Tesis de Máster de IARFID:

Hibridación en lenguas distantes

Going Hybrid in distant languages

Alexandre Helle

Septiembre del 2013

Universidad Politécnica de Valencia
Departamente de Sistemas Informáticos y Computación

Alumno
Alexandre HELLE

Supervisores
Dr. Francisco CASACUBERTA
Antonio LAGARDA

Tabla de contenidos

Resumen

Este trabajo de tesis está basado en su totalidad a la traducción automática, una disciplina dentro de la inteligencia artificial y de la lingüística computacional. Todo el desarrollo de la tesis está dentro de la aproximación estadística y aplicación de reglas lingüísticas para mejorarla.

Veremos las distintas aproximaciones que ha habido a la traducción automática, tanto para la traducción estadística como para la traducción basada en reglas, hasta llegar al estado del arte actual, que es en la combinación de ambas.

También veremos un resumen de las principales aplicaciones de software, tanto libres como comerciales, que usan la traducción automática.

Uno de los problemas, donde suele fallar más la traducción automática estadística, es en los pares lingüísticos en el que una de las lenguas es más compleja lingüísticamente, que la otra lengua, lo que llamaremos lenguas distantes. Para minimizar dicho problema propondremos distintas reglas lingüísticas, entre ellas el reordenamiento de la estructura gramatical, o la tokenización de lenguas donde no se utilizan los espacios entre las palabras o tokens de la frase.

Todo este trabajo, tiene como finalidad última, la aplicación de todas las mejoras propuestas dentro del sistema comercial de traducción automática PangeaMT.

1 Introducción

Pangeanic, la empresa de traducción en la que trabajo, uno de los servicios que ofrece es la traducción automática. Dicho servicio se ofrece a través de la herramienta PangeaMT, desarrollada entre el ITI y Pangeanic [YHH+12], y está creada usando como base el popular toolkit de traducción automática estadística Moses [KHB+07], y sobre el cual se han aplicando reglas lingüísticas para mejorar sus resultados, dando lugar a un sistema "híbrido".

Algunos de los problemas que podemos encontrar al ofrecer traducción automática, es que al traducir entre lenguas muy distintas lingüísticamente, los resultados en ocasiones no son lo bastante buenos. Dichos problemas algunas veces vienen originados porque el orden de las palabras entre la lengua destino y la lengua origen difieren tanto que los sistemas de traducción automática acaban dejando las palabras en la posición incorrecta. Para minimizar este problema, en esta tesis de máster plantearemos unas propuestas de reordenación, en nuestro caso del inglés, que tendrá lugar durante el preproceso, para que cuando traduzcamos al japonés el orden de las palabras de la salida del sistema sea lo más cercano al orden correcto.

Otro problema que podemos encontrar, es que en ocasiones no hay suficientes datos de entrenamiento o que los datos tienen tantos valores atípicos, que acaban afectando a la calidad de la traducción. Por lo que también propondremos en esta tesis de máster una propuesta de detección y exclusión de valores atípicos para así tratar de minimizar su impacto en la calidad del sistema.

La tesis está dividida en distintos capítulos:

- Introducción

  En este primer capítulo de introducción, veremos un resumen de las distintas aproximaciones históricas que ha habido a la traducción automática, tanto para la traducción estadística como para la traducción basada en reglas, hasta llegar al estado del arte actual, que es la combinación de ambas. También veremos un resumen de técnicas de la lingüística computacional que son aplicadas a la traducción automática.

- Lenguas distantes

  En este capítulo explicaremos desde el punto de vista lingüístico las distintas lenguas con las que trataremos en los experimentos, y que entre ellas llamaremos lenguas "distantes" por ser entre si bastantes distintas desde el punto de vista lingüístico.

- Aplicaciones de MT

  Aquí veremos un resumen de las aplicaciones de MT más usadas en el mundo profesional de la traducción, las cuales las clasificaremos en dos puntos de vista, las de licencia libre y las comerciales.

- Propuestas de hibridación y optimización

  En este capítulo veremos distintas propuestas para tratar de solventar los problemas lingüísticos de algunas de las lenguas lenguas distantes. También veremos que, incluso en corpus producidos en el ámbito de la industria de la traducción, se necesita aplicar un proceso de exclusión/detección de valores atípicos (o sucios) ya que todos estos afectarán en la creación de motores de traducción estadística.

- Experimentación

  Aquí explicaremos los experimentos, que se han centrado en comprobar la validez de las distintas propuestas presentadas y además ver si el estado del arte actual en MT (los modelos basados en frases) siguen siendo válidos para lenguas tan distantes lingüísticamente como el japonés y el inglés.

- Conclusiones

  En este capítulo se resumen las aportaciones más relevantes de esta tesis, así como las principales conclusiones que podemos sacar de este trabajo.

- Futuro trabajo

  En este capítulo presentamos las futuras direcciones hacia donde se desarrollaran algunos de los puntos presentados en esta tesis de máster.

1.1. Aproximaciones históricas a la Traducción Automática (MT)

Podemos clasificar las distintas aproximaciones históricas que ha habido en la traducción automática basándonos en la tecnología usada: los sistemas basados en reglas y los sistemas empíricos o basados en reglas.

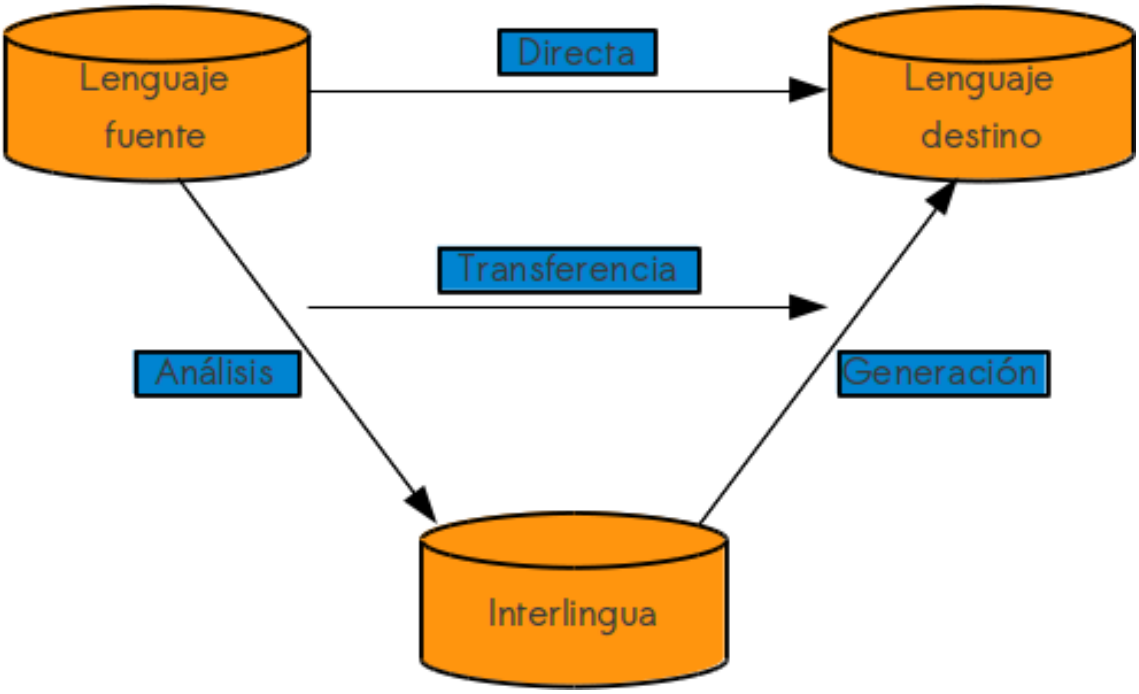
1.1.1. Sistemas basados en reglas

En los sistemas basados en reglas los expertos establecen una serie de reglas de cómo debe realizarse la traducción. Normalmente la creación de esas reglas requiere de un gran trabajo humano, en el que se necesita el conocimiento de lingüísticas expertos en ambas lenguas.

En la traducción por reglas destacan dos métodos que se caracterizan por utilizar representaciones intermedias. Por este motivo, se conocen como métodos indirectos

de traducción por reglas. Existen dos tipos de métodos indirectos: el de interlingua y el de transferencia. Así pues, una distinción preliminar es la que distingue los métodos directos de los indirectos.

Figura 1.1: Sistemas basados en reglas



1.1.1.1. Traducción directa

En los métodos directos no se utilizan representaciones intermedias y la traducción se realiza palabra a palabra, en donde solamente se realiza análisis morfosintáctico, el cual trata de identificar categorías gramaticales y otra información como genero, número, tiempo, etc. Pero esta técnica tan sencilla de traducción palabra por palabra hace tiempo que se ha abandonado por inviable y todos los sistemas conocidos proclaman utilizar representaciones intermedias.

1.1.1.2. Método de Interlingua

El método de interlingua plantea la traducción a través de una única representación conceptual independiente de las lenguas entre las que se va a traducir, lo que permite que la traducción se realice en sólo dos fases: análisis -del texto origen- y generación -de la traducción a la lengua destino-. Para llegar a esa representación conceptual el texto debe ser entendido antes de ser traducido y solo tener una correspondencia entre cada lengua y la Interlingua.

DLT (de la empresa holandesa BSO) que utiliza el esperanto como interlingua, y ROSETTA (PHILIPS), basada en la gramática de Montague, son ejemplos de sistemas basados en este método. Este método fue muy popular en Japón durante los setenta y ochenta, donde muchas empresas de gran tamaño contaban con su propio proyecto de traducción, como ATLAS (Fujitsu) o PIVOT (NEC).

1.1.1.3. Método de transferencia

El método de transferencia propone dos representaciones intermedias a cada lengua. La traducción se realiza en tres fases: análisis, transferencia y generación. La transferencia se puede realizar a distintos niveles:

- Transferencia léxica: la búsqueda del término equivalente en la lengua meta se realiza a partir de información contenida en el diccionario.
- Transferencia sintáctica: el árbol de análisis de la oración de origen se transforma en un árbol de generación equivalente para la oración meta.
- Transferencia semántica: se transforman representaciones profundas, como patrones de casos, redes semánticas, o estructuras lógicas.

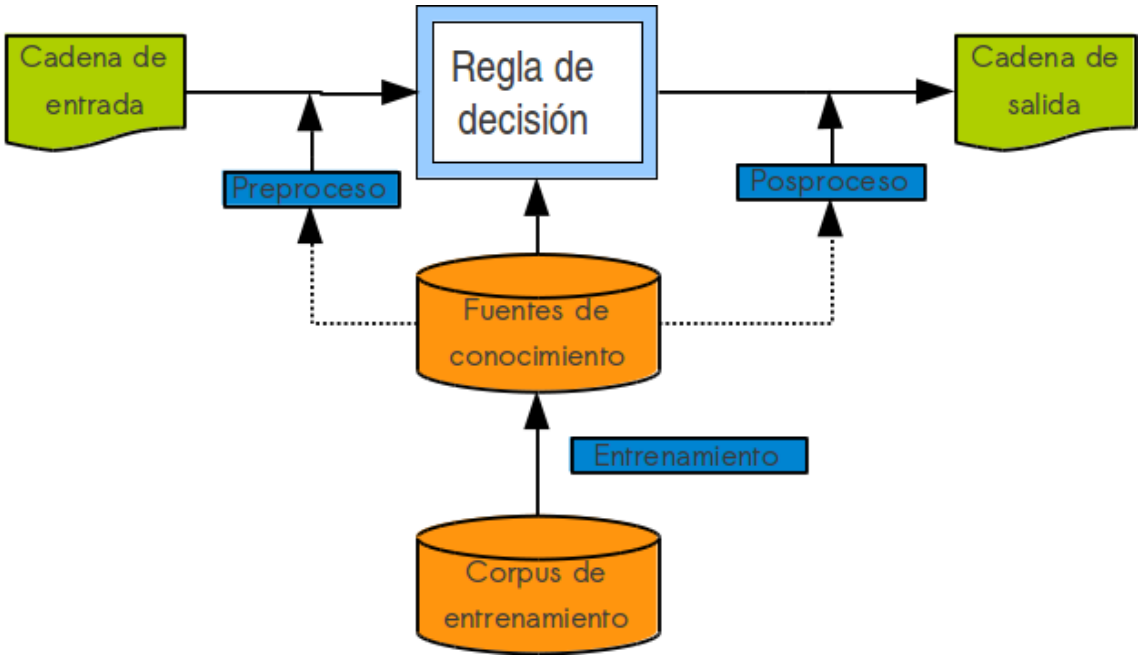
METAL, MÉTÉO, SUSY, EUROTRA, LOGOS y GETA son ejemplos de sistemas basados en este método.

1.1.2. Sistemas basados en corpus

En los noventa, se produjo un cambio en el enfoque de la traducción automática. Gracias al boom de internet que produjo que proliferaran las colecciones de textos en formato electrónico invitó a usar los métodos probabilísticos y conexionistas que tan buen resultado había dado en el reconocimiento de voz.

Los sistemas basados en corpus (ver figura 1.2) se caracterizan es que toda la información que usan para construir (o entrenar) el sistema son corpus de ejemplos de traducciones entre las lenguas origen y destino, y todo ello realizado de manera automática. Pero dicho corpus de entrenamiento tiene que ser de gran tamaño.

Figura 1.2: Arquitectura sistemas basados en corpus



Dichos corpus deben haber sido generado por traductores expertos, y durante la fase de entrenamiento se obtienen todas las fuentes de conocimientos para el par de lenguas escogidos. Dicho entrenamiento puede llevar implícito una fase de preproceso del corpus para facilitar el entrenamiento. La regla de decisión debe combinar de forma óptima todo lo aprendido en las fuentes de conocimiento para de dicho modo obtener la traducción óptima en la lengua destino. Adicionalmente también puede hacerse una fase de preprocesado -de la lengua origen- y posprocesado -de la lengua destino-. Dicho preprocesado debe ser el mismo que el realizado en la fase de entrenamiento, y el posprocesado debe deshacer dichos cambios extras añadidos en el preprocesado.

Basados en corpus existen dos grandes métodos, los basados en ejemplos y los estadísticos.

1.1.2.1. Sistemas basados en ejemplos (EBMT)

Este sistema fue inicialmente propuesto en 1981 por el investigador japonés Makoto Nagao [Nag84], pero la técnica no fue probada hasta finales de la década de manera simultánea por el propio Nagao en la Universidad de Kyoto y por el grupo del proyecto DLT [Sad89] en Holanda.

El método se basa en que los textos traducidos pueden servir de modelo a las nuevas traducciones. El proceso de traducción de una nueva frase con estos sistemas se obtiene mediante: una comparación de fragmentos con una base de ejemplos reales,

una identificación de la traducción de los fragmentos y una combinación de esos fragmentos traducidos para obtener la traducción.

1.1.2.2. Sistemas estadísticos (SMT)

La estadística como método útil en traducción automática ya fue objeto de reflexión por parte de Warren Weawer en 1949. Weaver propuso que el uso de técnicas estadísticas de la teoría de la información (rama de la matemática aplicada) podían hacer posible el uso de ordenadores digitales para traducir texto de un lenguaje natural a otro automáticamente.

A pesar de que la idea de Weaver era inconcebible debido a lo limitado que eran los ordenadores del momento, un grupo de investigadores de IBM, a finales de los 80, pensó que los avances de la computación en los últimos cuarenta años hacían razonable la aplicabilidad de las técnicas estadísticas en la traducción. Así, en 1990, nace el proyecto CANDIDE de IBM [BCP+90]. El experimento se realizó sobre el corpus Hansard de las Actas del Parlamento canadiense (unos tres millones de oraciones en inglés y francés). Primero se alinearon oraciones, grupos de palabras y palabras sueltas, para después calcular las probabilidades de que una palabra de una oración en una lengua se correspondiera con otras palabras en la traducción.

1.1.3. Sistemas basados en memorias (MBMT)

Estos sistemas, también conocidos como sistemas de traducción asistida por ordenador (o CAT), están basados en "memorias de traducción" (TM). Se pueden considerar un caso particular de los sistemas basados en corpus.

La técnica consiste en almacenar traducciones, realizadas manualmente y validadas por un traductor humano, para reutilizarlas posteriormente en la traducción de textos similares. Esta tecnología ha sido llevada al mercado con un considerable éxito en paquetes de software que incluyen los módulos de gestión de las memorias, además de programas para crear y mantener bases de datos terminológicas, alineadores automáticos y filtros para la conversión de formatos.

Algunos de los más conocidos son: DÉJÀ-VU (ATRIL), TRADOS Workbench, SDL Studio, SDLX, WORDFAST, WordBee, MemoQ, etc.

1.2. Traducción automática estadística (SMT)

Tal como habíamos introducido en la sección Subsubsección 1.1.2.2, las ideas que hay detrás de la traducción automática estadística vienen de la teoría de la información.

Esencialmente, el documento se traduce en la probabilidad $p(e|f)$ de que una cadena e de la lengua destino sea la traducción de una cadena f en la lengua fuente.

El Teorema de Bayes se aplica a $p(e|f)$, la probabilidad de que la cadena de la lengua fuente produzca la cadena destino para conseguir $p(e|f) \propto p(f|e)\, p(e)$, donde el modelo de traducción $p(f|e)$ es la probabilidad de que la cadena destino sea la traducción de la cadena fuente, y el modelo de lenguaje $p(e)$ es la probabilidad de ver aquella cadena destino. Matemáticamente hablando, encontrar la mejor traducción$\tilde{e}$ se consigue escogiendo aquella que de la probabilidad más alta:

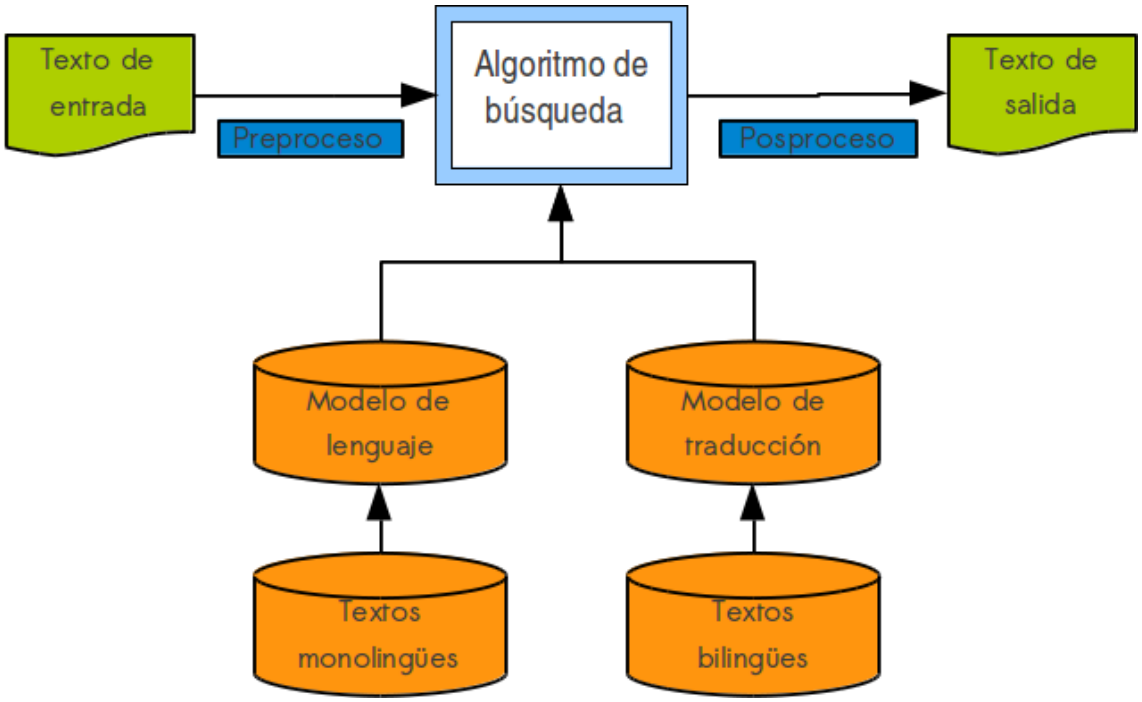
$$\tilde{e} = \underset{e \epsilon e^*}{arg\,max}\; p(e|f) = \underset{e \epsilon e^*}{arg\,max}\; p(f|e)\, p(e)$$

Esta es la fórmula fundamental de la traducción automática estadística.

Los modelos de lenguaje son típicamente aproximados por modelos de n-grama suavizados.

Los modelos de traducción estuvieron inicialmente basados en palabras (Modelos 1-5 de IBM Ocultos de Markov Model de Stephan Vogel y el Modelo 6 de Franz-Joseph Och), pero se lograron avances significativos con la introducción de los modelos basados en frases [KOM03] (ver figura 1.3), que son estado actual del arte, aunque se han tratado de mejorar con la introducción de sintaxis o estructuras casi sintácticas [YK01] [Chi05].

Figura 1.3: Arquitectura sistemas basados en estadística



El algoritmo de búsqueda (o decodificación) busca la frase que maximiza la probabilidad de traducción ($p(f|e)\, p(e)$).

1.2.1. Modelos de traducción basados en palabras

En la traducción basada en palabras, la unidad esencial de la traducción es una palabra de un lenguaje natural. Normalmente, el número de palabras en frases traducidas son diferentes, por palabras compuestas, morfología y modismos. La relación de la longitud de las secuencias de palabras traducidas se llama fertilidad, que indica el número de palabras en la lengua fuente que cada palabra en la lengua destino produce.

Necesariamente se asume por la teoría de la información, que cada una cubre el mismo concepto. En la práctica esto no es realmente verdad. Por ejemplo, la palabra inglesa *corner* puede ser traducida en español por rincón o bien por esquina, dependiendo de si es en el sentido de su ángulo interno o externo.

