticamente al igual que el resto de programas del sistema. Estas dos opciones se pueden desactivar en la configuración si el traductor lo desea.

Además, permite crear *scripts* para guardar los distintos ficheros de los proyectos en lugares determinados a la hora de exportar el proyecto. Se podría, por ejemplo, copiar las memorias de traducción y los glosarios respectivamente a carpetas en la que estén todos almacenados, de modo que su acceso sea más rápido en el futuro. También se puede cambiar el nombre de esos archivos automáticamente con estos *scripts*. Otra manera de automatizar el programa de manera manual sería con los anteriormente mencionados atajos de búsqueda externa pero, de nuevo, la documentación sobre esa característica es deficiente y, por el momento, no consideramos que sean utilizables.

4.8 Memorias de traducción y terminología

El sistema de memorias de traducción utiliza, al igual que la mayoría de herramientas del mercado, la distancia de edición para calcular la similitud entre el segmento de origen y la unidad de traducción almacenada. Como hemos visto anteriormente, es un sistema con muchas limitaciones, ya que no puede calcular la similitud semántica entre oraciones, lo que en ocasiones lleva a errores. Sin embargo, no hemos notado muchas desventajas en cuanto a su funcionamiento con respecto a otras herramientas TAO del mercado, por lo que consideramos que en este aspecto está a la par con ellas. Está claro que este sistema de concordancias nece-

sita mejoras, pero aún se requiere más investigación en el campo de los sistemas de memorias de traducción que utilizan métodos de aprendizaje profundo.

Una desventaja que tiene este programa es que es muy complicado cargar memorias adicionales en los proyectos. Hay que copiar de manualmente los archivos a la carpeta correspondiente, lo que puede resultar complicado para novates y, además, hace que se pierda tiempo a la hora de configurar los proyectos. Se debería incluir un botón en las propiedades del proyecto para añadir recursos como glosarios, memorias de traducción y diccionarios.

La gestión terminológica es más sencilla que en otros programas. OmegaT guarda los glosarios en archivos TSV, que tienen muy buena compatibilidad por ser un formato estándar y muy usado. Se puede exportar a programas de hojas de cálculos y, probablemente, a otros sistemas de terminología como MultiTerm, aunque no hemos podido probarlo. Esto hace que los glosarios sean muy portables y, como los archivos son texto plano, las bases de datos consumen muy pocos recursos del sistema. Por otro lado, sus funciones son muy limitadas. Solo permite añadir tres campos: lengua de origen, lengua de destino y notas. Otras herramientas tienen muchas más funciones en sus bases terminológicas como categorización más compleja y archivos multimedia de ningún tipo, por lo que importar bases terminológicas de estos programas puede causar problemas.

Su funcionamiento es bastante sencillo. Simplemente hay que seleccionar una palabra o conjunto de palabras, en origen o destino, y seleccionar «Añadir al glosario» en el menú. Permite añadir varias entradas para un mismo lema, pero en el archivo TSV se guardan por separado. Es decir, lo que en OmegaT se vería como:

palabra: 1. Lorem ipsum 2. Dolor sit amet

en el TSV se vería como:

palabra (tabulador) Lorem ipsum

palabra (tabulador) Dolor sit amet

En definitiva, consideramos que, aunque no tiene tantas funciones como otras bases terminológicas, esto puede ser ventajoso, porque la hace más fácil de utilizar y más compatible con otros programas.

4.9 Personalización

Consideramos que OmegaT es una TAO muy personalizable. Los distintos elementos de la interfaz se pueden colocar como paneles de manera libre. Se pueden tener tantos o tan pocos a la vista como se desee. También se pueden juntar varias funciones en un solo panel. Cuando se hace esto, aparecen pestañas que permiten cambiar entre ellas. Los paneles que le usuario decide minimizar se colocan en pestañas en la parte inferior de la ventana. Como sugerencia, sería conveniente que se pudieran ocultar completamente para poder limpiar aún más la interfaz, pero es posible que esto afectara a la función del programa, y, en cualquier caso, las pestañas no distraen demasiado de la función principal del programa.

El editor de traducciones también es muy personalizable. Se puede cambiar la fuente, el color y el tamaño de la letra y lo que se muestra (marcas de segmentos con entradas de glosario, con notas, autorrellenados, etc). También se puede elegir ver las marcas de salto de párrafo, lo que hace que el editor de traducciones se asemeje más a un procesador de texto tradicional. Además, se puede cambiar el color de fondo de todos los paneles en caso de que los colores por defecto resultan incómodos para la vista de le usuario.

En las últimas versiones existen varios temas para la interfaz de usuario, aunque la mayoría tienen un aspecto bastante anticuado. En sistemas GNU/Linux, existe la opción de utilizar el tema de GTK+, lo que hace que el programa se adapte al aspecto visual del sistema, de modo que la diferencia de aspectos visuales no sea molesta. Esto también permite utilizar temas oscuros, que algunas personas consideran menos molestos para la vista, sobre todo por la noche y en otras situaciones con poca luz ambiental. Si se usa Windows, hay que descargar una extensión para poder utilizar temas oscuros que es bastante complicada de instalar, lo cual es una desventaja importante. No hemos podido probar el programa en MacOS, así que no podemos decir nada sobre el aspecto visual en ese sistema.

Otro aspecto muy importante de la personalización son los atajos de teclado, que son completamente personalizables e incluyen todos los elementos del menú además de algunas opciones especiales. Esta capacidad de personalización puede ser útil en caso de estar familiarizado con otro programa de TAO que utilice atajos distintos, porque se podrían modificar los atajos para que coincidieran con el otro programa.

Existen varias extensiones para OmegaT que permiten ampliar sus funcionalidades, cambiar el tema y añadir más filtros de archivos. Sin embargo, no existe un gestor de extensiones integrado como en Trados Studio, sino que hay que copiar los ficheros de las extensiones a una carpeta del programa, lo que puede resultar confuso para usuarios sin mucha experiencia. Quizás convendría añadir una función al programa que permitiera instalar extensiones desde archivos descargados desde una página web o, aún mejor, que permitiera acceder a un repositorio de extensiones y descargarlas desde el propio programa. Esta es una característica que está presente en otros proyectos de *software* libre, como el entorno de escritorio KDE Plasma, que permite descargar temas visuales y extensiones desde su propia configuración.

4.10 Instrucción

Existen varias maneras de obtener ayuda sobre cómo usar el programa. En primer lugar, la herramienta ofrece un manual al que se puede acceder mediante el menú *Ayuda* → *Manual de usuario*, o con la tecla F1. El manual es bastante completo, pero la versión *offline* incluida con OmegaT 5.7.1 no está del todo actualizada. La versión estable más reciente al momento de escribir este trabajo es la 4.3.3, pero el manual en inglés sigue en la versión 4.3.1. El manual en español está bastante bien traducido, pero sigue en la versión 2.3.0, que data de 2012. Sin embargo, la documentación en línea para el idioma inglés está actualizada a la última versión beta.

También existe una guía de inicio rápido, que aparece siempre que se abre el programa en el panel de edición. Es una guía breve pero útil y ofrece la información esencial para poder usar el programa. Incluye apartados sobre cómo crear un proyecto, cómo traducir y cómo generar el archivo, así como información sobre las funciones de terminología, búsqueda, etiquetas y dónde encontrar los distintos archivos que genera el programa. También incluye enlaces

a otros recursos de ayuda. Es información bastante útil, aunque, de nuevo, no está del todo actualizada en las versiones más recientes, porque hay algunos botones que han cambiado de nombre.