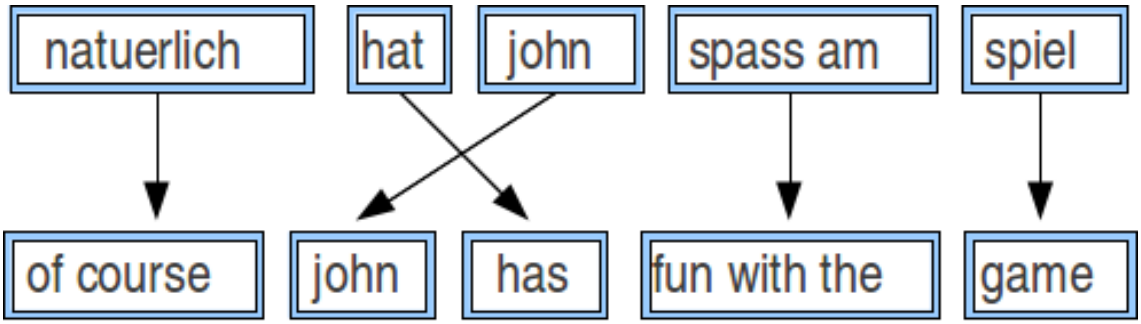
La traducción simple basada en palabras no se puede traducir entre lenguas con distinta fertilidad. Sistemas de traducción basados en palabras relativamente simples pueden ser hechos para hacer frente a altas tasas de fertilidad, pero podrían asignar una sola palabra a varias palabras, pero no al revés. Por ejemplo, si quisiéramos traducir del francés al Inglés, cada palabra en Inglés podría producir cualquier cantidad de palabras francesas, y no al revés. No hay forma de agrupar dos palabras en Inglés para producir una sola palabra francesa.

Un ejemplo de un sistema de traducción basado en la palabra es el paquete de distribución libre GIZA++ [ON03], que incluye el programa de entrenamiento para modelos de IBM y los modelos HMM y 6. La traducción basada en la palabra no se usa ampliamente hoy en día, los sistemas basados en frases son más comunes. La mayor parte de sistemas basados en la frase siguen utilizando GIZA++ para alinear el corpus. Los alineamientos se utilizan para extraer frases o deducir reglas de sintaxis.

1.2.2. Modelos de traducción basados en frases

En la traducción basada en frases [KOM03] se han intentado reducir las restricciones producidas por la traducción basada de palabras traduciendo secuencias de palabras a secuencias de palabras (ver figura 1.4), donde las longitudes de la frase nativa y la extranjera pueden ser diferentes.

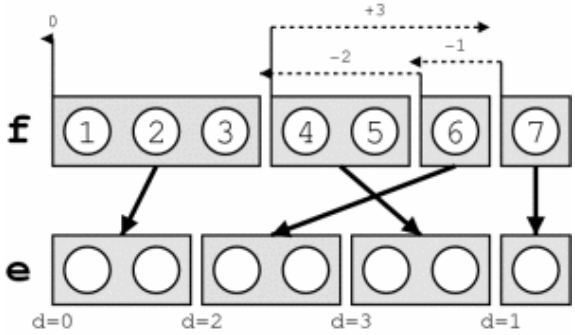
Figura 1.4: Alineación en modelos basados en frases



A las secuencias de palabras se les suele llamar bloques o frases, pero típicamente no son frases lingüísticas sino frases encontradas en el corpus utilizando métodos estadísticos.

Los bloques traducidos son reordenados para generar una salida coherente. Estos modelos no tienen un representación explícita de como reordenar las frases. Para evitar problemas de búsqueda, muchos sistemas colocan un límite en la distancia que los elementos del segmento fuente pueden ser movidos dentro del segmento destino. Este límite, junto con el límite de la longitud de frase, determinan el alcance del sistema de reordenamiento (o modelo de distorsión) en un sistema basado en frases (ver figura 1.5).

Figura 1.5: Distancia de reordenamiento



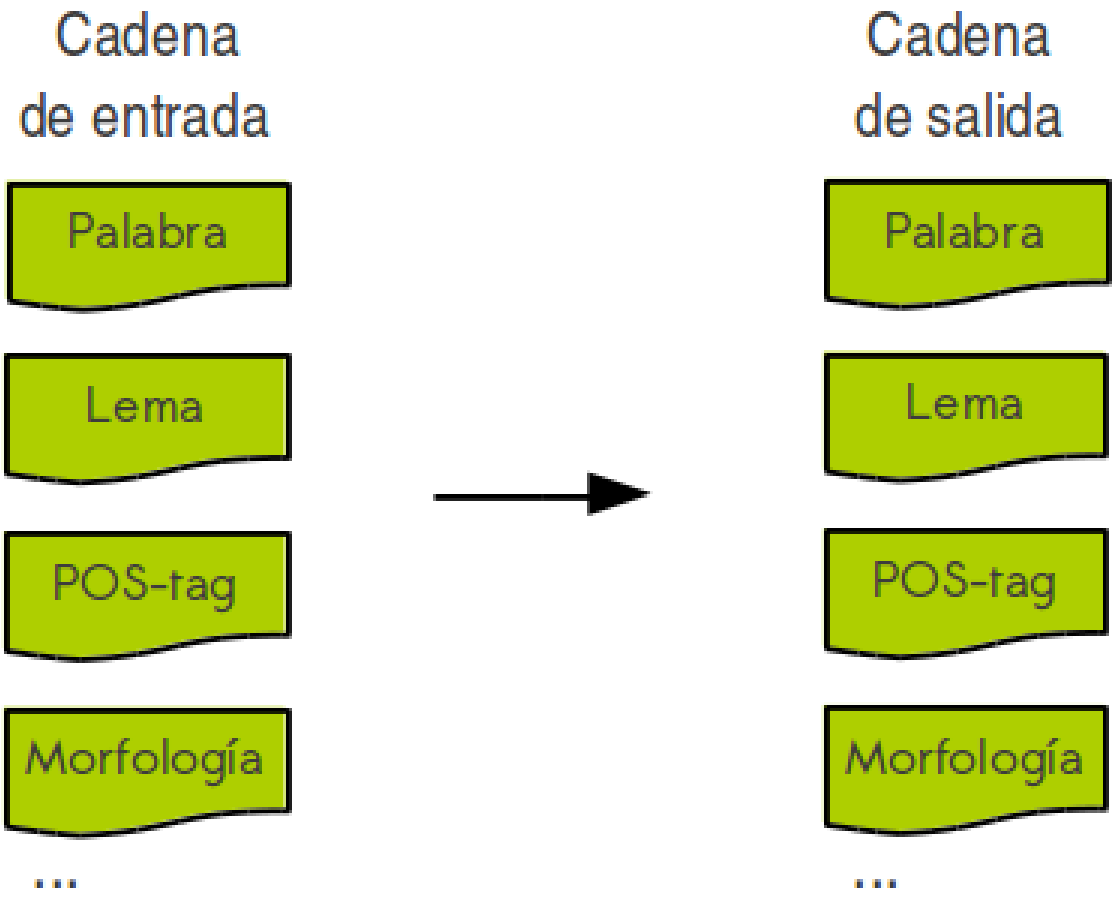
Este reordenamiento es más adecuado para reordenamientos locales que pueden haber sido contemplados en los modelos de lenguaje. Pero para los reordenamientos a larga escala son frecuentemente arbitrarios y afectan negativamente a la calidad de la traducción.

1.2.3. Modelos de traducción factoriales

Los modelos de traducción factoriales [BK03] [Axe06] son una extensión de los modelos basados en frases, que permiten la integración de información adicional lingüística

(lema, categoría gramatical...) a nivel de palabra en los modelos de lenguaje (ver imagen 1.6).

Figura 1.6: Modelos factoriales



La principal diferencia entre los modelos basados en frases y estos reside en la preparación de los datos de entrenamiento y el tipo de modelos aprendidos de los datos.

1.2.4. Modelos sintácticos

Los sistemas basados en modelos sintácticos, también conocidos como modelos jerárquicos o basados en árboles, siguen varias metodologías diferentes. Por lo que no pueden ser explicados de un modo unificado.

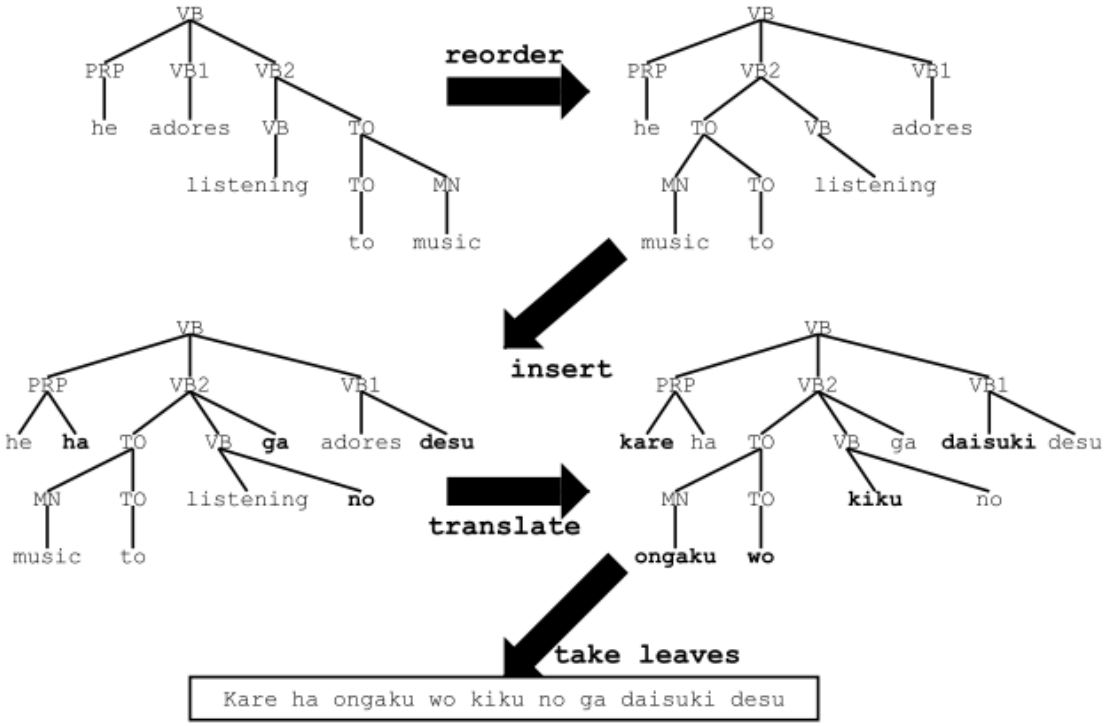
Algunas de estas aproximaciones usan un árbol parseado ("tree") como entrada, donde aplican transformaciones y obtienen una cadena ("string") como resultado

del árbol [YK01]. Estos sistemas también pueden llamarse decodificadores árbol-a-cadena ("tree-to-string"). La frase de entrada es preprocesada por un parseador sintáctico; y posteriormente el decodificador (algoritmo de búsqueda) realiza algunas operaciones sobre los nodos del árbol:

- reordenación de nodos hijo
- inserción de palabras extra en cada nodo
- traducción de los nodos hoja

Finalmente una frase se consigue como resultado de dicho árbol. Podemos ver un ejemplo del proceso de decodificación en la siguiente figura 1.7 (fuente [YK01]).

Figura 1.7: Modelos sintácticos



Otras aproximaciones obtienen una gramática estocástica (SDTS) [AU69] a partir de los datos de entrenamiento. Una SDTS es una gramática que genera simultáneamente cadenas de dos lenguas. Entonces, para producir la traducción de frase, parsea la cadena de texto de entrada y obtienen una cadena de texto de salida al mismo tiempo. En [GHKM04], los autores presentan un algoritmo que obtiene el conjunto mínimo de reglas de transformación sintácticas ("synchronous grammar rules") a partir de un corpus paralelo. Usando el árbol parseado de la salida y los alineamientos de entrada-salida del conjunto de entrenamiento, obtienen una derivación de SDTS. Finalmente extraen las reglas de las derivaciones, y prueban que

su sistema puede explicar las transformaciones de cualquier árbol parseado del texto de entrada, en una cadena del texto de salida. Finalmente en [Chi05] es presentado un trabajo que usa gramáticas sin contextos estocásticas SCFG obtenidas a partir de corpus paralelos siguiendo un método de dos pasos:

1. Identificar los pares de frases iniciales siguiendo el método común de extracción de frases.
2. Entonces obtienen las reglas de las frases buscando frases que contengan otras frases y reemplazando las subfrases por nodos no-terminales.

ne X1 pas -> not X1 (French-English)

El resultado de este algoritmo es un conjunto de reglas muy grande, por lo que deben ser filtradas siguiendo alguna restricciones, y en orden de suavizar el sistema otras reglas son introducidas. Las reglas son clasificadas usando un modelo log-lineal. Finalmente, es usado como decodificador un parseador CKY con un búsqueda en haz ("beam search") junto con un posprocesador para mapear las derivaciones de entrada en las cadenas de salida.

Hay que tener en cuenta que las reglas usan solo un símbolo no-terminal, y no tienen información lingüística. Una extensión de este trabajo que usa más símbolos no-terminales con significados lingüísticos es presentado en [ZV06].

ne VB pas -> not VB (French-English)

1.3. Sistemas híbridos de traducción automática

En el principio de los años noventa, las aproximaciones estadísticas y las basadas en reglas eran vistas en estricto contraste. Pero los puntos fuertes y flojos son complementarios (fuente [Eis07]):

Cuadro 1.1: Comparación sistemas traducción automática

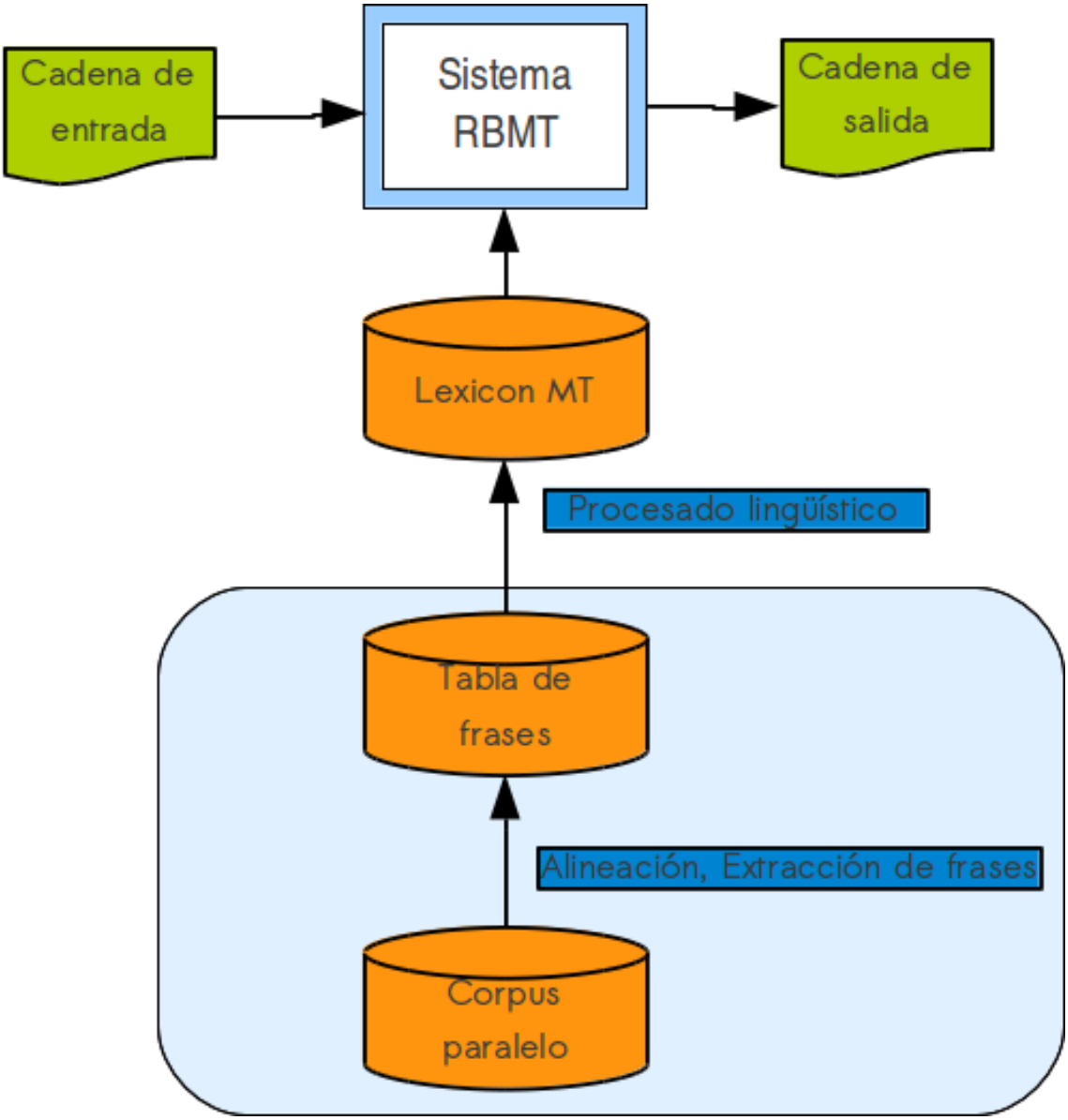
	Sintaxis	Semántica estructural	Semántica léxica	Adaptabilidad léxica
RBMT	++	+	–	–
SMT	–	–	+	+
EBMT	–	–	–	++

Los sistemas híbridos (HMT) aprovechan los puntos fuertes de las metodologías de traducción estadística y de traducción por reglas. Varias entidades relacionadas con la traducción automática (Asia Online, Systran, PangeaMT, Apertium...) dicen tener un aproximación híbrida hoy en día.

La combinación de ambas metodologías se puede hacer de dos modos distintos:

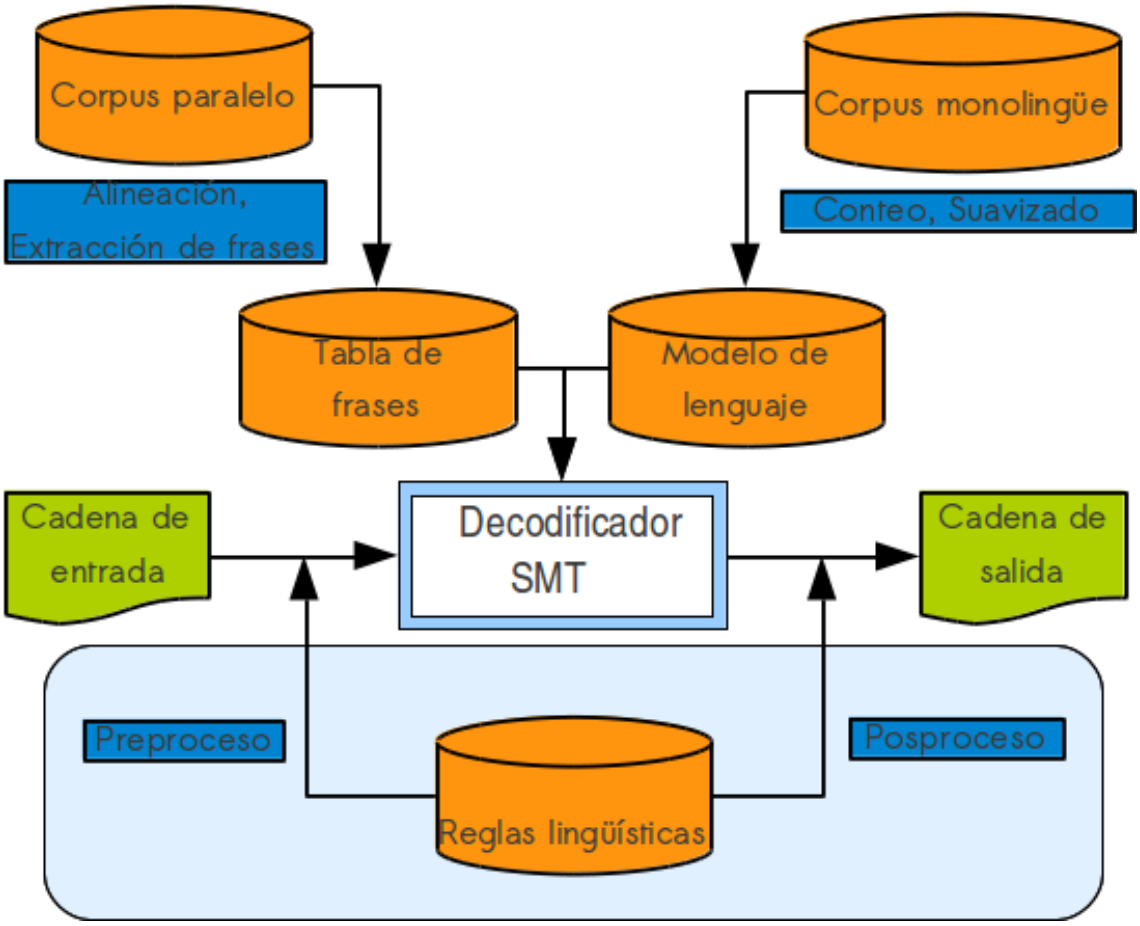
- Reglas posprocesadas por estadística (ver figura 1.8): las traducciones son realizadas usando un sistema basado en reglas. Entonces se utiliza estadística para corregir/ajustar la salida o para aumentar su base de conocimiento.

Figura 1.8: Sistema basado en reglas guiadas por estadística



- Estadísticas guiadas por reglas (ver figura 1.9): se utilizan reglas durante el preproceso de los datos en un intento de mejorar/facilitar la traducción estadística. Reglas también pueden ser usadas en el output para, entre otras cosas, realizar funciones de normalización. Esta aproximación tiene mayor poder, flexibilidad y control en las traducciones. El sistema que proponemos en esta tesis está basada en esta aproximación.

Figura 1.9: Sistema basado en estadística guiadas por reglas



1.4. Técnicas de la lingüística computacional aplicada a la MT

La lingüística computacional (campo multidisciplinar de la lingüística y la informática, también conocida como Procesamiento de Lenguaje Natural) intenta analizar/modelar de forma lógica el lenguaje natural.