Además, hay varios tutoriales escritos por la comunidad, entre los que destaca *OmegaT for Beginners* (Welsh et al., 2020), que además tiene una traducción al español (Welsh y Prior, 2010). Se trata de un documento que muestra las funciones básicas y guía le usuarié en el proceso de traducción. La versión en inglés está bastante actualizada, pero la última edición aún no se ha traducido al español. Hay otros tutoriales disponibles en la página web de OmegaT. También existen tutoriales básicos en YouTube que pueden resultar útiles por su naturaleza visual, aunque no hay tantos como para otros programas y, de nuevo, suelen ser bastante antiguos.

Otra opción para obtener ayuda sobre el programa son los las listas de correo, que son bastante activas, y foros de terceros como el de ProZ, que no tiene tanta actividad. La herramienta también tiene cuentas en redes sociales, pero desconocemos si les responsables de ellas responden a preguntas por esos medios.

La razón principal por la que los recursos de ayuda están tan desactualizados es que, como en muchos otros proyectos de software libre, están elaborados por voluntaries. Las funciones y la disposición de OmegaT no han cambiado demasiado a lo largo de los años, así que, aunque no son ni mucho menos perfectos, los recursos disponibles siguen siendo de ayuda para las funciones básicas, aunque es necesario ampliar y revisar la documentación para las características nuevas del programa.

5 Conclusiones y algunas propuestas de mejora

Hoy en día, la tecnología de TAO es indispensable para ejercer nuestra profesión de manera eficiente. Por ello, consideramos que tener la posibilidad de utilizar una herramienta de *software* libre de calidad a disposición de los traductores es esencial, ya que otorga más libertades al usuario. La elección de programas TAO de *software* libre es limitada y aún queda mucho por mejorar en este ámbito, por lo que sólo queda esperar que más voluntarios y empresas decidan aportar mejoras a las herramientas de este tipo, algo que ocurrirá si más personas las usan.

En cualquier caso, teniendo en cuenta las características que hemos analizado, consideramos que OmegaT es una herramienta que cumple bien sus funciones y se puede utilizar tanto para el trabajo profesional como para fines educativos en su estado actual de desarrollo. Además, incluye algunas características que sus alternativas privativas no tienen. Sin embargo, puede ser algo complicada de aprender y tiene mucho margen de mejora. La gestión de archivos, en especial, es un paso fundamental del proceso de traducción asistida por ordenador y requiere de muchos cambios para que sea más fácil y rápida de usar, como poder importar recursos desde la propia herramienta y la mejora del explorador de archivos integrado. Además, incluir opciones de formato básicas a la herramienta ahorraría mucho tiempo a los traductores, mejoraría inmensamente el flujo de trabajo y reduciría la necesidad de utilizar programas externos, ya que es muy frecuente tener que añadir marcas de formato que no están presentes en el texto original. También debería priorizarse implementar la opción de juntar y separar segmentos de manera manual, ya que la edición de reglas de segmentación es complicada de utilizar e ineficiente en algunos casos.

La personalización es uno de los puntos fuertes de la herramienta, que mejoraría aún más con la inclusión de una barra de herramientas opcional y personalizable. El rendimiento de la aplicación es muy bueno en la gran mayoría de los casos, lo que facilita el proceso de traducción y hace que podamos extender la vida útil de nuestros equipos. Además, si se utiliza con GNU/Linux, se puede ejecutar en ordenadores que otros sistemas operativos y programas TAO han dejado de soportar, con lo que podría ayudar a reducir los residuos electrónicos

generados en nuestra profesión, aunque este tema se deberá tratar en investigaciones futuras. Por último, con la integración de diccionarios en línea en la versión 6.0, consideramos que la usabilidad de la herramienta ha aumentado, ya que esta característica evita a le usuaria tener que cambiar de ventana constantemente para consultar este tipo de recursos. Esperamos que en el futuro se amplíe el catálogo de recursos en línea al que puede acceder OmegaT.

Este trabajo es meramente preliminar, y queda pendiente para investigaciones futuras realizar un estudio a mayor escala con usuarias para determinar con mayor exactitud la usabilidad y las actitudes de los traductores respecto a esta herramienta. Un estudio de estas características ayudaría aún más a determinar en qué aspectos se deberían centrar los desarrolladores de esta herramienta para mejorar su usabilidad, algo especialmente necesario teniendo en cuenta que proyectos como este suelen tener recursos más limitados que sus contrapartes comerciales, y por tanto necesitan centrar su atención en los problemas que más marcan la diferencia. También sería interesante realizar la misma tarea con otras herramientas de *software* libre de TAO o de subtitulado, y compararlas con sus alternativas comerciales.

6 Bibliografía

- Alotaibi, H. M. (2020). Computer-Assisted Translation Tools: An Evaluation of Their Usability among Arab Translators. *Applied Sciences*, 10(18), 6295. <https://doi.org/10.3390/app10186295>
- ALPAC (Organization). (1966). *Language and Machines: Computers in Translation and Linguistics: A Report*. National Academy of Sciences, National Research Council.
- Arthern, P. J. (1979). Machine translation and computerized terminology systems: A translator's viewpoint' in SNELL, BM ed. Translating and the computer. *Proceedings of a seminar, London, 14th November 1978*. <https://aclanthology.org/1978.tc-1.5>
- Asare, E. K. (2011). *An Ethnographic Study of the Use of Translation Tools in a Translation Agency: Implications for Translation Tool Design*. http://rave.ohiolink.edu/etdc/view?acc_num=kent1310587792
- Bevan, N. (1995). Usability is Quality of Use. En *Advances in Human Factors/Ergonomics* (Vol. 20, pp. 349-354). Elsevier. [https://doi.org/10.1016/S0921-2647\(06\)80241-8](https://doi.org/10.1016/S0921-2647(06)80241-8)
- Chatzitheodorou, K. (2015). *Improving translation memory fuzzy matching by paraphrasing*. <https://aclanthology.org/W15-5204>
- Dillon, A. (2001). The evaluation of software usability. En *Encyclopedia of Human Factors and Ergonomics*. London: Taylor and Francis. <http://hdl.handle.net/10150/105344>
- Gupta, R., & Orasan, C. (2014). *Incorporating paraphrasing in translation memory matching and retrieval*. https://www.researchgate.net/publication/228366548_Formalizing_Translation_Memory

- Hódasz, G., & Pohl, G. (2005). MetaMorpho TM: A linguistically enriched translation memory. En W. von Hahn, J. Hutchins, & C. Vertan, *International Workshop Modern Approaches in Translation Technologies: Proceedings ; Borovets, Bulgaria, 24 September 2005* (pp. 26-31). Incoma. https://www.researchgate.net/publication/228772807_Evaluation_Methods_of_a_Linguistically_Enriched_Translation_Memory_System
- Hutchins, J. (1998). The origins of the translator's workstation. *Machine Translation*, 13(4), 208-307.
- ISO/IEC. (2016). *International Standard ISO/IEC 25022 Systems and software engineering—Systems and software quality requirements and evaluation (SQuaRE)—Measurement of quality in use ISO/IEC 25022:2016* (Versión 1). <https://www.iso.org/standard/35746.html>
- Kirakowski, J. (1996). Background notes on the sumi questionnaire: The use of questionnaire methods for usability assessment. *Dostępne w Internecie: http://sumi. ucc. ie/sumi-papp. html [dostęp 25.03. 2012]*.
- Krollmann, F. (1971). Linguistic Data Banks and the Technical Translator. *Meta: Journal des traducteurs*, 16(1-2), 117. <https://doi.org/10.7202/003352ar>
- Lagoudaki, P. M. (2009). *Expanding the possibilities of translation memory systems: From the translators wishlist to the developers design* [PhD Thesis, Imperial College London]. <https://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.501201>
- LeBlanc, M. (2013). Translators on translation memory (TM). Results of an ethnographic study in three translation services and agencies. *The International Journal of Translation and Interpreting Research*, 5(2). <https://doi.org/10.12807/ti.105202.2013.a01>