Hay cinco niveles análisis lingüístico que se pueden abordar al procesar el lenguaje natural:

- Fonética/fonología (se ocupa de la exploración de las características del sonido)
- Morfología (se ocupa de la estructura interna de las palabras y el sistema de categorías gramaticales de las lenguas)
- Sintaxis (se ocupa de estudiar las relaciones entre las palabras de la frase)
- Semántica (se ocupa de entender/interpretar la frase)

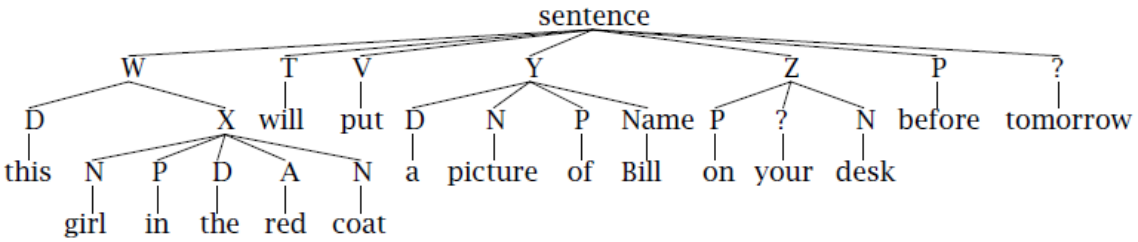
- Pragmática (se ocupa de las relaciones entre la oración y el mundo externo)

Haciendo un procesamiento parcial de los cinco niveles del análisis lingüístico, algunas aplicaciones de la lingüística computacional que servirían para la traducción automática serían:

- Analizadores sintácticos[1]

  Los analizadores sintácticos convierten el texto de entrada en otras estructuras (comúnmente árboles) (ver figura 1.10), que son más útiles para el posterior análisis y capturan la jerarquía implícita de la entrada.

Figura 1.10: Análisis sintáctico



- Etiquetadores morfológicos[2] o POS-taggers[3]

  Los etiquetadores morfológicos o gramaticales se encargan de asignar una etiqueta POS a cada una de las palabras de un texto su categoría gramatical (determinante, sustantivo, pronombre, verbo, adjetivo, adverbio, preposición, conjunción, interjección) (ver figura 1.11).

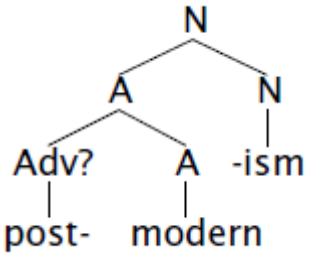
Figura 1.11: Etiquetado morfológico

- Analizadores morfológicos[4]

  Los analizadores morfológicos analizan la estructura interna de la palabras (unidad mínima de significado) (ver figura 1.12). Las palabras pueden estar compuestas por lexemas y morfemas. Los lexemas o raíces son la parte de la palabra que no varia, contiene su significado. Los morfemas es la parte de la palabra que varia y modifica el significado: sufijos, prefijos, morfemas de género, nombre, tiempo y persona.

[1] http://es.wikipedia.org/wiki/Analizador_sint%C3%A1ctico
[2] http://es.wikipedia.org/wiki/Etiquetado_gramatical
[3] http://es.wikipedia.org/wiki/Categor%C3%ADa_gramatical
[4] http://www.estudiantes.info/lengua/lexema_y_morfema.htm

Figura 1.12: Análisis morfológico



- Lematizadores[5]

  Los lematizadores, dada una forma flexionada (es decir, en plural, en femenino, conjugada, etc), tratan de hallar el lema correspondiente. El lema es la forma básica o palabra tal y como la encontraríamos en un diccionario o lexicón, que por convenio se acepta como representante de todas las formas flexionadas de una misma palabra.

- Segmentadores o tokenizadores

  Los tokenizadores se encargar de segmentar/partir en unidades indivisibles o tokens el texto de entrada. El lugar en el que se debe separar las palabras a menudo depende de cuál es la posibilidad que mantenga un sentido lógico tanto gramatical como contextual. En lenguas como el chino mandarín o el japonés, este proceso se vuelve bastante importante debido a que no tienen separaciones entre las palabras.

[5]http://es.wikipedia.org/wiki/Lematizaci %C3 %B3n

2 Lenguas distantes

El tema principal de esta tesis es la traducción entre lenguas "distantes". Cuando hablamos de lenguas "distantes" nos referimos a que entre la lengua destino y la lengua origen hay bastantes diferencias lingüísticas, las cuales pueden consistir en que sea distinto: el sistema de escritura, la estructura gramatical de las frases, la morfología de las palabras, los tiempos verbales, el uso de artículos...

Por lo que cuando queremos traducir de una lengua muy compleja lingüísticamente a otra más sencilla, la traducción es más sencilla de realizar, pero cuando queremos hacer el camino inverso, dicha traducción será más compleja.

En este capítulo veremos un resumen de las lenguas distantes que veremos en esta tesina, y las compararemos lingüísticamente con el inglés, ya que siempre será este el usado de lengua origen o lengua destino.

Aunque lo primero que haremos será ver una de las posibles clasificaciones que hay para catalogar las distintas lenguas del mundo, el cual se basa en el modo en el que funcionan morfológicamente (sistema que se emplea para clasificar las lenguas creado por los hermanos Friedrich y August von Schlegel):

- Lenguas analíticas o aislantes[1]

  Este tipo de lenguas, apenas hay derivación morfológica. El orden de las palabras es muy importante para dar a entender el significado y la relación sintáctica de las palabras. El radical permanece solitario y no se le modifica.

  Algunos ejemplos de lengua analítica son el chino, el malayo, el indonesio, el tailandés y el vietnamita.

- Lenguas sintéticas

  - Lenguas sintéticas aglutinantes[2]

    Estas lenguas tienen cierta complejidad morfológica, pero los morfemas (elementos estructurales) siempre se pueden separar claramente. Aunque el radical se modifique, permanecen inalterables en el sentido de que la modificación se realiza mediante afijos que se añaden al radical. Los afijos que se usan con más frecuencia son sufijos. Se añaden dependiendo de la función de la palabra. El orden de las palabras es algo menos importante que en las lenguas analíticas. Esto se debe a que los sufijos añaden información útil para averiguar el papel sintáctico de la palabra.

[1] http://es.wikipedia.org/wiki/Lengua_anal%C3%ADtica
[2] http://es.wikipedia.org/wiki/Lengua_aglutinante

Algunos ejemplos de lengua aglutinante son el coreano, el turco, el japonés, el finés, el swahili y el vasco.

- Lenguas sintéticas fusionantes o flexivas[3]

  Las lenguas fusionantes son las lenguas de mayor complejidad morfológica de los tres tipos. A menudo, no se puede separar los morfemas del lexema o radical. En ocasiones, el lexema no se puede diferenciar de los afijos. El orden de las palabras no es importante en absoluto, ya que la gran parte o la totalidad de la información de la estructura sintáctica se revela mediante la morfología de las palabras. Es decir, el orden de los sintagmas no altera el significado de la oración.

  Algunos ejemplos de lengua sintética son el latín, el español, el árabe, el polaco y el alemán.

A continuación veremos en más detalle las lenguas que contendrán los corpus de entrenamiento usados en los experimentos de esta tesis.

2.1. Japonés

El japonés[4] es una lengua de estructura aglutinante que mezcla tres sistemas de ortografía: hiragana, katakana y kanji. Tiene poca morfología, las raíces léxicas aparecen invariables y tienen una significación fija y apta para existir separadamente. El japonés es casi exclusivamente sufijante, con muy pocos prefijos, por lo que los únicos procesos para la formación de palabras son la composición y la derivación mediante sufijos.

Su estructura gramatical es sujeto-objeto-verbo (SOV) y usa posposiciones en lugar de preposiciones.

Usualmente no utiliza espacios entre las palabras de la frases. Para diferenciar cuando se ha terminado una palabra o sintagma, hay partículas especiales para tal efecto (indicadora de sujeto, tema, lugar, complemento indirecto...).

No existen artículos, ni género gramatical (masculino/femenino), ni número obligatorio y el caso es indicado por clíticos.

Los verbos, que son básicamente impersonales, solo tienen dos conjugaciones, el pasado y el presente. El futuro se deduce a partir del uso de otras partículas.

[3] http://es.wikipedia.org/wiki/Lengua_fusionante
[4] http://es.m.wikipedia.org/wiki/Idioma_japon%C3%A9s#section_3

2.2. Chino

El chino[5] es una lengua aislante (lo contrario de aglutinante), donde las diversas estructuras sintácticas como las de sujeto, objeto, diversos complementos, etc. vienen dadas por su posición en la frase. Normalmente el sujeto ocupa el primer lugar y cada sintagma tiene un lugar prefijado en la oración.

La escritura del chino se caracteriza por usar caracteres hàn, que en español se denominan frecuentemente sinogramas.

El chino tiene poca morfología, las raíces léxicas aparecen invariables. No se modifican en su forma sino que es su posición y la existencia de partículas diversas las que precisan su significado y función.

Predomina el orden gramatical sujeto-verbo-objeto (SVO), aunque la mayoría de sintagmas colocan el núcleo en posición final.

No existen artículos, ni género gramatical (masculino/femenino), ni número (excepto por algunas formas de plural marginales en los pronombres personales).

Al igual que el japonés, usualmente no utiliza espacios entre las palabras de la frases.

Los verbos no tienen conjugaciones. El chino usualmente no marca el tiempo gramatical sobre el verbo sino dicha información está en los adverbios de tiempo ('ayer', 'hoy', 'mañana', etc).

Desde el punto de vista de las categorías gramaticales destaca la existencia de coverbos y de clasificadores. Los clasificadores son obligatorios entre un determinante y el nombre al que rige, y están relacionados generalmente con la forma de objeto al que se refiere el nombre o el campo semántico del nombre.

2.3. Inglés

El inglés[6] antiguo (o anglosajón) era originalmente una lengua de estructura fusionante (como el alemán), que debido a influencias de otras lenguas (como latín, nórdico antiguo y normando), se hizo una lengua algo analítica pero también con algunos rasgos aglutinantes.

El inglés moderno tiene muchos rasgos típicos de las lenguas europeas, principalmente fusionantes. El nombre presenta diferencia entre singular y plural. En inglés moderno, a diferencia del anglosajón, el nombre no hace distinciones de género o caso. Las diferencias de caso se restringen en inglés moderno al pronombre.

En el sistema verbal, el inglés moderno, al igual que el alemán y las lenguas romances, ha sufrido una evolución similar. Se han creado "formas compuestas de perfecto" para

[5] http://es.m.wikipedia.org/wiki/Idioma_chino#section_2
[6] http://es.wikipedia.org/wiki/Idioma_ingl%C3%A9s

expresar el aspecto perfecto y "formas perifrásticas" con el verbo ser para expresar el aspecto progresivo o continuo. Otra similaridad es el desarrollo de formas de futuro a partir de verbos auxiliares. Una diferencia importante entre el inglés y otras lenguas germánicas y romances es el debilitamiento del modo subjuntivo. Igualmente el inglés, al igual que el alemán, el holandés o las lenguas románicas, ha creado artículos definidos genuinos a partir de formas demostrativas.

En inglés hay cuatro tiempos fundamentales: presente, pasado y futuro y condicional. Estos tres tiempos se combinan con tres aspectos (imperfecto, continuo, perfecto), las combinaciones de aspecto posible son cuatro ([-perf][-cont],[-perf][-cont], [-perf][-cont] y [-perf][-cont]). Las combinaciones de tiempo y aspecto anteriores dan lugar a un número importante de tiempos verbales.

En el idioma inglés, el orden básico de las palabras en la oración es del tipo sujeto-verbo-objeto (SVO).

2.4. Polaco

El polaco[7] es una lengua eslava de estructura fusionante (como el alemán).

El polaco tiene u un sistema de cinco géneros: neutro, femenino y tres géneros masculinos (personal, animado e inanimado). Existen 7 casos, como los demás idiomas eslavos (excepto macedonio y búlgaro) que son nominativo, acusativo, genitivo, dativo, instrumental, locativo y vocativo y 2 números.

Los sustantivos, adjetivos y verbos son flexivos, y tanto la declinación de los sustantivos como la conjugación son difíciles de aprender debido a que tienen muchas reglas y excepciones.

Los verbos tienen 4 conjugaciones diferentes y suelen venir en parejas, en que uno es imperfectivo y el otro perfectivo (que suele ser el imperfectivo con un prefijo), pero también hay una gran cantidad de verbos perfectivos con diferentes prefijos para un solo verbo imperfectivo.

En el idioma polaco, el orden básico de las palabras en la oración es del tipo sujeto-verbo-objeto (SVO), si bien, dado que es una fusionante, tal orden no es fundamental. La conjugación del verbo permite la omisión del sujeto y, de igual modo, el complemento también puede desaparecer si es evidente por el contexto.

[7] https://es.wikipedia.org/wiki/Idioma_polaco

3 Aplicaciones de MT

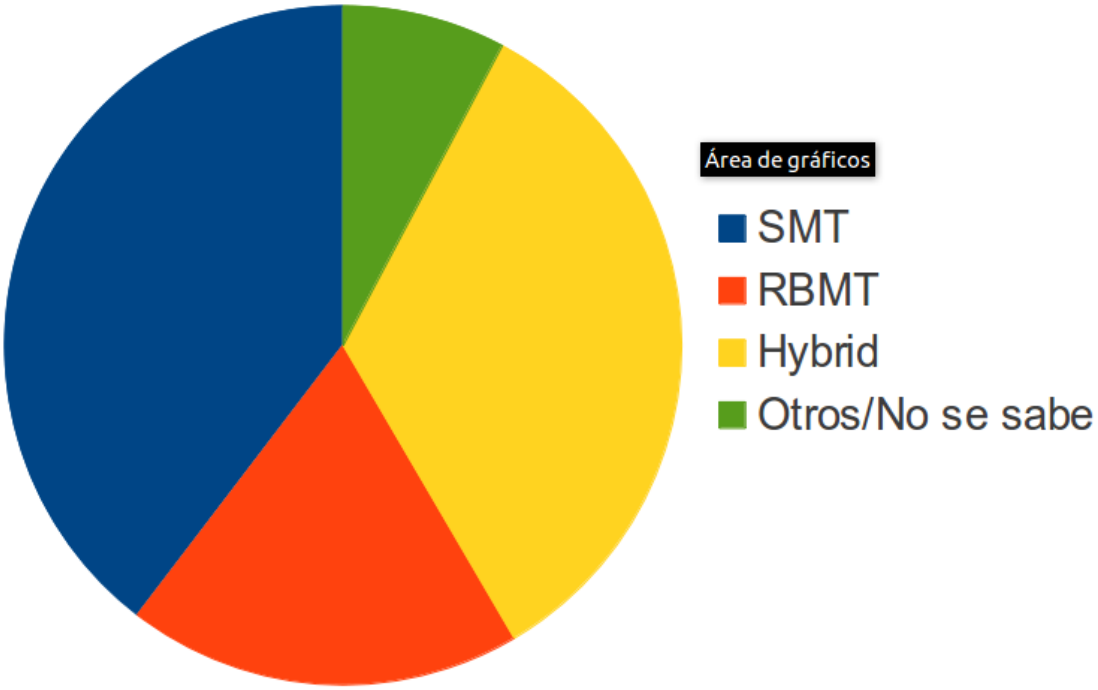
En el mundo profesional de la traducción hay mucho escepticismo sobre la traducción automática. Y aunque no es adecuada para reemplazar a seres humanos en aplicaciones donde la calidad es realmente importante, ya hay profesionales que han empezado a entender que la traducción automática es imperativa para facilitar/abaratar determinados casos:

- Contenido altamente repetitivo donde las ganancias de productividad con MT pueden exceder lo que es posible con solo TMs
- Contenido que no podría ser traducido de otro modo
- Contenido que no puede permitirse traducción humana
- Contenido de alta valor que cambia cada hora y cada día
- Contenido de conocimiento que facilita y mejora la difusión de conocimiento crítico
- Contenido que es creado para mejorar y acelerar la comunicación con clientes globales que prefieren un modelo de auto-servicio
- Contenido que no necesita ser perfecto sino simplemente entendible

Para algunos profesionales hay un debate sobre si usar sistemas basados en reglas o basados en estadísticas, aunque en la actualidad se ha vuelto de moda el enfoque híbrido.

En la figura 3.1 podemos ver cual es tipo de sistema usado entre los miembros (principalmente agencias profesionales de traducción) de la asociación TAUS (extraído del reporte de junio del 2013 que TAUS que hizo entre sus miembros).

Figura 3.1: Tipo sistema usado por miembros TAUS



A continuación veremos los sistemas más relevantes del panorama de la traducción automática, o en el que más sistemas están basados. Los clasificaremos en dos categorías, los "libres" con código fuente abierto, y los "propietarios".

3.1. Sistemas de MT libres

3.1.1. Moses

Moses [KHB+07] es un kit de herramientas de código abierto para la traducción automática estadística que permite el entrenamiento automático de modelos de traducción para cualquier par de lenguas que vengan alineadas (corpus paralelo). Está distribuido bajo la licencia LGPL, tanto bajo Windows como Linux.

El decodificador del Moses (el cual es una de las principales partes de la herramienta) fue principalmente desarrollado por Hieu Hoang y Philipp Koehn en la Universidad de Edinburgo y posteriormente desarrollado bajo la financiación del proyecto EuroMatrix[1] y GALE[2].

[1]http://www.euromatrix.net/
[2]http://projects.ldc.upenn.edu/gale/

Moses puede crear dos tipos distintos de modelos de traducción: modelos basados en frases ("phrase-based") y modelos basados en árboles o modelos sintácticos ("tree-based").

Además, Moses puede usar modelos de traducción factoriales (con información lingüística), tanto en los modelos basados en frases, como en los basados en árboles.

Moses soporta únicamente la traducción entre texto plano, no lleva ningún módulo de conversión/parseado de ficheros de uso cotidiano (DOCX de Microsoft Office, ODT de OpenDocument...) o ficheros bilingües del ámbito de la traducción profesional (TTX de SDL Trados, XLIFF, TMX...).

Actualmente Moses es uno de los sistemas de traducción automática más usados tanto a nivel académico como a nivel profesional; tan solo que para su uso profesional requiere de interfaces de uso (web, API, aplicación Windows...) y de parseadores de ficheros. El sistema en el que se basará la tesis, PangeaMT, es un sistema basado en Moses, donde le añade todo lo necesario para su uso profesional en la industria de la traducción automática.

3.1.2. Apertium

Apertium [AOCBFZ+07] es una plataforma de código abierto para la traducción automática basada en reglas que usa un sistema de transferencia superficial ("shallow-transfer"). El proyecto fue iniciado en el Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Alicante y está distribuido bajo la licencia GPL para sistemas Linux.

Para crear un motor en este sistema, tan solo bastaría con desarrollar datos lingüísticos (diccionarios, reglas) en lenguaje específico XML.

Apertium usa transductores de estado finito para todas sus transformaciones léxicas, y modelos ocultos de Markov para el etiquetado POS o para la desambigüación de la categoría de las palabras.

Originalmente fue diseñado para traducir entre lenguas cercanas, aunque recientemente fue expandido para tratar otras pares de lenguas más divergentes.

Apertium soporta la traducción directa de distintos tipos de ficheros, tales como página web HTML, ficheros OpenDocument (ODT); Microsoft Word, Excel y PowerPoint (DOCX, XLSX y PPTX); y ficheros de texto plano TXT, texto enriquecido (RTF) y ficheros con formato LATEX.

3.1.3. Joshua

Joshua [LCBD+09] es un kit de herramientas de código abierto para la traducción automática estadística basada en modelos sintácticos que usa gramáticas sin

contextos estocásticas. Ha sido desarrollado por el grupo de investigación de Chris Callison-Burch de la universidad Johns Hopkins. Está distribuido bajo la licencia LGPL para sistemas Linux.

Joshua usa técnicas de computación paralela y distribuida para la escalabilidad, y utiliza un lenguaje de matriz sufijo fuente para extraer solo aquellas reglas que se utilizarán en realidad para traducir un conjunto de test en particular. El resultado es un conjunto de reglas más pequeño que extraer todas las reglas del conjunto de test.

Joshua, al igual que comentamos en Moses, tan solo puede trabajar sobre texto plano, por lo que para su uso cotidiano o profesional, necesitaría de una interfaz de uso (web, API, aplicación Windows...) y de parseadores de ficheros.

3.1.4. OmegaT

OmegaT un software de traducción asistida por ordenador basado en memorias de traducción desarrollado por Keith Godfrey en 2000, y mantenido actualmente por el grupo liderado por Didier Briel. Está distribuido bajo licencia GPL para sistemas Windows, Linux y Mac.

OmegaT comparte muchas funcionalidades con las principales herramientas propietarias de traducción asistida (CAT) del mercado (SDL TRADOS, MEMOQ y SWORDFISH que veremos a continuación).

OmegaT soporta la traducción directa de numerosos tipos distintos de ficheros, tales como lenguajes de marcado y formatos de etiquetado como HTML, XML, LATEX, TEX, XLIFF, SDLXLIFF (formato nativo del Studio que está basado en el XLIFF), ficheros OpenDocument (ODT, ODS, ODP...); ficheros de código fuente (de Java, Android o ResX por ejemplo); Microsoft Word, Excel y PowerPoint (DOC, DOCX, XLS, XLSX, PPT y PPTX); formato para imágenes SVG; y el formato para subtítulos SubRip (SRT).

OmegaT permite el uso simultáneo de múltiples memorias de traducción y glosarios. El sistema de memorias de traducción de OmegaT se clasifica en tres memorias TMX: un TMX nativo de OmegaT, un TMX de nivel 1 y un TMX de nivel 2. En el TMX de nivel 1 solo está almacenada la información textual, y en el TMX de nivel 2 está almacenada, además de información textual, también las etiquetas internas del propio segmento; ambos TMX pueden ser usados con otras herramientas CAT que soporten TMX (tales como TRADOS o SDLX). A su vez, OmegaT también permite importar ficheros TMX 1.4b para su posterior uso.

Para los glosarios, OmegaT principalmente usa ficheros TXT tabulados con codificación UTF8. La estructura del glosario es de tres columnas: la primera columna contiene la palabra de la lengua origen, la segunda columna contiene las palabras de la lengua destino, y la tercera columna (que es opcional) contiene comentarios.

La traducción automática también es soportada desde sistemas como Apertium y Google Translate.

3.2. Sistemas de MT propietarios

3.2.1. Systran

Systran es un software comercial de traducción automática basado en reglas. Fue creado por Systran Ltd en 1968. Está distribuido para sistemas Windows, Linux y Solaris; o a través de su aplicación web.

Desde la versión 7 de 2010, implementa un híbrido entre traducción automática basada en reglas y estadística, el cual es el primero de esta clase en el mercado.

Soporta los siguientes tipos de formatos de Microsoft (DOCX, DOC, XLSX, XLS, PPTX, PPT y MSG); Open Office Documents (ODT, ODS y ODP), Adobe PDF, documentos de etiquetado (HTML Y TMX), ficheros de texto plano (TXT) y enriquecido (RTF).

Para la integración con otros entornos, ofrece una interfaz API con soporte para SOAP.

3.2.2. Language Weaver

Language Weaver es un software comercial de traducción automática estadístico. Fue creado por Kevin Knight y Daniel Marcu de la Universidad de California del Sur en 2002. Está distribuido para entornos Windows, y también puede ser usado a través de su aplicación web.

Language Weaver usa los modelos de traducción basados en frases, aunque también está trabajando en los modelos sintácticos para determinados pares de lenguas.

Soporta los siguientes tipos de formatos: de Microsoft, Adobe PDF, documentos de etiquetado (HTML, TMX Y XLIFF) y Open Office Writer (ODF).

En Julio del 2010, Language Weaver fue adquirido por SDL, compañía de software de traducción asistida.

Para la integración con otros entornos, ofrece una interfaz API con soporte para SOAP.

3.2.3. Google Translate

Google Translate es un software de traducción automática estadística gratuita de Google Inc. Puede ser usado a través de su aplicación web, o en plataformas Android y iOs donde es distribuido.

Antes de Octubre de 2007, Google Translate usaba Systran para las traducciones (excepto para las lenguas árabe, chino y ruso), entonces pasaron al sistema basado en estadística propuesto por Franz Josef Och.

Para la integración en otros entornos, ofrece una interfaz API AJAX, pero en Diciembre del 2011, debido a la gran carga económica debido a su gran uso, el uso del API pasa a ser de pago.

Google a menudo no traduce de una lengua a otra directamente (L1 $\rightarrow$ L2), sino que traduce primero a Inglés y entonces traduce a la lengua destino (L1 $\rightarrow$ EN $\rightarrow$ L2).

Cuando Google Translate genera una traducción, busca patrones en cientos de millones de documentos web para ayudar a decidirse en la mejor traducción, y también proponer traducciones alternativas, por ejemplo para términos técnicos, que el usuario puede cambiar para que sea incluido en futuras actualizaciones del proceso de traducción.

3.2.4. Asia Online

Asia Online es un software comercial de traducción automática estadística de la firma Asia Online Pte Ltd que fue fundada en 2007 entre otros por Philip Koehn. Está distribuido para las plataformas Windows y Linux, y también puede ser usado por su interfaz web.

Su sistema de traducción está basado en el kit de Moses.

Soporta los siguientes tipos de formatos: Microsoft (DOCX, DOC, XLSX, XLS, PPTX y PPT), documentos de marcado y formatos de etiquetado como HTML, TMX, XML, XLIFF, SDLXLIFF; y ficheros de texto plano (TXT) y enriquecido (RTF).

Para la integración con otros entornos, ofrece un API de servicios web REST con soporte para SOAP.

3.2.5. Lucy

Lucy [AT03] es un software comercial de traducción automática basado en reglas. Fué creado por la firma Lucy Software and Services GmbH y está distribuido para la plataforma Windows, y próximamente también en Linux.

Lucy consiste en dos principales componentes, el módulo de manejo de texto y el núcleo de MT. El núcleo de MT, basado en reglas (RBMT), tiene en sus raíces el sistema de transferencia METAL; y solo puede manejar ficheros de texto plano. Es el módulo de manejo de ficheros el encargado de que documentos que no están en texto plano, que puedan ser traducidos por el núcleo. Dicho módulo puede manejar

diversos formatos como Microsoft Word (DOC Y RTF), Microsoft Outlook, HTML y PDF.

Permite la creación, importación y exportación de memorias de traducción en formato TMX y TXT.

Para la integración en otros entornos, ofrece una interfaz API: Java API (y Web API), COM API, Corba API y SOAP (Servicio Web) API.

3.2.6. ProMT

ProMT es un software comercial de traducción automática basada en reglas desarrollado por ProMT Ltd. Se distribuye para entornos Windows; también puede usarse a través de su aplicación web.

Aunque originalmente ProMT es un sistema de transferencia, en los dos últimos años han estado trabajando en técnicas estadísticas para crear un sistema de traducción híbrido.

En los diccionarios de transferencia, las inflexiones de cada lengua son almacenadas en forma de árbol para así reducir información redundante y repetida, y de este modo, ya no es necesario tener distintas entradas de diccionario para las todas las formas relacionadas porque ahora son almacenadas en una única entrada.

Desde la versión 7.0, la aproximación multidimensional es usada en la arquitectura del diccionario, a nivel de la estructura de la descripción de la palabra. Cada palabra o expresión tiene al menos una traducción activa, y además puede tener múltiples traducciones inactivas. Las traducciones activas son usadas directamente durante el proceso de traducción, mientras que las variantes de la traducción que están inactivas pueden ser buscadas para aportar información adicional del significado de la palabra. Cualquier variante de traducción que no está activa, puede ser convertida en activa y viceversa, permitiendo de este modo traducciones ilimitadas.

Los algoritmos del sistema de traducción están basados en la aproximación jerárquica que provee una la subdivisión de los procesos de traducción dentro de los procesos interconectados para las distintas unidades de análisis lingüístico. El sistema diferencia los siguientes niveles: nivel de unidad léxica, nivel de grupo, nivel de oración simple y nivel oración compuesta. Todos estos procesos están interconectados e interactúan jerárquicamente de acuerdo a la jerarquía de la unidad de texto, y además intercambian atributos sintetizados y heredados.

ProMT provee a los clientes con el Dictionary Editor que permite a los usuario ver, crear y editar información en sus diccionarios. También provee el Terminology Manager (ProMT TerM) que provee a los usuarios una herramienta automatizada de extracción, minería y gestión terminológica.

Soporta los siguientes tipos de formatos: Microsoft (DOCX, DOC, XLSX, XLS, PPTX, PPT y MSG), Adobe PDF, documentos de etiquetado (HTML Y XML) y Open Office Writer (ODT).

Para la integración con otros entornos, ofrece una interfaz API con soporte para SOAP.

3.2.7. SDL Trados

SDL Trados es una suite de software de traducción asistida por ordenador (o CAT) basado en memorias de traducción, desarrollada originalmente por la empresa alemana Trados GmbH, y actualmente disponible desde SDL Internacional, un proveedor de gestión de traducciones, gestión de contenido y servicios de lenguaje. Provee gestión de memorias de traducción y terminología. Se distribuye para sistemas Windows.

Desde finales de los años 90, es la herramienta CAT más usada del mercado de las traducciones.

La versión Studio 2011 soporta más de 70 tipos distintos de ficheros, incluyendo lenguajes de marcado y formatos de etiquetado como SGML, XML, HTML, XLIFF, SDLXLIFF (formato nativo del Studio que está basado en el XLIFF), ficheros OpenDocument (ODT, ODS, ODP...); ficheros de código fuente (de Java o .NET por ejemplo); Microsoft Word, Excel y PowerPoint (DOCX, XLSX y PPTX versión 2000, 2003, 2007 y 2010); y algunos formatos de Adobe, como PDF, FrameMaker, InDesign, e InCopy.

El formato para los archivos de traducción (archivo intermedio donde se realizarán las traducciones) en la versión 2011 es SDLXLIFF, y en la versión 2007 es TTX.

El formato usado para las memorias de traducción (TM) en la versión 2011 es SDLTM, donde se almacena una base de datos con todas las unidades de traducción. Además también almacena información estructural y de contexto para así relacionar todos los diferentes segmentos a su posición dentro del documento de partida usado. Esto permite a la suite seleccionar el segmento más relevante de la memoria de traducción.

En la versión anterior del programa, Trados 2007, se creaba una red neural de ficheros para permitir la capacidad de búsqueda difusa. La memoria de traducción consistía de 5 ficheros, TMW como el el fichero principal y MDF, MTF, MWF e IIX como ficheros de redes neuronales.

En la versión 2011 permite importar TMs en formato TMX (aunque no exportar); mientras que la versión 2007 permite tanto importar como exportar en formato TMX.

Mientras que la versión 2007 solo permitía el uso de una única TM, la versión 2011 si que permite el uso de múltiples TMs.

Trados también puede trabajar con memorias de traducción remotas, al estilo cliente-servidor, permitiendo que desde distintas instancias del Trados acceder a la misma

TM, y por lo tanto segmentos traducidos por un traductor son conocidos por el resto de traductores.

La terminología es gestionada desde la aplicación MultiTerm. Los glosarios terminológicos pueden ser bilingües o multilingües y también pueden ser remotas.

Trados ha integrado la traducción automática y posedición (proceso de mejorar una traducción generada por una maquina) dentro de su flujo de traducción. Con la apropiada configuración, Trados puede insertar la traducción realizada por un motor de MT en un segmento a traducir (unidad de traducción o TU) si no se ha encontrado ningún resultado o ninguno suficientemente bueno en la TM. El traductor puede entonces posteditar dicha traducción.

La integración de la traducción automática puede hacerse de dos modos distintos, vía importando un TMX donde se hayan traducido unidades de traducción mediante MT (realizando las traducciones de MT al principio, antes de empezar a posteditar); o vía el uso del API abierto y la arquitectura de plugins de SDL OpenExchange (realizando las traducciones de MT una a una tal como se soliciten).

3.2.8. Swordfish

Swordfish es un software de traducción asistida por ordenador basado en memorias de traducción. Está distribuido por MaxPrograms con licencia comercial en Windows, Mac y Linux.

El formato para los archivos de traducción es XLIFF. Puede extraer el texto desde una gran variedad de formatos tales como TXT, RTF, Microsoft Word, Excel y PowerPoint (DOCX, XLSX y PPTX versión 2007 y 2010); ficheros OpenDocument (ODT, ODS, ODP...); ficheros de código fuente (de Java o ResX por ejemplo); algunos formatos de Adobe, como FrameMaker (MIF y XML) e InDesign (INX y IDML); compatibilidad con formatos de otras herramientas CAT tales como SDL TRADOS 2007 (TTX) y WordFast (TXML); y algunos formatos de aplicaciones gráficas, como Photoshop (SVG), Illustrator (SVG) y CorelDraw (SVG).

Almacena memorias de traducción en la memoria interna y puede exportar e importar al formato TMX.

También permite una versión servidor de las TMs, RemoteTM, que permite que distintas instancias del Swordfish acceder a la misma TM, y por lo tanto segmentos traducidos por un traductor son conocidos por el resto de traductores.

3.2.9. MemoQ

MemoQ es un software de traducción asistida por ordenador basado en memorias de traducción. Está distribuido por Kilgray con licencia comercial (y también en versión con licencia gratuita) en sistemas Windows.

Puede extraer el texto desde una gran variedad de formatos incluyendo lenguajes de marcado y formatos de etiquetado como XML, HTML, TMX, XLIFF; ficheros OpenDocument (ODT y ODF); ficheros de código fuente (de Java o ResX); Microsoft Word, Excel y PowerPoint (DOC, DOCX, XLS, XLSX, PPT y PPTX versión 2000, 2003, 2007 y 2010); Microsoft Visio (VDX); compatibilidad con formatos de otras herramientas CAT tales como SDL TRADOS 2007 (TTX), SDL STUDIO 2011 (SDLXLIFF) y WordFast (TXML); y algunos formatos de Adobe, como PDF, FrameMaker e InDesign.

Almacena memorias de traducción en la memoria interna y puede exportar e importar al formato TMX. Permite el uso de múltiples TMs.

También permite una versión servidor de las TMs y las bases terminológicas, que permite que distintas instancias del MemoQ acceder a la misma TM y base terminológica.

La integración de la traducción automática puede hacerse de dos modos distintos, vía importando un TMX donde se hayan traducido unidades de traducción mediante MT (realizando las traducciones de MT al principio, antes de empezar a posteditar); o vía el uso del interfaz API compatible con SOAP.

3.2.10. PangeaMT

PangeaMT [YHL+10] es un software comercial de traducción automática estadística basado en Moses [KHB+07], y ha sido creado por Pangeanic/B.I-Europa con la colaboración del ITI de la Universidad Politécnica de Valencia, y será donde todas las propuestas planteadas en esta tesis serán usadas. Es un sistema multiplataforma que es usado a través de su aplicación web.

Pangeanic es un proveedor español de servicios de lenguaje (LSP), que ha estado realizando traducciones de documentación técnica durante más de 20 años. Ha estado realizando trabajos en sectores como la electrónica, automoción, generación de energía, maquinaría, software, medicina, biotecnología y muchas otras industrias.

En el año 2005, Pangeanic se convierte en uno de los miembros fundadores de la asociación de datos TAUS (TAUS Data Association, TDA[3]). Los programas de compartición de datos solo benefician a los miembros actuales o futuros de TAUS. Cuanto más información haya disponible, mejores procesos y tecnologías del lenguaje (HLT) pueden ser desarrollados con representatividad del dominio, velocidad y rentabilidad. Los datos compartidos también servirán de sobremanera como datos de entrenamiento para la creación de motores de traducción automática.

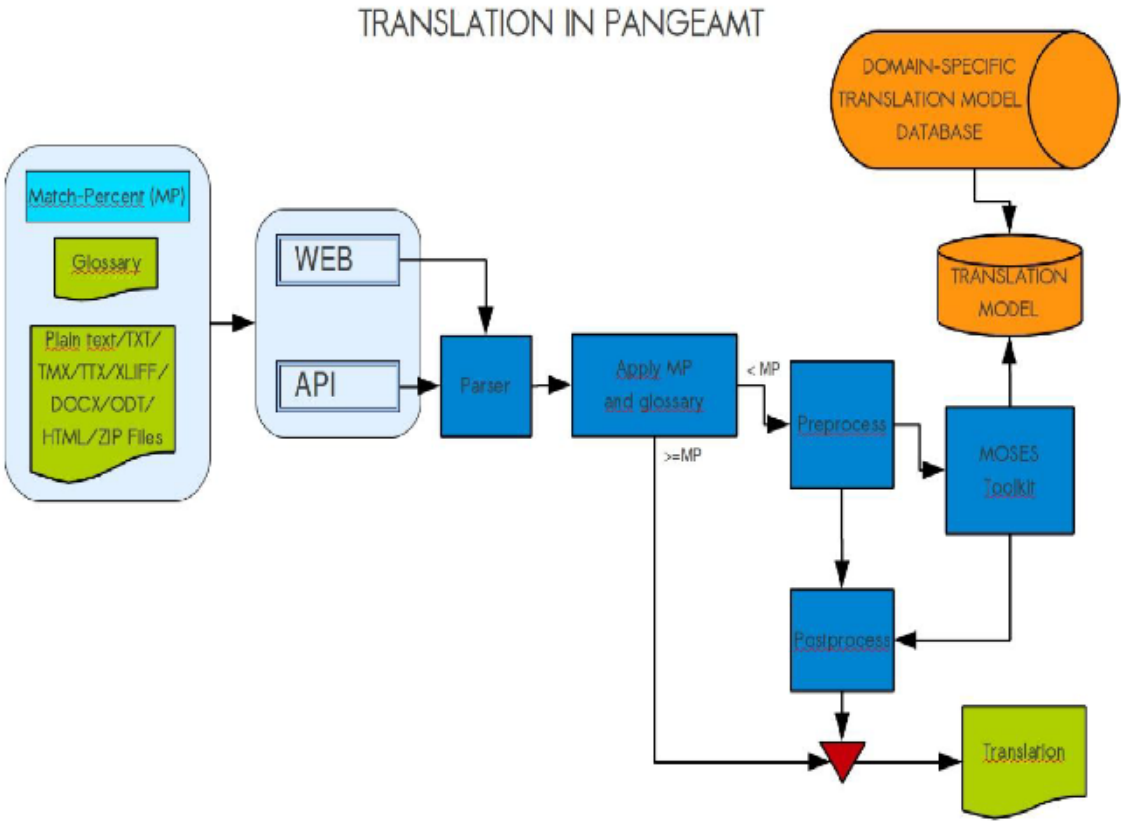
Después de probar y evaluar diversos sistemas comerciales de MT, Pangeanic decide a finales de 2008 invertir en crear un sistema de traducción automática estadística propio basado en MOSES, llamado Pangematic. Originalmente fue concebido para

[3]http://www.tausdata.org/

uso interno, y desde entonces a permanecido la motivación de más desarrollo y ha hecho que, Pangematic bautizado como PangeaMT a principios de 2010, se convirtiera en un verdadero sistema comercial MT.

PangeaMT está constituido por varios módulos de pre y posproceso encima de Moses, al igual que interfaces web y procedimientos de reentrenamiento (ver figura 3.2 y 3.3), en orden de desarrollar una alternativa de MT orientada a la industria.

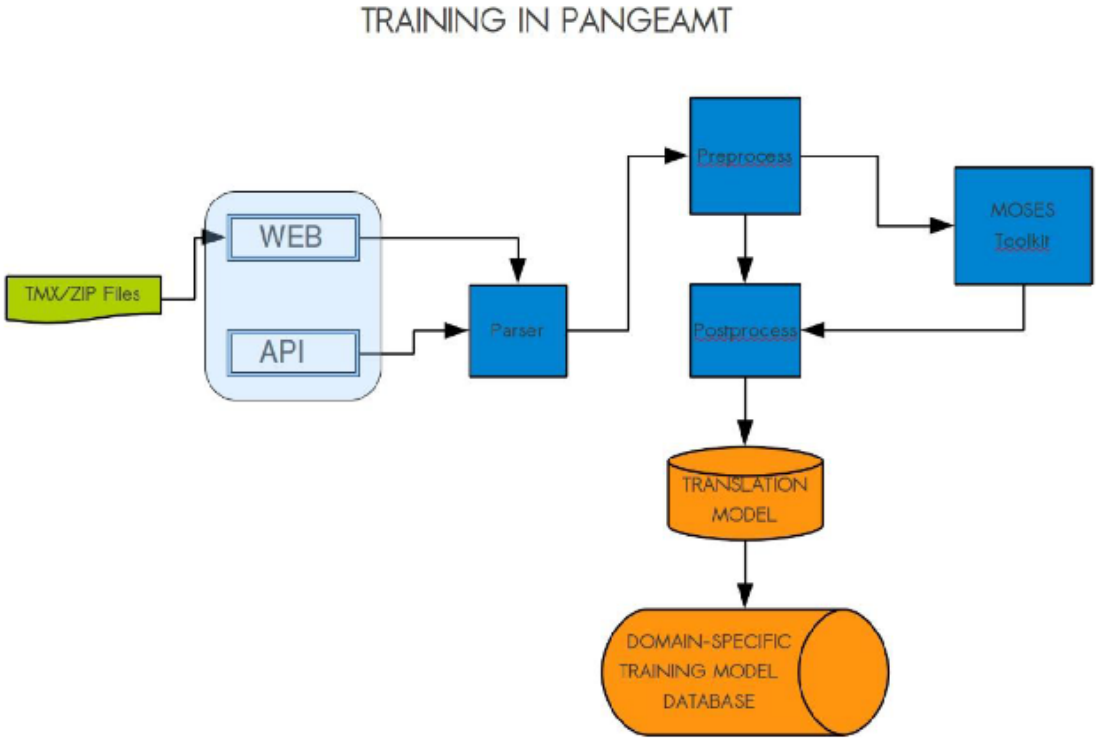
Figura 3.2: Diagrama traducción en PangeaMT



Con el fin de fomentar la accesibilidad a las soluciones de MT customizadas y personalizables, así como una mayor independencia de los proveedores de MT para la experimentación y reentrenamiento con motores de MT, Pangeanic lanza oficialmente la solución PangeaMT SMT DIY ("Do It Yourself") [YHH+12] para E-FIGS (inglés hacia: francés, italiano, alemán y español) en la Localization World Conference de Barcelona en junio de 2011. Su principal baza consiste en el (re)entrenamiento automático del motor, donde el usuario sube directamente los datos de entrenamiento -en formato TMX 1.4b- a carpetas de un FTP, y el sistema crea un motor por cada carpeta, y en caso de que ya estuviera creado lo reentrena si ha habido un incremento en la cantidad de datos.

En versiones posteriores de PangeaMT, el sistema de subida de datos de entrenamiento y la creación de motores cambia, y todo se hace directamente vía web (y vía

Figura 3.3: Diagrama entrenamiento en PangeaMT



API). Las datos de entrenamiento en vez de estar clasificados en carpetas, están clasificados en "dominios" a través de una BBDD; y el entrenamiento/reentrenamiento de los motores puede ser lanzado directamente "a mano" por el usuario o puede ser programado para que se realice cada cierto tiempo.

La interfaz web además de permitir gestionar los archivos de traducción y los motores, también permite la gestión de usuarios del sistema, el seguimiento de la actividades de traducción y la traducción directa de texto plano y de ficheros.