- Libro Blanco de los derechos de autor de las traducciones de libros en el ámbito digital.*
(2016). Ace Traductores. <https://ace-traductores.org/publicaciones/estudios-sobre-el-sector/>
- O'Brien, S., Ehrensberger-Dow, M., Connolly, M., & Hasler, M. (2017). Irritating CAT Tool Features that Matter to Translators. *HERMES - Journal of Language and Communication in Business*, 56, 145-162. <https://doi.org/10.7146/hjlc.v0i56.97229>
- Oliver, A. (2016). *Herramientas tecnológicas para traductores*. UOC. https://brumario.usal.es/permalink/34BUC_USAL/edmht/cdi_proquest_ebookcentral_EBC4721948
- Pekar, V., & Mitkov, R. (2007). New generation translation memory: Content-sensitive matching. *Proceedings of the 40th anniversary congress of the swiss association of translators, terminologists and interpreters*. <https://aclanthology.org/2021.ranlp-srw.25>
- Planas, E., & Furuse, O. (1999). *Formalizing translation memories*. https://www.researchgate.net/publication/228366548_Formalizing_Translation_Memory
- Ranasinghe, T., Orasan, C., & Mitkov, R. (2020). *Intelligent Translation Memory Matching and Retrieval with Sentence Encoders* (arXiv:2004.12894). arXiv. <http://arxiv.org/abs/2004.12894>
- Reinke, U. (2004). *Translation Memories: Systeme-Konzepte-Linguistische Optimierung*. Europäischer Verlag der Wissenschaften. <https://doi.org/10.1007/s10590-007-9020-3>
- Reinke, U. (2018). *State Of The Art In Translation Memory Technology*. Zenodo. <https://doi.org/10.5281/ZENODO.1291930>
- Seal, T. (1992). *ALPNET and TSS: the commercial realities of using a computer-aided translation system*. <https://aclanthology.org/1991.tc-1.11>

- Simard, M., & Isabelle, P. (2009). *Phrase-based machine translation in a computer-assisted translation environment*. <https://aclanthology.org/2009.mtsummit-papers.14>
- Stallman, R. M. (, & Lessig, L. (, (2004). *Software libre para una sociedad libre* (2^a ed). Traficantes de Sueños. <https://traficantes.net/sites/default/files/pdfs/Software%20libre-Traficantes%20de%20Sue%C3%B1os.pdf>
- Utiyama, M., Neubig, G., Onishi, T., & Sumita, E. (2011). *Searching translation memories for paraphrases*. <https://aclanthology.org/2011.mtsummit-papers.37>
- Vanallemersch, T., & Vandeghinste, V. (2014). *Improving fuzzy matching through syntactic knowledge*. <https://aclanthology.org/2014.tc-1.11>
- Vargas Sierra, C. (2019). La evaluación de la usabilidad de un sistema de memoria de traducción. *Quaderns de Filologia - Estudis Lingüístics*, 24(24), 119. <https://doi.org/10.7203/qf.24.16302>
- Welsh, S., & Prior, M. (2010). *OmegaT para principiantes TAO*. https://omegat.org/files/guide/OmegaT_for_Beginners_es.pdf
- Welsh, S., Prior, M., & Miura, H. (2020). *OmegaT for CAT Beginners*. https://hackmd.io/@omegat/for_cat_beginners
- Zaretskaya, A., Corpas Pastor, G., & Seghiri, M. (2018). User Perspective on Translation Tools: Findings of a User Survey. En I. Durán-Muñoz, *Trends in E-Tools and Resources for Translators and Interpreters*. BRILL. <https://doi.org/10.1163/9789004351790>