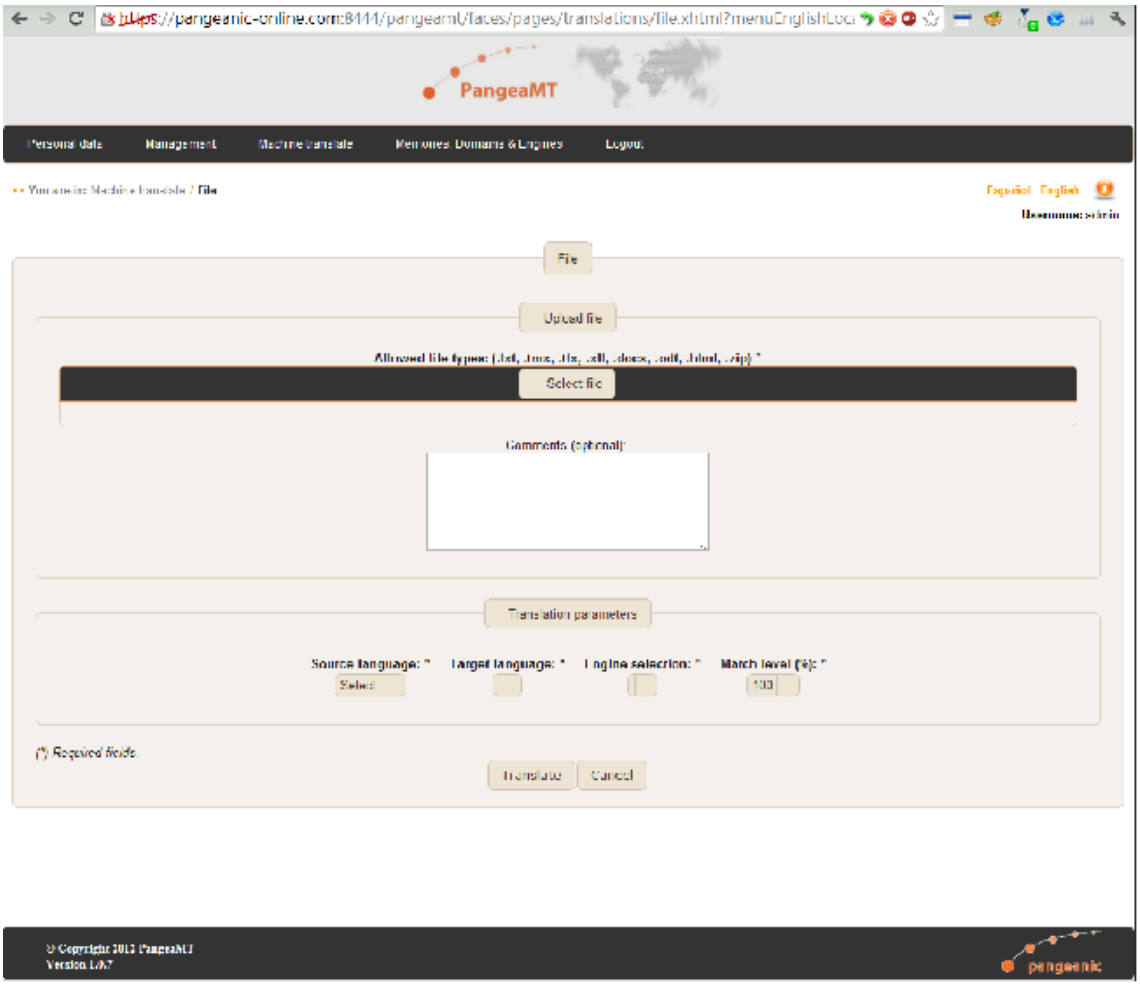
Los ficheros compatibles para traducir son los ficheros con lenguajes de marcado y formatos de etiquetado como HTML, TMX y XLIFF; ficheros OpenDocument (ODT); Microsoft Word (DOCX); y con el formato de SDL TRADOS 2007 (TTX).

Otra característica útil, es que cuando solicitamos la traducción de un fichero (ver figura 3.4), podemos hacer uso de glosarios terminológicos (pasados mediante un fichero TXT tabulado con dos columnas, término origen y término salida), que permiten durante la traducción del fichero, que determinada terminología sea respetada y traducida de la manera indicada en el fichero de glosario, lo cual permite también la creación de términos DNT ("Do Not Translate"), algo muy necesario en la industria de la traducción.

Para la integración con otros entornos, PangeMT ofrece un API de servicios web REST con soporte para SOAP.

A finales del 2010, Pangeanic y Toshiba deciden hacer pruebas de hibridación en-

Figura 3.4: Interfaz web PangeaMT



tre sus dos tecnologías de traducción automática, PangeaMT (sistema estadístico) y The Honyaku (sistema basado en reglas). La hibridación, inicialmente para los pares EN<->JP, era la meta de la colaboración desde el primer día: como combinar el poder de la estadística con la potencial ayuda de las reglas lingüísticas, y así determinar si esta combinación funcionaba bien y justificaba el aunamiento de esfuerzos. Las reglas lingüísticas de Toshiba consistían básicamente en el reordenamiento -durante el preproceso- del inglés para asemejarlo al japonés para así facilitar la traducción al sistema estadístico PangeaMT (dichas reglas lingüísticas son parte de las propuestas tratadas en esta tesina y serán tratadas en más profundidad posteriormente en la Sección 4.2).

3.3. Integración de herramientas CAT y MT

Cuando se trabaja de manera profesional en la traducción, las herramientas más usadas para traducir son las herramientas de traducción asistida (herramientas CAT, donde algunos ejemplos serían OmegaT, SDL Trados o MemoQ) que hacen uso de memorias de traducción (TM), donde se muestra al traductor traducciones anteriores parecidas a la que se está traduciendo actualmente.

Pero en los años 90, un cliente de Systran (sistema de traducción automática basado en reglas), pensó en la integración de su herramienta CAT (Trados) y el propio Systran; lo cual supuso un gran esfuerzo para que sucediera.

Este tipo de integraciones, lo que hacen es que, desde la herramienta CAT, además de mostrar las traducciones más similares de la TM, también muestra como alternativa la salida de un motor de traducción automática.

A día de hoy, estas integraciones de sistemas de MT (basados en corpus o en reglas) en herramientas CAT, es uno de los modos más utilizados para traducción en el mundo profesional. Dicho deseo de integración entre ambas tecnologías es el que ha propiciado el desarrollo de APIs entre los principales aplicaciones de MT para su integración en herramientas CAT.

4 Propuestas de hibridación y optimización

En este capítulo veremos distintas propuestas para tratar de solventar los problemas lingüísticos de algunas de las lenguas analizadas en el Capítulo 2. También veremos que, incluso en corpus producidos en el ámbito de la industria de la traducción, se necesita aplicar un proceso de exclusión/detección de valores atípicos (o sucios) ya que todos estos afectaran en la creación de motores de traducción estadística.

Uno de los problemas lingüísticos que comentamos en el Capítulo 2, y el tema principal de esta tesis, es que en lenguas como el japonés, su estructura gramatical (sujeto-objeto-verbo) difiera de la estructura del inglés o el español (sujeto-verbo-objeto). La propuesta consiste en hacer uso del sistema de reordenación del inglés que hace uso Toshiba en su sistema basado en reglas "The Honyaku", para así usarlo dentro del sistema estadístico PangeaMT (basado en Moses) [YE11], y así no depender del reordenamiento estadístico, como ha estado ocurriendo mayoritariamente hasta el momento en el que se estuvieron haciendo los experimentos.

Otro de los problemas que observamos era que lenguas como el japonés o el chino no utilizan espacios entre las palabras, por lo que para poder realizar los experimentos nos veremos obligados a usar analizadores morfológicos que permitan segmentar dicho tipo de lenguas. Para ello repasaremos las distintas herramientas que hemos encontrado que nos permiten realizar dicha segmentación/tokenización. Las herramientas en cuestión son el Mecab y el KyTea para el japonés, y el Peterson Segmentor para el chino.

Por último, propondremos un tipo de detección de valores atípicos (o sucios) más extensivo que el que hace uso por defecto sistemas como Moses (que solo excluye los segmentos vacíos o muy largos), ya que veremos que en motores, aunque entrenados con corpus de la industria de la traducción, dicha limpieza básica no será insuficiente. La limpieza propuesta comprenderá: la comprobación de errores ortográficos, detección de incoherencias en números y signos, división de frases muy largas y la exclusión de segmentos con origen y destino idénticos.

Todas las propuestas se aplicarán en la fase de preproceso de los datos, que hay previo al entrenamiento del motor de traducción automática. El entrenamiento de los motores se hará en unos casos con el sistema Moses, y en otros con el sistema PangeaMT (basado en Moses).

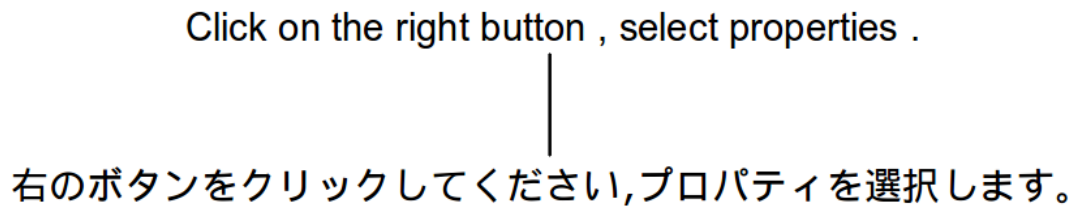
4.1. Tokenización

Un proceso tradicional en la traducción automática estadística es la segmentación de los textos, para así separar los signos de puntuación de las palabras y así que los alineadores (basados en palabras) funcionen correctamente. Dicha segmentación no requiere ningún análisis lingüístico o morfológico.

Un proceso tradicional en la traducción automática estadística es la segmentación de los textos, para así separar los signos de puntuación de las palabras y así que los alineadores (basados en palabras) funcionen correctamente. Dicha segmentación no requiere ningún análisis lingüístico o morfológico.

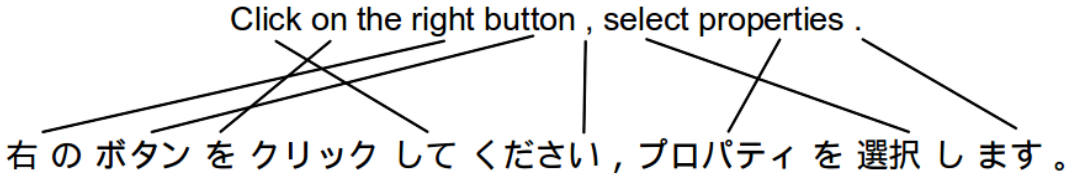
Tradicionalmente para la traducción de lenguas de dicho tipo se habían usado sistemas basados en reglas (RBMT), los cuales, debido al análisis lingüístico/morfológico previo que realizaban, no tenían problema alguno que apenas tuvieran espacios entre las palabras. Pero los sistemas estadísticos basados en palabras o frases, no pueden trabajar directamente con los datos así, ya que tratarían de alinear una gran palabra -el japonés o el chino- con múltiples palabras -el inglés-, por lo que en frases con múltiples palabras fallarían completamente.

Figura 4.1: Mala alineación en el japonés



La idea de tokenizar o segmentar es para dejar una cantidad parecida de palabras o morfemas entre la lengua fuente y la lengua destino para que así puedan los alineadores de los sistemas estadísticos funcionar.

Figura 4.2: Correcta alineación en el japonés



Para realizar dicha segmentación utilizaremos analizadores morfológicos que permitan detectar palabras o morfemas y así detrás de cada uno introducir un espacio detrás, y así de este modo que los alineadores a nivel de palabra (GIZA++ en nuestro caso) puedan relacionar palabras o grupo de palabras más fácilmente entre la lengua fuente y la lengua destino.

Los analizadores morfológicos que probaremos serán los únicos que encontramos en el momento de los experimentos, el Mecab y el KyTea para japonés, y el Peterson Segmentor para el chino.

4.2. Reordenación

El otro problema lingüístico que habíamos visto, es que en algunos idiomas el orden gramatical de las palabras difiere del de otros, llegando a extremos como el del japonés cuya estructura gramatical difiere bastante del inglés o del chino, sujeto-objeto-verbo (SOV) en vez de sujeto-verbo-objeto (SVO). Por ejemplo, para la frase inglesa siguiente:

Continue to press the button to scroll through the components of the program until the display shows the desired current selection.

Su orden en japonés sería el siguiente:

the display the desired current selection shows until the components the program of through to scroll the button to press continue.

Por lo que podemos ver, el orden cambia radicalmente. Para poder solucionar dicho problema, veremos que ha habido distintas aproximaciones para modelar la reordenación que habría que realizarse.

Actualmente, los sistemas basados en frases [Och02] [KOM03] [ON04], con Moses como claro ejemplo, son el estado actual del arte por su robustez en el modelado del reordenamiento local de palabras y en el eficaz algoritmo de decodificación. Sin embargo, cuando dichos sistemas son aplicados entre lenguas SOV y SOV, el reordenamiento en frases largas se vuelve su clara debilidad. Muchos métodos de reordenamiento han sido propuestas en los últimos años para solventar dicho problema.

El primer tipo de aproximación trata explícitamente de modelar la distancia de reordenamiento de la frases. El modelo de distorsión basado en distancias [Och02] [KOM03], que es una manera de modelar dicho reordenamiento de frases, penaliza la no monotonicidad aplicando un peso al número de palabras entre dos frases origen y sus dos frases destino consecutivas. Más tarde dicha aproximación fue extendido

al reordenamiento de frase lexicalizado [Til04] [KAM+05] [AOP06] aplicando diferentes pesos a diferentes frases. Más recientemente, los modelos de reordenamiento de frases jerárquicas [GM08] fueron propuestos para determinar dinámicamente las distintas partes de la frase haciendo uso del eficaz algoritmo de parseado "shift-reduce". También siguiendo esta linea de investigación, los modelos de reordenación discriminativos basados en clasificadores de máxima entropía [ZN06] [XLL06] también mostraron mejoras los modelos de distorsión. Ninguna de los aproximaciones comentadas cambia el alineamiento de las palabras de los sistemas basados en frases, y además están limitados a una distancia máxima de reordenamiento que puede ser usadas durante la decodificación.

El segundo tipo de aproximación, también conocido como modelos jerárquicos, incluye análisis sintáctico de la lengua destino, tanto en el modelado como en la decodificación. El modelado directo de los constituyentes de la lengua destino, tanto en árboles de constituyentes [YK01] [GGK+06] [ZVOP08] o en árboles de dependencias [QMC05] han mostrados buenos resultados. Otra alternativa, los modelos jerárquicos [Chi05] [Wu97] también han mostrado buenos resultados. De manera similar a los modelos de reordenamiento basados en distancias, los modelos jerárquicos (o sintácticos) también dependen de otros modelos para conseguir el alineamiento de las palabras. Este tipo de modelos que combinan la decodificación con el parseado de tablas ("chart parsing") incrementan la complejidad de la decodificación (y por lo que luego veremos en los experimentos) e incrementan el tiempo de traducción/entrenamiento. A pesar de los recientes trabajos para mejorar la eficiencia en la decodificación de los modelos jerárquicos [HC07], aún no son tan eficaces como los modelos basados en frases, sobretodo cuando los modelos de lenguajes son de gran tamaño.

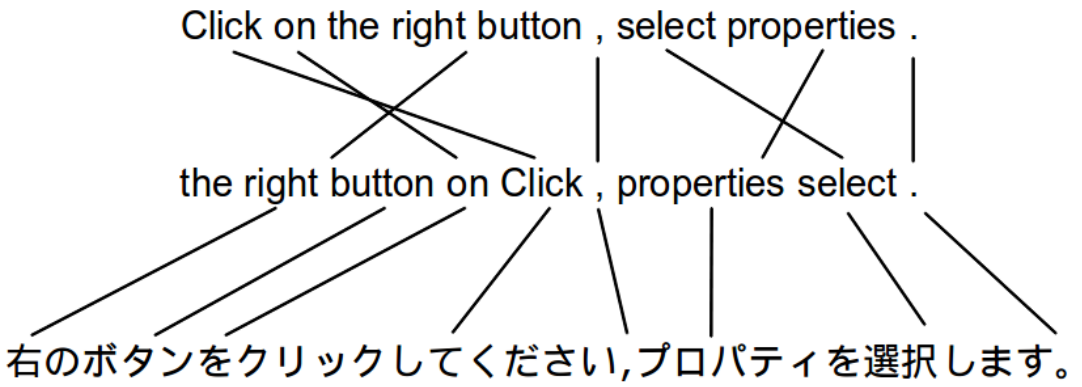
El tercer tipo de aproximación ha sido tratar de incluir análisis sintáctico de la lengua origen en la reordenación para que de manera determinista[1] reordenar la cadena de entrada [XM04] [CKK05] [WCK07] [Hab07] [XKRO09] o para proveer múltiples reordenamientos como opciones ponderadas [ZZN07] [LLZ+07] [Elm08]. En estas aproximaciones las cadenas de entrada son reordenadas durante el preproceso basándose en el análisis sintáctico y algunas reglas de reordenamiento. Dichas reglas de reordenamiento pueden haber sido realizadas manualmente o pueden haber sido extraídas automáticamente de los datos. El reordenamiento determinístico durante el preproceso (tanto en el proceso de traducción como en el entrenamiento) basándose en el análisis sintáctico de la cadena de entrada ha probado ser un buen modo de solucionar el problema de reordenación de largas distancias, sin introducir ninguna complejidad en la decodificación, y además permite ser integrado sin ningún problema en los sistemas basados en frases.

La propuesta de reordenación de esta tesis, que ya fue presentada en [YE11] (artículo conjunto entre Pangeanic y Toshiba presentado en la AAMT), está principalmente motivada en la aproximación de reordenamiento durante el preproceso, donde las

[1] http://es.wikipedia.org/wiki/Determinismo

reglas de reordenamiento usadas son las usadas por el sistema RBMT de Toshiba (The Honyaku). Dichas reglas han sido realizadas y perfeccionadas por expertos lingüistas de Toshiba durante cerca de 30 años, y por lo que veremos durante los experimentos han probado ser eficientes en la traducción entre el inglés y el japonés, y que para una entrada como la siguiente en inglés es reordenada al orden inglés, que llamaremos de ahora en adelante “niponización”.

Figura 4.3: Reordenación del inglés y alineación directa con el japonés



El anterior ejemplo muestra una frase en inglés y su correspondiente traducción en japonés además de los alineamientos entre grupos de palabras. Pero si recordamos lo que comentamos en la sección anterior (sección Sección 4.1), dicho texto no puede entrar directamente a los sistemas basados en frases, por lo que antes habría aplicar alguno de los sistemas de tokenización expuestos en la sección anterior, para que así se pueda hacer las siguientes alineaciones:

Figura 4.4: Alineaciones inglés-japonés

the right button → 右のボタン

on → を

Click → クリックしてください

, → ,

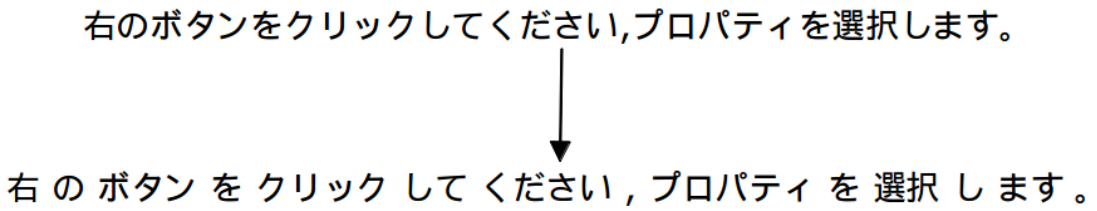
properties → プロパティを

select → 選択します

. → 。

Tras tokenizar se quedaría de este modo:

Figura 4.5: Tokenización



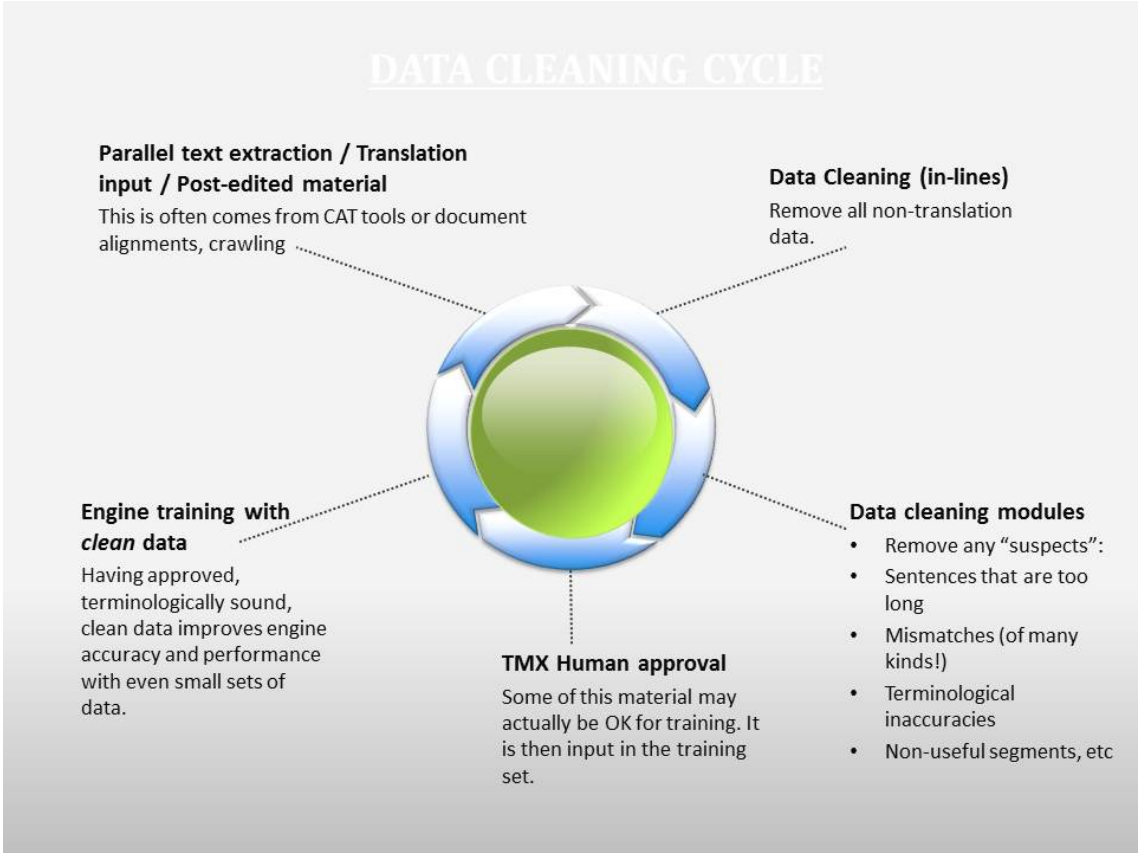
Una vez realizada la reordenación y tokenización de los textos de entradas, tendremos que asegurarnos que no se aplica ningún otro modelo de reordenamiento que tenga el sistema de MT usado, en nuestro caso los modelos de distorsión de Moses, obligándolo de ese modo a que durante la decodificación se haga una traducción monótona.

4.3. Detección de valores atípicos

Por último, el otro problema que habíamos comentado que se puede encontrar en un corpus de entrenamiento bilingüe, son los valores atípicos o sucios, que son segmentos que introducirán ruido en nuestros experimentos, pero que su localización en corpus de gran tamaño se convierte en una tarea casi imposible.

Para ello propondremos un conjunto de distintas detecciones de valores atípicos más extensivo que el que hace uso por defecto sistemas como Moses, ya que veremos que incluso en corpus de la industria de la traducción, dicha limpieza básica no será suficiente. Los segmentos detectados serían excluidos de los datos de entrenamiento, y si quisiéramos que fueran incluidos en el entrenamiento tendríamos que revisarlos a "mano", tarea bastante ardua y que tendría que ser realizada por expertos lingüistas.

Figura 4.6: Flujo de limpieza



Los valores atípicos mencionados pueden consistir en lo siguiente:

- Errores ortográficos
- Inconsistencias en los signos de puntuación y numeración
- Segmentos demasiado largos
- Segmentos vacíos
- El segmento de la lengua de origen coincide plenamente al de la lengua de destino

Tradicionalmente la detección de valores atípicos (o limpieza), por ejemplo, en un sistema base de Moses, ha consistido únicamente en la exclusión de los datos de entrenamiento de los segmentos demasiado largos o de los segmentos vacíos.

En la limpieza que proponemos, además de hacer dicha limpieza, se haría lo siguiente:

- Comprobación ortográfica (para lenguas orientales como el coreano, japonés y chino no se podrían comprobar)
- Comprobación de los signos
- Comprobación de segmentos idénticos

- División de frases

Dicho proceso de detección y exclusión de valores atípicos ya ha sido integrada en PangeaMT, y es seleccionable por el usuario del sistema a la hora de subir los datos de entrenamiento por el motor de traducción automática.

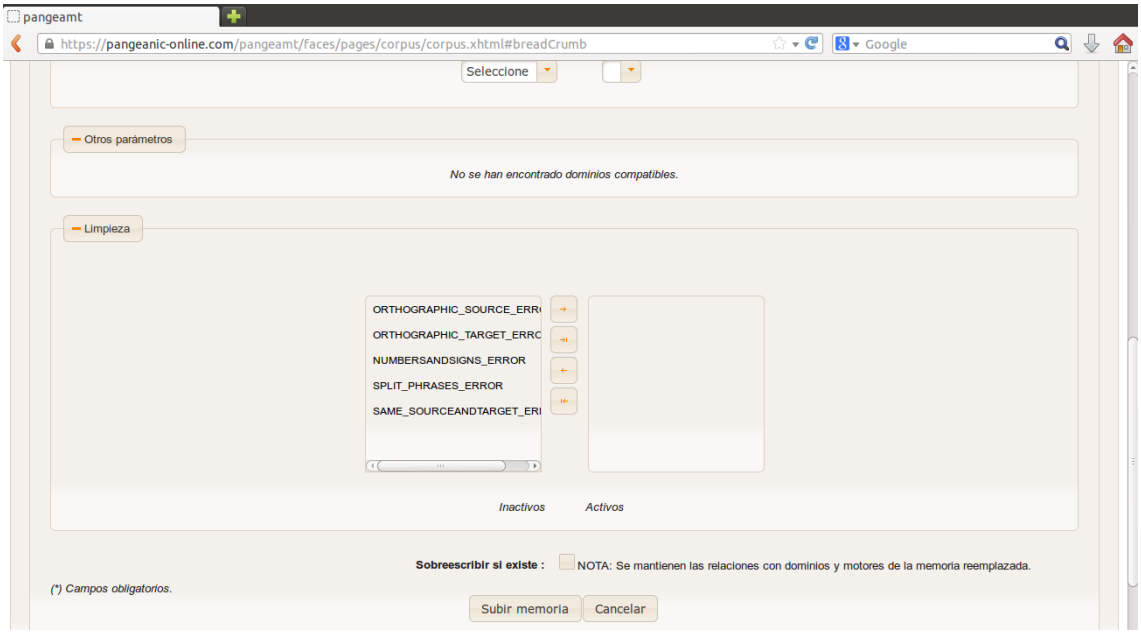


Figura 4.7: Limpieza en PangeaMT

En el cuadro 4.1 podemos ver algunos ejemplos de segmentos que podríamos encontrar en un corpus inglés-japonés (algunos de los ejemplos han sido sacados del corpus 2 TAUS Inglés-Japonés que veremos posteriormente).

Cuadro 4.1: Ejemplos de casos de valores atípicos

Lengua de origen (EN)	Lengua de destino (JP)	Caso
F ront panel components	フロント パネルの各部	Error ortográfico en la lengua de origen
Pour plus d'informations sur ce programme en Amerique du Nord, consultez le site Web HP.	Pour plus d'informations sur ce programme en Amerique du Nord, consultez le site Web HP.	Error ortográfico en la lengua de origen
/**h[elp]** or /?	/h[elp]または/?	Error ortográfico en la lengua de origen
“F ront panel” components	フロント パネルの各部	Desearíamos que esto no fuera detectado como error ya que es algo que han querido destacar entre comillas dobles
Front “pannnel” components	フロント パネルの各部	Desearíamos que esto no fuera detectado como error ya que es algo que han querido destacar entre comillas dobles
Optional PCI-X	PCI-X ・（オプション）	Error de puntuación en la lengua de destino
SmartStart	SmartStart **8**	Error de numeración en la lengua de destino
SmartStart	スマートスタート	Desearíamos que esto no fuera detectado como error ya que esto es un nombre propio
SmartStart Eight	SmartStart **8**	Error de numeración en la lengua de destino
Microsoft Office	Microsoft Office	Segmentos idénticos
Click on the right button. Select properties.	右のボタンをクリックしてください。プロパティを選択します。	Frase que puede ser dividida por los puntos
Front panel components	フロント パネルの各部	Error de puntuación en la lengua de destino por el símbolo “&” que esta siendo usado para codificar las etiquetas “”
Click on the right button. Select properties.	右のボタンをクリックしてください。プロパティを選択します。	Este segmento al ser realmente dos frases, lo desearíamos dividido en dos segmentos distintos.

A continuación veremos en más detalle cada una de las distintas detecciones y que realizadas juntas comprenderán la limpieza propuesta.

4.3.1. Comprobación ortográfica

Esta comprobación consistiría en analizar con un analizador ortográfico la lengua origen y/o la lengua destino, en nuestro caso sería con la herramienta Aspell que cuenta con licencia GNU, aunque dicha herramienta actualmente no cuenta con la capacidad de analizar textos orientales como el japonés, el coreano y el chino.

Serviría para detectar si algún segmento se ha quedado sin traducir (misma lengua en el segmento de la lengua de destino que en el de la lengua de origen) o el texto en ambos segmentos está en una lengua distinta de las lenguas implicadas en el desarrollo o entrenamiento, con el chequeo de ortografía se excluirían estos segmentos, aunque a riesgo de perder cierta cobertura lingüística de las palabras que si fueran correctas y que no están contempladas en los diccionarios del Aspell.

Para tratar de minimizar dicha pérdida de cobertura lingüística, omitiríamos de la comprobación: las palabras en mayúsculas (ya que podrían ser nombres propios o nombre de producto comercial); palabras entre símbolos de comillas/cita[2] (ya que podrían ser términos que se han querido destacar por algún motivo), los códigos SGML/XML o HTML, y las URLs y emails.

4.3.2. Comprobación de los signos

Esta comprobación sería para comprobar que los números o determinados signos estén tanto en la lengua de origen como en la lengua de destino.

Esto serviría para detectar segmentos que no han sido fielmente traducidos, o que no corresponde la traducción.

Los signos de puntuación a vigilar serían:

- Símbolos matemáticos "+ = % *"

  Se vigilaría "+ = % *" ya que son términos que sabemos son usados principalmente en fórmulas y que por lo tanto no tienen una traducción distinta entre idiomas. El símbolo "menos" ("-") no querríamos que fuera comprobado ya que dicho símbolo es en ocasiones usado para otros fines no matemáticos (para palabras "dobles" como "post-proceso", o para añadir información extra como "He is happy -but not really much, but he want to leave") y que en otros idiomas puede no aparecer y ser igualmente correcto. El símbolo "barra" ("/") tampoco querríamos que fuera comprobado ya que dicho símbolo tiene bastante uso para abreviaciones ("s/n", sin número) o para ofrecer dos opciones ("niño/a"), y que en otros idiomas puede no aparecer y ser correcto.

- El ampersand ("&")

  Aunque en ocasiones el ampersand es también es usado como alternativa al "y" en el inglés ("Hot & Cool"), su uso principal en corpus de la industria de la traducción es para codificar "tags" (por ejemplo la etiqueta XML "" se codificaría como" "), las cuales se suponen que tienen que estar en ambos idiomas por lo que querremos comprobar que se mantienen.

- La almohadilla o "hashtag" ("#")

  La almohadilla es un símbolo cuyo uso principal es indicar un "tema" (o "topic"), el cual principalmente se usa en IRC o en Twitter. Dicho símbolo no cuenta con traducción distinta entre idiomas distintos por lo que consideramos deseable que entre las lenguas origen y distinto se mantenga dicho símbolo. En algunas ocasiones puede ocurrir en lenguas como la lengua inglesa que dicho símbolo sea usado para numerar (Nº 1 → #1), aunque dicho uso es minoritario y responde más a "estilos" de traducción.

[2] http://en.wikipedia.org/wiki/Non-English_use_of_quotation_marks

- Los dos puntos (":")

  Los dos puntos es un símbolo que puede indicar un división (cuando va entre dos números "4:2=2"), una separación entre minutos y hora ("12:30h"), o puede indicar el inicio de una lista de elementos. Tiene su uso en muchas lenguas y no es traducido, por lo que querremos comprobarlo.

- Los paréntesis ("()", "{}" y "[]")

  Por último, también querremos comprobar los paréntesis (los propiamente dichos "()", las llaves "{ }" y los corchetes "[ ]") que tienen un uso extendido entre la mayoría de las lenguas y que no son traducidos. Su uso suele estar en introducir una información adicional en la traducción, la cual no está en el segmento de la lengua de origen, por lo que veremos como deseable quitar dicha información adicional.

4.3.3. Comprobación de segmentos idénticos

Esta comprobación buscaría los segmentos idénticos en la lengua de origen y la lengua de destino (segmentos no traducidos, o nombre de persona, producto empresarial, URLs, código de programación...), ya que podrían ser considerados como errores al no añadir ninguna información para el entrenamiento o desarrollo del motor.

4.3.4. División de frases

La división de frases, no sería por si una comprobación como el resto, ya que en vez de excluir frases, lo que hace dividir la frase si la lengua de origen y la lengua de destino tienen por lo menos un recuento de 2 signos "punto" (también se tendría en cuenta el punto japonés) en cada parte (excluyendo los puntos suspensivos), y el recuento en cada parte es la misma. Esta división serviría principalmente para las frases demasiado largas, que acaban siendo descartadas en la limpieza original de Moses, que no sean descartadas.

5 Experimentación

Los experimentos se han centrado en comprobar la validez de las distintas propuestas presentadas presentadas en el Capítulo 4 y además ver si el estado del arte actual en MT, los modelos basados en frases, siguen siendo válidos para lenguas tan distantes lingüísticamente como el japonés y el inglés.

5.1. Preparación de los experimentos

En esta sección presentamos las distintas partes implicadas en los experimentos tales como las herramienta de preprocesado, entrenamiento y evaluación de los experimentos; además del hardware usado. Para la evaluación usaremos la métricas de calidad BLEU y TER. Para el entrenamiento de los motores utilizaremos Moses y PangeaMT, además de otras herramientas que veremos más adelante (Subsección 5.1.2).

5.1.1. Medidas de evaluación

Para evaluar la calidad de las traducciones se han utilizado dos métricas estadísticas distintas: el BLEU y el TER. La primera, el BLEU, porque tiene una alta correlación con el juicio humano; y la segunda el TER, para medir el esfuerzo en conseguir la salida deseada.

El BLEU [PRWZ02] fue desarrollado originalmente por investigadores de IBM en 2002. Evalúa comparando a nivel de frase la precisión en ngramas entre la salida de la traducción automática y la traducción humana de referencia, y es independiente de la lengua usada.

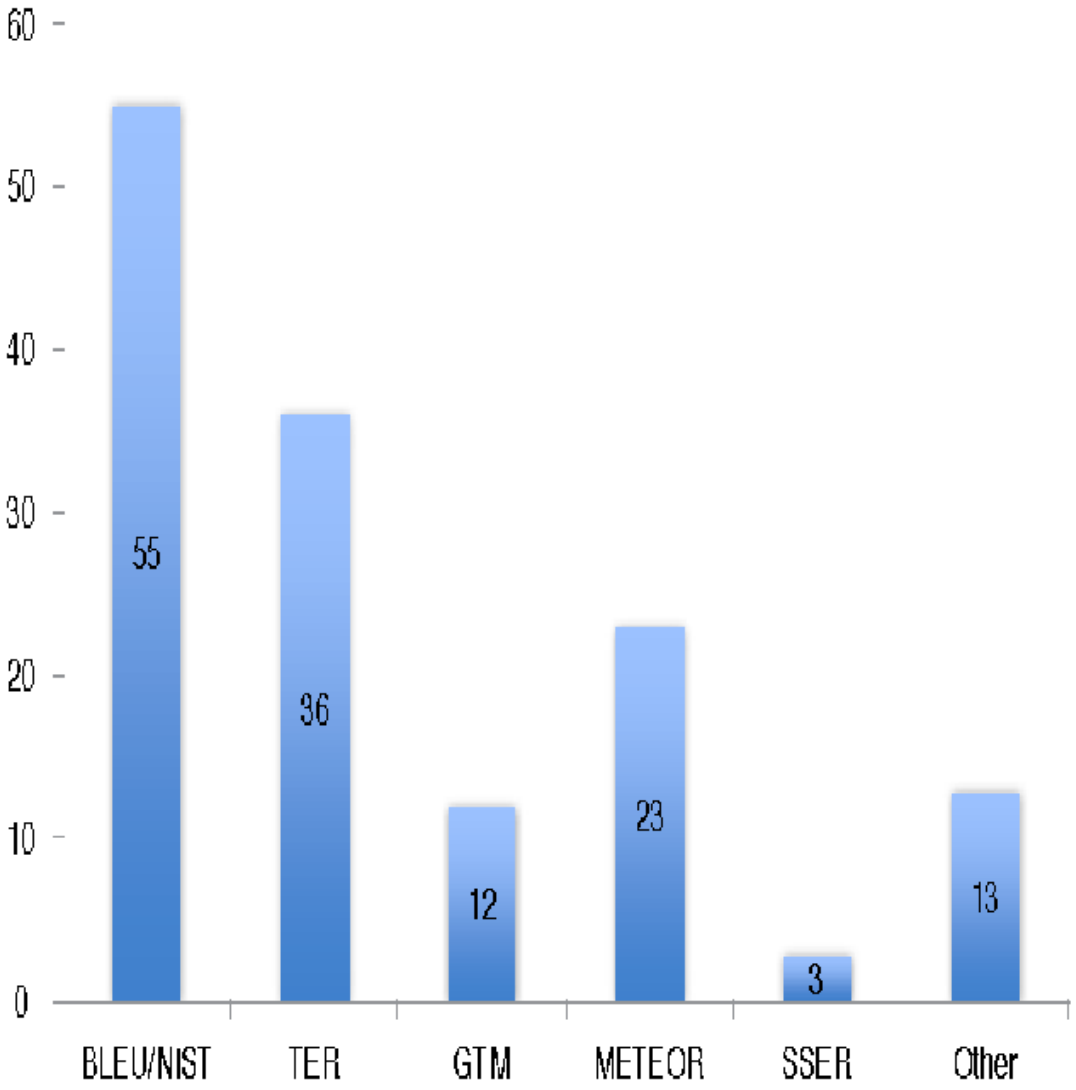
Por muchos años, la principal manera que han tenido los investigadores para medir las mejoras en la calidad de la traducción automática ha sido optimizar los sistema de MT contra los resultados obtenidos mediante el BLEU. Por lo que ha sido la herramienta de evaluación estándar para la investigación académica en MT.

El TER [SDS+06] evalúa midiendo la cantidad de ediciones necesarias para cambiar la salida de la traducción automática hasta obtener la traducción humana de referencia, y, al igual que el BLEU, es independiente de la lengua usada. El resultado del TER es normalizado por la distancia media de las referencias.

Mientras que en el BLEU cuanto mayor sea el score mejor resultado (siendo el mejor resultado 100), en el TER cuanto menor sea el score mejor resultado (siendo el mejor resultado 0).

Según la encuesta realizada por TAUS en el año 2012 entre sus miembros (principalmente agencias de traducción), las medidas de evaluación más usadas en la industria de traducción profesional siguen siendo las dos propuestas (véase la figura 5.1).

Figura 5.1: Resultados encuesta métrica de evaluación por TAUS en 2012



Y para los casos donde no podremos discernir si un resultado es claramente mejor que otro utilizaremos la técnica de significancia estadística, en concreto mediante el "Paired difference interval" propuesto en [ZV04].

Los intervalos de confianza contienen a priori una probabilidad elevada (0.95 y un

0.90) de contener el valor poblacional desconocido "media poblacional" del valor en estudio (en nuestra caso del BLEU) a partir de las muestras de test.

Para evidenciar diferencias estadísticamente significativas simplemente hemos de observar si el 0 está o no incluido en el intervalo. En el caso de no estarlo existe evidencia estadísticamente significativa con un riesgo de equivocarnos en la decisión tomada del 5 % (y del 10 %) de que los dos sistemas producen valores del valor en estudio diferentes.

Durante los experimentos, uno de los problemas que veremos que tienen estas herramientas de evaluación es que, en lenguas como el japonés y el chino, donde no hay espacios entre las palabras, se precisa tener segmentado los distintos tokens o palabras del segmento a comparar. Posteriormente, la salida final que se ofrecería al usuario sería una salida donde se hayan detokenizado los espacios de más añadidos entre palabras.

5.1.2. Herramientas

Ya que las propuestas expuestas en esta tesis han sido para aplicarlas posteriormente en el aplicación PangeaMT, algunos de los experimentos ha sido realizados con la aplicación de MT Moses (visto en Subsección 3.1.1), y otros con la aplicación PangeaMT (visto en Subsección 3.2.10).

El resto de herramientas usadas, todas con licencia GNU, han sido:

- GIZA++

  Para la alineación no-supervisada de las palabras usaremos la herramienta GIZA++[1], que es un sistema basado en palabras (visto en Subsección 1.2.1), pero es el sistema en el que muchos de los sistemas basados en frase se basan. Como resultado final producirían el modelo de traducción basado en frases (visto en Subsección 1.2.2).

- IRSTLM y KEN-LM

  Los modelos de lenguaje (visto en Sección 1.2) se consiguen con herramientas como el IRSTLM[2], SRILM[3] o el Ken-LM[4]. En el momento de los experimentos, la aplicación de MT Moses usaba por defecto SRILM, pero esta tiene una cara licencia para cuando es para uso no académico. Pero las herramientas IRSTLM y Ken-LM tienen licencia LGPL, por lo que con ellas no tendríamos problema para su posterior uso en PangeaMT, por lo que será con estas últimas dos con las que haremos los entrenamientos.

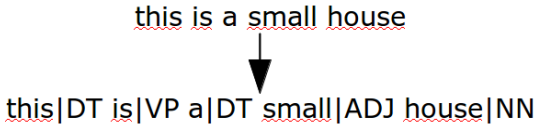
- MXPOST y Collins Parser

[1] http://code.google.com/p/giza-pp/
[2] http://hlt.fbk.eu/en/irstlm
[3] http://www.speech.sri.com/projects/srilm/
[4] http://kheafield.com/code/kenlm/

En los experimentos (de la Sección 5.2.4) donde compararemos los modelos basados en frases contra los modelos factoriales (visto en Subsección 1.2.3) y los modelos sintácticos (visto en Subsección 1.2.4), las herramientas que usaremos para añadir la información extra en el inglés que necesitan dichos modelos serán MXPOST[5] y Collins Parser[6], ambas recomendadas desde la web de Moses[7].

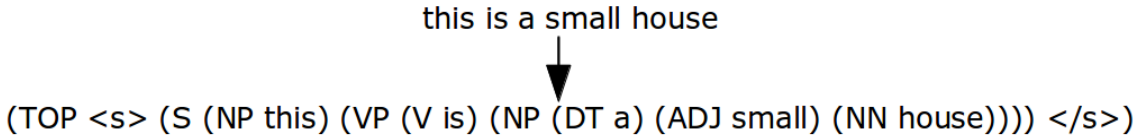
El MXPOST es una herramienta para el etiquetado gramatical ("part-of-speech") del inglés desarrollado por Adwait Ratnapakhi, y está basados en modelos de máxima entropía. Esta herramienta ha sido la utilizada para los modelos factoriales. Véase un ejemplo a continuación:

Figura 5.2: Ejemplo POS-tagueado con MXPOST



La otra herramienta usada para los modelos sintácticos ha sido el Collins Parser, que fue desarrollada por Michael Collins. Es una herramienta para parseado sintáctico del inglés, que a nivel interno usa el etiquetado gramatical del MXPOST. Véase un ejemplo a continuación:

Figura 5.3: Ejemplo parseado sintáctico con Collins Parser



- KyTea, Mecab y Peterson Segmentor

  Para los experimentos de tokenización del japonés y del chino (de la Subsección 5.2.1), las herramientas que usaremos serán el KyTea[8], el Mecab[9] y el Peterson Segmentor[10]. Se utilizaron dichas herramientas ya que en el momento de los experimentos fueran las únicas halladas que funcionaran para dicho fin, ya que los tokenizadores utilizados por Moses no son compatibles para el japonés y el chino.

[5]http://www.inf.ed.ac.uk/resources/nlp/local_doc/MXPOST.html
[6]http://www.cs.columbia.edu/~mcollins/code.html
[7]http://www.statmt.org/moses/
[8]http://www.phontron.com/kytea/
[9]http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html
[10]http://www.mandarintools.com/segmenter.html

La herramienta KyTea (Kyoto Text Analysis) es un analizador morfológico del japonés desarrollado por Graham Neubig. Permite la segmentación en palabras o morfemas del japonés. Para ello utiliza un clasificador punto a punto basado en máquinas de soporte vectorial (SVM), lo cual significa que cada separación entre dos caracteres es estimada de manera separada.

La herramienta Mecab es un analizador morfológico del japonés desarrollado por Yuuichi Teranishi. Permite la segmentación en palabras o morfemas del japonés. Está basado en modelos de Markov para el análisis del texto. También trata de localizar los nombres propios.

La herramienta Peterson Segmentor fue realizada por Erik Peterson para la segmentación del chino simplificado. Para ello utiliza un lexicón de tokens del chino, y la herramienta trata de encontrar el tokem más largo. También trata de localizar los nombres propios.

- Aspell

  Para la detección de valores atípicos (de la Subsección 5.2.3), la única herramienta externa usada ha sido Aspell[11], la cual utilizaremos para la comprobación ortográfica de texto plano.

5.1.3. Corpus

Para los experimentos se han utilizado distintos corpus de entrenamiento, cada uno utilizado para una finalidad distinta. En cada corpus se ha podido realizar un preprocesado distinto, el cual será explicado en cada experimento.

Para las estadísticas iniciales que presentaremos en cada corpus se ha realizado este preprocesado básico usando los script de Moses: tokenizado (usando el script "tokenizer.perl"), limpieza (usando el script "clean-corpus-n.perl ") y paso a minúsculas (usando el script "lowercase.perl").

Todos los corpus (excepto el corpus 1 KFTT que ya venía segmentado) han sido barajados aleatoriamente y han sido divididos en tres particiones para el entrenamiento de los motores:

- Training: datos bilingües con los que se entrenaran los modelos
- Tuning: datos bilingües, distintos a los del training, con los que se optimizan el modelo estadístico obtenido durante el entrenamiento
- Testing: datos bilingües, distintos a los del training, con los que se evaluará el sistema

Nótese que en las estadísticas mostradas de los corpus hemos abreviado las cantidades con "k" para los miles y con "M" para los millones.

[11] http://aspell.net

Corpus 1 KFTT Inglés-Japonés

Corpus libre de japonés-inglés de la página de la "The Kyoto Free Translation Task" [Neu11], cuya temática está centrado en artículos de la Wikipedia referentes a Kyoto.

El corpus de la KFTT tiene como finalidad la evaluación y desarrollo de sistemas de traducción japonés-inglés (visto en Sección 5.2.1), y sobre el que ya hay publicados resultados de motores creados con el Moses, por lo que nos será útil para comparar los segmentadores KyTea y Mecab, y comparar los modelos basados en frases de Moses y los modelos basados en otras aproximaciones que permite también el Moses.

El corpus KFTT, a pesar de no ser de gran tamaño, es un corpus de gran complejidad, ya que ha sido sacado de distintos artículos de la Wikipedia, por lo que ha habido distintos traductores involucrados que no tienen que ser por que sí traductores profesionales. Al no haber sido una traducción profesional (la cual implica un "Quality Check" posterior), se han podido encontrar diversas maneras distintas para traducir lo mismo.

En las estadísticas del corpus, que vemos en el cuadro 5.1, no se ha utilizado la tokenización del japonés con las herramientas KyTea y Mecab (visto en Figura 5.1.2).

Tal como podemos observar en las estadísticas, es un corpus de pequeño tamaño. El japonés difiere mucho del inglés en la cantidad de palabras, que es muy baja, y la cantidad de palabras distintas (vocabulario), que es muy alta. Ello se debe a que cuenta "prácticamente" la frase entera como una "única" palabra, la cual es muy difícil que se repita, lo que provoca un alto número de palabras distintas (vocabulario).

Cuadro 5.1: Estadísticas corpus 1 KFTT

	Segmentos	**Palabras**	**Vocabulario**	**Fuera de vocabulario**	**Perplejidad (5-grama)**
Training	440M	(EN) 11,5M (JA) 440k	(EN) 190k (JA) 419k		
Tuning	1,2k	(EN) 31k (JA) 1,2k	(EN) 4,3k (JA) 1,1k	(EN) 607 (JA) 1202	(EN) 157.01 (JA) 104.74
Testing	1,2k	(EN) 26k (JA) 1,2k	(EN) 4,4k (JA) 1,1k	(EN) 443 (JA) 1142	(EN) 140.61 (JA) 98.07

Corpus 2 TAUS Inglés-Japonés

El corpus 2 es un corpus privado obtenido por Pangeanic a través de la corporación TAUS para el par lingüístico Inglés-Japonés. El dominio del corpus es software.

Hemos escogido este corpus con finalidades comerciales, y sobre el cual aplicaremos la reordenación propuesta en la Sección 4.2 (visto en Subsección 5.2.2). De dicho

corpus se esperan buenos resultados por provenir de fuentes bastante fiables, por contener muchas variantes de frases similares (lo cual producirá que las palabras desconocidas de la partición de test sean 0), y por contener una gran cobertura lingüística en dicho dominio.

En las estadísticas del corpus, que vemos en el cuadro 5.2, no se ha utilizado la tokenización del japonés con las herramientas KyTea y Mecab (visto en Figura 5.1.2).

Tal como podemos observar en las siguientes estadísticas, es un corpus de gran tamaño, y que al igual que el Corpus 1, la partición del japonés difiere mucho del inglés en la cantidad de palabras, que es muy baja, y la cantidad de palabras distintas (vocabulario), que es muy alta; lo cual se debe a que cuenta "prácticamente" la frase entera como una "única" palabra, la cual es muy difícil que repita.

Cuadro 5.2: Estadísticas corpus 2 TAUS

	Segmentos	**Palabras**	**Vocabulario**	**Fuera de vocabulario**	**Perplejidad (5-grama)**
Training	5,3M	(EN) 82,3M (JP) 16,2M	(EN) 423k (JP) 4,8M		
Tuning	2000	(EN) 31k (JP) 6,2k	(EN) 4,9k (JA) 4,4k	(EN) 0 (JP) 0	(EN) 92.75 (JP) 125.33
Testing	2000	(EN) 30k (JP) 6,1k	(EN) 4,8k (JA) 4,4k	(EN) 0 (JP) 0	(EN) 90.34 (JP) 130.01

Corpus 3 TAUS Inglés-Chino

El corpus 2 es un corpus privado obtenido por Pangeanic a través de la corporación TAUS para el par lingüístico Inglés-Chino. El dominio del corpus es software y electrónica.

Hemos escogido dicho corpus para probar la viabilidad de la traducción del inglés-chino con el sistema PangeaMT (visto en Sección 5.2.1), aunque para ello veremos que será necesario la tokenización del chino propuesta en la Sección 4.1, mediante la herramienta Peterson Segmentor (visto en Figura 5.1.2).

En las estadísticas del corpus, que vemos en el cuadro 5.3, no se ha utilizado la tokenización del chino.

Tal como podemos ver en el cuadro de estadísticas siguiente, es un corpus de no gran tamaño, y del que hemos visto que no tiene gran cobertura lingüística.

Cuadro 5.3: Estadísticas corpus 3 TAUS

	Segmentos	**Palabras**	**Vocabulario**	**Fuera de vocabulario**	**Perplejidad (5-grama)**
Training	445k	(EN) 5,7M (ZN) 2,2M	(EN) 97k (ZN) 626k		
Tuning	2000	(EN) 25,6k (ZN) 9,9k	(EN) 6,1k (ZN) 4,9K	(EN) 168 (ZN) 2256	(EN) 88.66 (ZN) 54.92
Testing	2000	(EN) 27,2k (ZN) 10,5k	(EN) 6,3k (ZN) 5,1k	(EN) 245 (ZN) 2362	(EN) 93.17 (ZN) 63.03

Corpus 4 privado Inglés-Polaco

El corpus 4 es un corpus privado obtenido por Pangeanic a través de un cliente para el par lingüístico Inglés-Polaco. El dominio del corpus es automoción, medicina y electrónica.

La finalidad del corpus ha sido la creación de un motor para dicho cliente. Sobre este corpus probaremos la limpieza propuesta en la Sección 4.3 (visto en Sección 5.2.3).

Tal como podemos ver en el cuadro 5.4 de estadísticas siguiente, es un corpus de no gran tamaño. Tiene gran cobertura lingüística debido a la mezcla de distintos dominios que tiene.

Cuadro 5.4: Estadísticas corpus 4 privado

	Segmentos	**Palabras**	**Vocabulario**	**Fuera de vocabulario**	**Perplejidad (5-grama)**
Training	770k	(EN) 8,7M (PL) 8,3M	(EN) 166k (PL) 256k		
Tuning	2000	(EN) 21,7k (PL) 20,8k	(EN) 5,9k (PL) 8,2k	(EN) 239 (PL) 319	(EN) 78.72 (PL) 116.05
Testing	2000	(EN) 23,7k (PL) 22,8k	(EN) 6,1k (PL) 8,7k	(EN) 235 (PL) 333	(EN) 82.18 (PL) 115.62

Corpus 5 privado Inglés-Japonés

El corpus 5 es un corpus privado obtenido por Pangeanic a través de un cliente para el par lingüístico Inglés-Japonés. El dominio del corpus es electrónica.

La finalidad del corpus ha sido la creación de un motor para dicho cliente. Sobre este corpus probaremos la limpieza propuesta en la Sección 4.3 (visto en Sección 5.2.3).

Tal como podemos ver en el cuadro 5.5 de estadísticas siguiente, es un corpus de pequeño tamaño. Tiene una baja cobertura lingüística, por lo que no esperamos grandes resultados de él.

Para los valores que se muestran del japonés, el proceso de tokenizado con Mecab en el preproceso ya viene realizado.

Cuadro 5.5: Estadísticas corpus 5 privado

	Segmentos	**Palabras**	**Vocabulario**	**Fuera de vocabulario**	**Perplejidad (5-grama)**
Training	216k	(EN) 2,6M (JP) 758k	(EN) 44k (JP) 291k		
Tuning	2000	(EN) 23,5k (JP) 6,8k	(EN) 4,3k (JP) 2,7K	(EN) 187 (JP) 124	(EN) 52.66 (JP) 23.10
Testing	2000	(EN) 26,4k (JP) 6,9k	(EN) 4,5k (JP) 2,8k	(EN) 218 (JP) 184	(EN) 54.09 (JP) 23.36

5.1.4. Hardware

Para la realización de los experimentos se ha contado con un servidor Linux con las siguientes características:

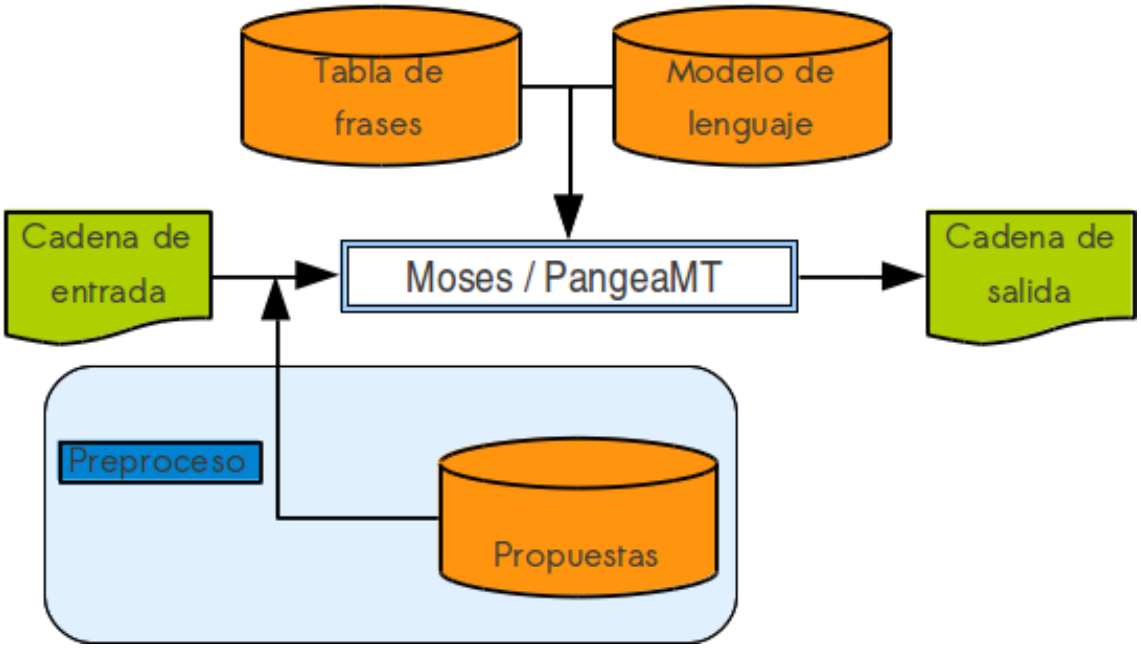
- Sistema operativo: GNU/Linux Ubuntu 11.04 con kernel de 64 bits
- Procesador: Intel(R) Core(TM) i7 CPU 960 @ 3.20GHz
- Memoria RAM: 24GB
- Disco duro: Disco SATA a 6 Gb/s de 2TB y 5900 revoluciones

5.2. Experimentos

Como ya introdujimos en puntos anteriores, los experimentos serán llevados a cabo con la aplicación Moses principalmente, en especial, para los experimentos que queremos comparar con otros resultados publicados que tengamos disponibles; y con la aplicación PangeaMT (basada en Moses), para el resto de experimentos.

Las propuestas de reordenación (del japonés), segmentación (del japonés y del chino) y detección de valores atípicos (del japonés y del polaco) propuestas serán llevadas a cabo durante el preproceso de los datos de entrada (ver figura 5.4).

Figura 5.4: Diagrama de traducción durante las propuestas



Los experimentos van a dividirse básicamente en cuatro partes:

- Tokenización: donde nuestro principal objetivo es ver el comportamiento de las distintas herramientas de tokenización del inglés y el chino que hemos encontrado
- Reordenación: aquí comprobaremos que la técnica de reordenación del japonés
- Detección de valores atípicos: aquí comprobaremos que hay casos donde una limpieza exhaustiva puede mejorar los resultados además de ahorrar tiempo en el entrenamiento de los motores
- Otros modelos: aquí finalmente comprobaremos si el estado actual del arte, los modelos basados en frases, sigue funcionando mejor que las aproximaciones presentadas en la Subsección 1.2.3 y Subsección 1.2.4, cuando los usamos entre lenguas tan distantes lingüísticamente como el inglés y el japonés

5.2.1. Tokenización

En estos experimentos aplicaremos la segmentación del inglés-japonés y del inglés-chino propuesta en la Sección 4.1 para solventar el problema de la falta de espacios (entre palabras o morfemas) que tienen tanto el japonés como el inglés.

Para estos experimentos utilizaremos los corpus: Corpus 1 KFTT inglés-japonés y el Corpus 3 TAUS inglés-chino.

Además para el Corpus 1 KFTT, el cual tiene resultados de BLEU publicados y un baseline (con una determinada configuración del Moses), veremos si dicho baseline es el más adecuado o si hay otras configuraciones que puedan afectar al resultado.

Durante la evaluación de los motores entrenados para inglés-japonés e inglés-chino, se han tenido que mantener la segmentación introducida entre las palabras o morfemas de las cadenas de texto de salida, ya que las medidas de evaluación usadas (BLEU y TER) evalúan a nivel de palabra. Pero en la salida final que se mostraría al usuario del sistema, dichos espacios son eliminados.

Inglés-Japonés

Para estos experimentos se ha utilizado el "Corpus 1 KFTT inglés-japonés".

Como la meta de estos experimentos es sobretodo comparar segmentadores del japonés. Para poder comparar los resultados que obtengamos con los publicados en la web de la KFTT hemos mantenido las particiones que ya habían sobre el corpus y hemos realizado el mismo preproceso propuesto.

También con estos experimentos hemos querido comprobar si la configuración propuesta en la KFTT es el más adecuado; por lo que también comprobaremos lo siguiente:

- si usar Mecab o KyTea para la segmentación: por defecto en el baseline de la KFTT utilizan KyTea, por lo que queremos compararlo con el otro segmentador que hemos encontrado para el japonés, el Mecab, el cual tiene una comunidad de desarrolladores más grande y activa que el KyTea
- traducción monótona o no monótona: con esto queremos demostrar de forma empírica que el japonés es una lengua cuyo orden gramatical difiere bastante del inglés, y que por lo tanto hace falta reordenamiento; la traducción monótona consiste en desactivar los modelos de distorsión del Moses (modelos de reordenamientos basados en distancia)
- usar Ken-LM o IRSTLM para los modelos de lenguaje: la herramienta para los modelos de lenguaje usada en los experimentos originales de la [Neu11] es el SRILM, pero para nuestros experimentos hemos usado IRSTLM (con licencia libre a diferencia de SRILM), el cual querremos comparar con el Ken-LM, el cual es el otro modelo de lenguaje disponible para Moses
- tamaño de los n-gramas de los modelos de lenguaje: el tamaño usado en la KFTT es de 5-gramas, por lo que queremos comprobar si es el tamaño adecuado o si al aumentar el tamaño podemos ayudar a aliviar el problema de las reordenaciones, que en las frases largas puede superar el tamaño de los n-gramas

Comparativa entre segmentadores Primera de todo, comprobamos que cambios puede producir en el corpus de KFTT el uso de un segmentador (KyTea en este caso) en el japonés (ver cuadro 5.6).

Cuadro 5.6: Comparativa segmentadores

Sin segmentación		**Con Kytea**		**Con Mecab**	
Palabras	Vocabulario	Palabras	Vocabulario	Palabras	Vocabulario
(EN) 11,5M	(EN) 189k	(EN) 11,5M	(EN) 189k	(EN) 11,5M	(EN) 189k
(JA) 440k	(JA) 419k	(JA) 11,3M	(JA) 80k	(JA) 11,4M	(JA) 81k

Se puede observar en la tabla 5.6, que tras segmentar con KyTea y Mecab, la cantidad de palabras entre el inglés y el japonés es ahora muy cercana. Tan solo que Mecab ha hecho ligeramente más segmentaciones (sobre 100k más).

A continuación vamos a detallar los experimentos, su nomenclatura y configuración:

- Experimento 1.1 (KyTea)

  Aquí reproduciremos el experimento tal como lo detallan en la tarea KFTT, usando KyTea para segmentar el japonés, SRLIM para los modelos de lenguaje, 5-gramas y con traducción no monótona. Realizamos este experimento para poder comparar el BLEU que obtengamos con el BLEU publicado en la KFTT (Referencia 1.2), y así marcar un resultado baseline.

- Experimento 1.2 (KyTea → Mecab)

  Aquí utilizaremos la misma configuración que el experimento 1.1, tan solo que cambiaremos que en vez de usar KyTea, usaremos Mecab para la segmentación del Japonés. Hacemos esto para poder comparar los segmentadores.

- Experimento 1.4 (Sin segmentador)

  Aquí utilizaremos la misma configuración que el experimento 1.1, tan solo que no utilizaremos ningún tokenizador para el japonés. Hacemos esto para medir que pasaría si decidiéramos no utilizar ningún tokenizador para introducir los espacios entre palabras o morfemas que carece el japonés.

Contamos con los siguientes BLEUs de referencia, los cuales han sido obtenidos de la web de la KFTT [KFTT 2011] (para dichas referencias el TER no está publicado):

- Referencia 1.1 (Google Translate)

  Resultado del traductor Google Translate obtenido para la fecha 2011-2-18.

- Referencia 1.2 (Original)

  Resultado publicado en la Kyoto Free Translation Task [Neu11]

Cuadro 5.7: Comparativa segmentadores del japonés

		BLEU		TER	
		EN → JA	JA → EN	EN → JA	JA → EN
Referencia 1.1	Google Translate	11.53	5.27		
Referencia 1.2	Original	20.32	10.58		
Experimento 1.1	KyTea	20.50	10.28	0.749	0.816
Experimento 1.2	Mecab	19.49	10.49	0.751	0.812
Experimento 1.4	Sin segmentador	0.00	0.00	22.55	0.998

Como podemos observar en el cuadro 5.7, los resultados que hemos obtenido en el experimento 1.1 (10.28, 20,50), han sido bastante similares a los que obtuvieron los de la KFTT (10.58, 20.32) (Referencia 1.2). Las ligeras diferencias podrían ser principalmente debidas a cambios en el toolkit de Moses respecto a la versión utilizada cuando lo hicieron ellos.

Como se puede observar en el experimento 1.4 el no uso de segmentadores hace que el Moses falle completamente, por lo que con esto corroboramos que su uso es obligatorio para una correcta traducción.

Otra cosa a observar es que mientras que el segmentador KyTea (experimento 1.1) da mejores resultados para la traducción de inglés a japonés, Mecab (experimento 1.2) da mejores para la traducción de japonés a inglés. Pero de todos modos, como los resultados son cercanos, se ha decidido comprobar los resultados mediante la técnica de intervalos de confianza de Zhang04 ("Paired diference interval").

Cuadro 5.8: Intervalos de confianza de Zhang04

en → ja	**ja → en**
conf.int.95 = [3.14 , 4.42]	conf.int.95 = [-.70 , .27]
conf.int.90 = [3.25 , 4.32]	conf.int.90 = [-.61 , .21]

Tal como puede observarse en el cuadro 5.8, que para JA → EN los intervalos se solapan con 0, por lo que, con un riesgo de equivocarnos de un 95 %, los dos sistemas producen valores similares. Pero por el otro lado, para EN → JA, al no solaparse el 0, los dos sistemas producen valores diferentes.

Comparativa entre traducción monótona o no monótona Ahora comprobamos que ocurre cuando desactivamos los modelos de reordenamiento que incorpora el Moses (modelos de distorsión). Desactivarlo significa que hará una traducción "monótona", es decir palabra a palabra. Tradicionalmente, entre lenguas "algo" parecidas respecto al orden gramatical, por ejemplo español e inglés, se observa que la traducción monótona es lo que mejor resultado da, por lo que se quería corroborar

era si la traducción monótona funcionaba o no entre lenguas tan dispares como el inglés y el japonés.

A continuación vamos a detallar los experimentos, su nomenclatura y configuración:

- Experimento 1.2 (Traducción no monótona)

  Aquí usaremos Mecab para la segmentación del japonés, IRSTLM para los modelos de lenguaje, 5-gramas y con los modelos de distorsión activados

- Experimento 1.3 (Traducción monótona)

  Aquí utilizaremos la misma configuración que el experimento 1.2, tan solo que no dejaremos que los modelos de distorsión hagan reordenación alguna

Cuadro 5.9: Comparativa traducción monótona y no monótona

		BLEU		TER	
		EN → JA	JA → EN	EN → JA	JA → EN
Exp. 1.2	Traducción no monótona	19.49	10.49	0.751	0.812
Exp. 1.3	Traducción monótona	16.66	8.52	0.785	0.830

Como se puede observar en el cuadro 5.9 entre Experimento 1.2 y Experimento 1.3, se observa que la traducción monótona fracasa estrepitosamente, por lo que se llega a la conclusión que lo mejor es dejar que el Moses realice haga las reordenaciones que le permitan sus modelos de distorsión.

Comparativa entre modelos de lenguaje Ahora comprobamos el cambio que produce el cambiar la herramienta para el modelo de lenguaje. La herramienta para los modelos de lenguaje usada en los experimentos originales de la KFTT es el SRILM, pero para nuestros experimentos utilizaremos IRSTLM y Ken-LM (con licencia libre, a diferencia del SRILM).

A continuación vamos a detallar los experimentos, su nomenclatura y configuración:

- Experimento 1.1 (IRSTLM)

  Aquí usaremos KyTea para la segmentación del japonés, IRSTLM para los modelos de lenguaje, 5-gramas y con los modelos de distorsión activados

- Experimento 1.10 (Ken-LM)

  Aquí usaremos KyTea para la segmentación del japonés, Ken-LM para los modelos de lenguaje, 5-gramas y con los modelos de distorsión activados

Cuadro 5.10: Comparativa traducción monótona y no monótona

		BLEU		TER	
		EN → JA	JA → EN	EN → JA	JA → EN
Experimento 1.1	IRSTLM	20.50	10.28	0.749	0.816
Experimento 1.10	Ken-LM	20.03	10.92	0.731	0.866

Como se puede observar en el cuadro 5.10, para en->ja, los modelos de lenguaje realizados con Ken-LM funcionan ligeramente mejor, pero para ja->en ocurre lo contrario.

Comparativa entre tamaños del modelo de lenguaje Ahora comprobaremos como afecta a los resultados cambiar el tamaño de los n-gramas de los modelos de lenguaje. El tamaño usado en la KFTT es de 5-gramas (Referencia 1.2), por lo que queremos comprobar si es el tamaño adecuado o si al aumentar el tamaño podemos ayudar a aliviar el problema de las reordenaciones, que en las frases largas puede superar el tamaño de los n-gramas.

A continuación vamos a detallar los experimentos, su nomenclatura y configuración:

- Experimento 1.5 (3-gramas)

  Aquí usaremos KyTea para la segmentación del japonés, IRSTLM para los modelos de lenguaje, 3-gramas y con los modelos de distorsión activados

- Experimento 1.6 (4-gramas)

  Misma configuración que el experimento 1.5, pero con 4-gramas

- Experimento 1.6 (5-gramas)

  Misma configuración que el experimento 1.5, pero con 5-gramas

- Experimento 1.6 (6-gramas)

  Misma configuración que el experimento 1.5, pero con 6-gramas

- Experimento 1.6 (7-gramas)

  Misma configuración que el experimento 1.5, pero con 7-gramas

Cuadro 5.11: Comparativa traducción monótona y no monótona

		BLEU		**TER**	
		EN → JA	JA → EN	EN → JA	JA → EN
Referencia 1.2	5-GRAMAS	20.32	10.58		
Experimento 1.5	3-GRAMAS	20.46	9.77	0.745	0.811
Experimento 1.6	4-GRAMAS	20.74	10.22	0.741	0.812
Experimento 1.7	5-GRAMAS	21.08	10.33	0.738	0.817
Experimento 1.8	6-GRAMAS	20.87	10.33	0.743	0.817
Experimento 1.9	7-GRAMAS	20.54	10.22	0.739	0.810

Como se puede comprobar en el cuadro 5.11, 5-gramas es lo que mejor resultado da, y que aumentar el tamaño incluso empeora los resultados.

Inglés-Chino

Para estos experimentos se ha utilizado el "Corpus 3 TAUS Inglés-Chino".

Como la meta de estos experimentos es ver la viabilidad de la traducción del inglés-chino con el sistema PangeaMT, hemos utilizado directamente PangeaMT en vez de Moses.

Para estos experimentos, al haber utilizado un corpus privado (por motivos comerciales), no hay referencias con las que poder comparar.

Para solventar el problema de la falta de espacios que tiene el chino, veremos si el uso del segmentador Peterson Segmetor ayuda al entrenamiento del motor, pero primera de todo, comprobamos que cambios puede producir en el corpus 3 el uso del Peterson Segmentor.

Cuadro 5.12: Estadística aplicar o no segmentación en corpus 3 inglés-chino

Sin segmentación		**Con segmentación**	
Palabras	Vocabulario	Palabras	Vocabulario
(EN) 5,8M	(EN) 97k	(EN) 5,8M	(EN) 97k
(ZH) 2,2M	(ZH) 626k	(ZH) 5,8M	(ZH) 137k

Se puede observar en el cuadro 5.11, que tras segmentar con el segmentador Peterson, la cantidad de palabras entre las dos lenguas es ahora muy cercana.

A continuación vamos a detallar los experimentos, su nomenclatura y configuración:

- Experimento 2.1 (Sin segmentación)

  Aquí usaremos PangeaMT, IRSTLM para los modelos de lenguaje, 5-gramas y con los modelos de distorsión activados

- Experimento 2.2 (Con segmentación)

  Misma configuración que el experimento 2.1, pero aplicando en el preproceso del corpus, durante la tokenización, el Peterson Segmentor en el chino.

Cuadro 5.13: Resultados de aplicar segmentación en corpus 3 inglés-chino

		BLEU		**TER**	
		EN → ZH	JA → ZH	EN → ZH	JA → ZH
Experimento 2.1	Sin segmentación	1.53	11.04	5.145	0.869
Experimento 2.2	Con segmentación	39.54	25.65	0.501	0.612

Se puede observar en el cuadro 5.13, que al igual que en japonés, se observa también en el chino, que el uso de segmentadores es obligatorio para tener resultados aceptables, sobretodo cuando se quiera traducir hacia el chino.

5.2.2. Reordenamiento

El japonés, además del problema de no tener espacios entre las palabras o morfemas, tiene un orden gramatical bastante distinto del inglés, sujeto-objeto-verbo (SOV) en vez de sujeto-verbo-objeto (SVO).

El problema de los espacios, tal como vimos en la Sección 5.2.1, se soluciona con el uso, durante la tokenización del japonés, de herramientas como el KyTea o el Mecab.

Por lo que solo nos queda el problema del orden gramatical, que veremos si se puede solucionar con la técnica de niponizar (reordenar la orden gramatical) el inglés que introdujimos en la Sección 4.2, que era haciendo uso del reordenamiento del sistema RBMT de Toshiba. Dicho reordenamiento ha sido aplicado durante el preproceso, antes de aplicar el resto de preproceso (tokenizado, limpieza y paso a minúsculas).

Para estos experimentos se ha utilizado el "Corpus 2 TAUS Inglés-Japonés" (ver cuadro 5.14), y se ha hecho con el sistema Moses.

Cuadro 5.14: Estadísticas corpus 2 al aplicar segmentación

		Segmentos	**Palabras**	**Vocabulario**	**Perplejidad (5-grama)**
Inglés original y niponizado	Training	5,3M	82M	423k	
	Tuning	2k	31k	4,9k	
	Testing	2k	30k	4,9k	90.3451
Japonés original	Training	5,3M	16M	4,9M	
	Tuning	2k	6,1k	4,4k	
	Testing	2k	6,1k	4,4	130.014
Japonés segmentado con Mecab	Training	5,3M	112M	186k	
	Tuning	2k	42k	3,3k	
	Testing	2k	41k	3,3k	44.7675

A continuación vamos a detallar los experimentos, su nomenclatura y configuración:

- Experimento 3.1 (Sin reordenamiento)

  Aquí usaremos Moses, segmentación del japonés con el Mecab, IRSTLM para los modelos de lenguaje, 5-gramas y traducción monótona

- Experimento 3.2 (Con prereordenamiento)

  Misma configuración que el experimento 3.1, pero antes de realizar el preproceso del inglés, se ha aplicado la reordenación con el sistema RBMT de Toshiba, y luego hemos hecho traducción monótona para que el Moses mantenga el reordenado obtenido.

- Experimento 3.3 (Con reordenamiento Moses)

Misma configuración que experimento 3.1, pero en este caso con traducción no monótona, es decir, dejar que Moses haga una reordenación basada en distancias.

Cuadro 5.15: Resultados experimentos con reordenamiento

		BLEU		TER	
		EN → JA	JA → EN	EN → JA	JA → EN
Exp. 3.1	Sin reordenamiento	36.13	25.11	0.495	0.671
Exp. 3.2	Con prereordenamiento	44.02	28.07	0.405	0.563
Exp. 3.3	Con reordenamiento Moses	41.35	26.23	0.431	0.601

Tal como puede observarse en el cuadro 5.15, la conclusión a la que podemos llegar es que la niponización del inglés es realmente efectiva para las traducciones del inglés al japonés, pero para las traducciones del japonés al inglés, la traducción no monótona es mejor que la monótona. No tenemos en cuenta los resultados de la niponización para la traducción del japonés al inglés ya que no tenemos un ningún medio para deshacer la niponización de inglés niponizado.

5.2.3. Detección de valores atípicos

El otro problema que habíamos introducido en la Sección 4.3, es la suciedad o valores atípicos que podemos encontrar en los corpus. En estos experimentos probaremos el conjunto de limpiezas propuestos, las cuales serán aplicados durante el preproceso, antes de aplicar los preprocesos por defecto de Moses:

- Comprobación ortográfica (para lenguas orientales como el coreano, japonés y chino no se podrían comprobar)
- Comprobación de los signos
- Comprobación de segmentos idénticos
- División de frases
- Los segmentos detectados como "sucios" serán excluidos del entrenamiento a riesgo de perder cobertura lingüística.

La detección de valores atípicos se hará sobre el "Corpus 4 privado Inglés-Polaco" y en el "Corpus 5 privado Inglés-Japonés".

Además de evaluar con las métricas de calidad (BLEU y TER), también calcularemos los tiempos de los entrenamientos para ver que además de afectar a la calidad de la traducción, que también afectará a los tiempos de entrenamiento.

Inglés-Polaco

Ahora comprobaremos como afecta la detección de valores atípicos propuesta en el corpus privado recibido para la creación de un motor para un cliente, para el par lingüístico inglés-polaco. El corpus usado es el "Corpus 4 privado Inglés-Polaco" (ver cuadro 5.16).

Cuadro 5.16: Estadísticas corpus 4 Inglés-Polaco al aplicar limpieza

Sin limpieza			**Con limpieza**		
Segmentos	Palabras	Vocabulario	Segmentos	Palabras	Vocabulario
770k	(EN) 8,7M	(EN) 166k	581k	(EN) 6,3M	(EN) 101k
	(PL) 8,3M	(PL) 256k		(PL) 5,9M	(PL) 176k

A continuación vamos a detallar los experimentos, su nomenclatura y configuración:

- Experimento 4.1 (Sin limpieza)

  Aquí usaremos PangeaMT, IRSTLM para los modelos de lenguaje y 5-gramas.

- Experimento 4.2 (Con limpieza)

  Misma configuración que el experimento 4.1, pero antes de realizar el preproceso básico, hemos aplicado la limpieza propuesta.

Cuadro 5.17: Resultados experimentos con limpieza en corpus 4

		BLEU		**TER**		**Tiempo**
		EN → PL	PL → EN	EN → PL	PL → EN	**(minutos)**
Exp. 4.1	Sin limpieza	30.29	37.88	0.607	0.514	548
Exp. 4.2	Con limpieza	30.84	37.82	0.583	0.500	458

Como podemos observar en el cuadro 5.17, a pesar de que el corpus es un 25 % más pequeño, los resultados no se han visto prácticamente afectados, y el tiempo de entrenamiento tras la limpieza se ha reducido en un 17 %.

Inglés-Japonés

Aquí comprobaremos como afecta la detección de valores atípicos propuesta en el corpus privado recibido para la creación de un motor para otro cliente, para el par lingüístico inglés-japonés. El corpus usado es el "Corpus 5 privado Inglés-Japonés" (ver cuadro 5.18).

Cuadro 5.18: Estadísticas corpus 5 al aplicar limpieza

Sin limpieza			**Con limpieza**		
Segmentos	Palabras	Vocabulario	Segmentos	Palabras	Vocabulario
216k	(EN) 2,6M	(EN) 45k	93k	(EN) 1,2M	(EN) 36k
	(JA) 3,6M	(JA) 35k		(JA) 1,7M	(JA) 29k

A continuación vamos a detallar los experimentos, su nomenclatura y configuración:

- Experimento 4.3 (Sin limpieza)

  Aquí usaremos PangeaMT, IRSTLM para los modelos de lenguaje y 5-gramas.

- Experimento 4.4 (Con limpieza)

  Misma configuración que el experimento 4.3, pero antes de realizar el preproceso básico, hemos aplicado la limpieza propuesta.

Cuadro 5.19: Resultados experimentos con limpieza en corpus 5

		BLEU		**TER**		**Tiempo**
		EN → JA	JA → EN	EN → JA	JA → EN	**(minutos)**
Exp. 4.3	Sin limpieza	36.40	26.76	0.560	0.667	274
Exp. 4.4	Con limpieza	33.04	28.08	0.568	0.670	134

En esta ocasión tras aplicar la limpieza (ver cuadro 5.19) y reducir en un 57 % el corpus (y en un 51 % el tiempo de entrenamiento), los resultados en BLEU, en el sentido en→ja, si se han visto afectados, pero en el TER los resultados son prácticamente los mismos.

Esto se debe a que en la limpieza hemos eliminado muchos segmentos que en el japonés había información adicional contenida dentro de paréntesis, que no siempre estaba en otros segmentos con el mismo inglés (ver cuadro 5.20).

Cuadro 5.20: Ejemplos con información adicional

Lengua de origen (EN)	Lengua de destino (JP)	Caso
For more info visit our page.	詳細情報のページをご覧ください。(http://www.xxxx.com)	Error de puntuación en la lengua de destino por el "()"
For more info visit our page.	詳細情報のページをご覧ください。	Sin error de puntuación

5.2.4. Otros modelos

En esta tanda final de experimentos, finalmente comprobaremos si el estado actual del arte, los modelos basados en frases, siguen funcionando mejor que las aproximaciones con modelos factoriales y modelos sintácticos (presentadas en la

Subsección 1.2.3 y Subsección 1.2.4), cuando los usamos entre lenguas tan distantes lingüísticamente como el inglés y el japonés. Estos otros modelos mencionados, son soportados por Moses, por lo que serán interesantes de probar para ver si su adopción en PangeaMT es adecuada.

Con estos modelos podremos introducir información lingüística en los motores, que luego esperamos que el decodificador de Moses pueda aprovechar para inferir reglas de reordenamiento sin tener que recurrir a técnicas de reordenamiento durante el preproceso:

- modelos factoriales basados en frases
- modelos sintácticos
- modelos sintácticos con información lingüística

Los experimentos han sido realizados sobre el "Corpus 1 KFTT inglés-japonés". En las siguientes estadísticas de corpus, ya mostramos en japonés preprocesado con Mecab.

Cuadro 5.21: Estadísticas corpus 1 con segmentación ya aplicada

	Segmentos	**Palabras**	**Vocabulario**	**Fuera de vocabulario**	**Perplejidad (5-grama)**
Training	440M	(EN) 11,5M (JA) 11,4M	(EN) 189k (JA) 81k		
Tuning	1,2k	(EN) 30,8k (JA) 31,5k	(EN) 4,3k (JA) 4,4k	(EN) 607 (JA) 2206	(EN) 157.01 (JA) 253.55
Testing	1,2k	(EN) 26,7k (JA) 26,4k	(EN) 4,4k (JA) 4,1k	(EN) 443 (JA) 1691	(EN) 140.61 (JA) 220.68

También en los experimentos hemos querido observar, además de los cambios en las métricas de calidad, el incremento en tiempo que supone entrenar estos modelos que afectan a la decodificación del Moses.

Modelos factoriales basados en frases

En estos experimentos probaremos los modelos factoriales basados en frases, los cuales aprovechan información lingüística añadida durante el preproceso al corpus de entrenamiento. Además vamos a probar a modificar los "pesos" del modelo de distorsión para ver si así se usa más la información lingüística y se consigue mejores reordenamientos.

Comparativa contra los modelos factoriales A continuación vamos a detallar los experimentos, su nomenclatura y configuración:

- Experimento 5.1 (Modelo basado en frases)

  Aquí volveremos a entrenar un modelo basado en frases usando Moses, usando Mecab para segmentar el japonés, IRSTLM para los modelos de lenguaje, 5-gramas y con traducción no monótona.

- Experimento 5.2 (Modelo factorial)

  Aquí entrenaremos un modelo factorial basado en frases usando Moses, etiquetaremos con POS-tags las palabras de la lengua destino con Mecab (para el japonés) y con MXPOST (para el inglés), usaremos Mecab para segmentar del Japonés, IRSTLM para los modelos de lenguaje, 5-gramas y con traducción no monótona.

Cuadro 5.22: Comparativa modelos factoriales y modelos basados en frases

		BLEU		**TER**		**Tiempo**
		EN → JA	JA → EN	EN → JA	JA → EN	**(minutos)**
Exp. 5.1	Modelo basado en frases	18.98	9.86	0.772	0.820	1372
Exp. 5.2	Modelo factorial con POS-tags	12.96	9.01	0.818	0.848	1584

Como podemos observar en el cuadro 5.22, los modelos factoriales no han ayudado a la traducción. Por lo que podríamos decir que no han sabido aprovechar la información lingüística aportada en el preproceso.

Modificar los pesos del modelo de reordenamiento También hemos probado a variar los pesos con los que tiene en cuenta el modelo de reordenamiento (distorsión), y así ver si se conseguía mejorar los resultados de los modelos factoriales.

Para estas prueba, cuando ya tenemos un motor entrenado, hemos usado el argumento "-d" del Moses, el cual nos permite variar el modo el que se tiene en cuenta el modelo de reordenamiento. Esto lo hemos querido probar a raíz de la recomendación en la web de Moses de probar este argumento.

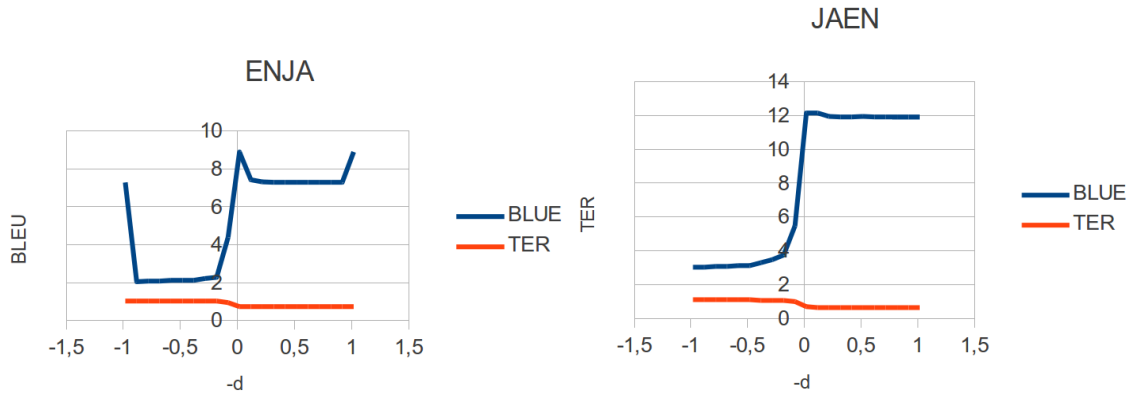
Dicho argumento (cuyo valor por defecto es "1") nos permite meter valores entre -1 y 1, por lo que los experimentos han consistido en usar el motor entrenado en el experimento 5.2 e ir variando el "-d" en incremento de 0.1 entre -1 y 1 (ver cuadro 5.23).

Cuadro 5.23: Comparativa al modificar los pesos del modelo de reordenamiento

		BLEU		TER	
		EN → JA	JA → EN	EN → JA	JA → EN
Exp. 5.2	Sin opción "-d"	12.96	9.01	0.818	0.848
Exp. 5.11	Con opción "-d 0"	12.27	9.00	0.853	0.851
Exp. 5.12	Con opción "-d 1"	12.04	9.01	0.796	0.842
Exp. 5.13	Con opción "-d -1"	3.17	7.4	1.247	1.131

En las siguientes figuras podemos observar los valores de BLEU y TER al modificar el argumento "-d".

Figura 5.5: Comparativa modificar parámetro "-d" en traducción



Como podemos observar en las gráficas, el resultado final de modificar el parámetro "-d" no afectado positivamente el resultado, al contrario, lo ha empeorado.

Modelos sintácticos

En estos experimentos vamos a probar los modelos sintácticos sin añadir ninguna información lingüística adicional en el corpus durante el preproceso; la única diferencia es en el decodificador.

Los modelos sintácticos nos permiten modificar la profundidad del árbol ("max_chart_span") que queremos entrenar.

En estos experimentos modificaremos la profundidad de los árboles, y además de medir la calidad, también mediremos los tiempos que implican entrenar este tipo de modelos que afectan a la decodificación.

A continuación vamos a detallar los experimentos, su nomenclatura y configuración:

- Experimento 5.1 (Modelo basado en frases)

  Este experimento basado en frases lo utilizaremos de referencia para compararlo con los modelos sintácticos. Está entrenado con Moses, usando Mecab para segmentar el japonés, IRSTLM para los modelos de lenguaje, 5-gramas y con traducción no monótona.

- Experimento 5.4 (Modelo sintáctico con 10 de profundidad)

  Aquí entrenaremos un modelo sintáctico usando Moses, con profundidad de árbol de 10 (el cual es el valor por defecto), usaremos Mecab para segmentar del Japonés, IRSTLM para los modelos de lenguaje, 5-gramas y con traducción no monótona.

- Experimento 5.5 (Modelo sintáctico con 20 de profundidad)

  Basado en el experimento 5.4 pero creando un árbol de profundidad 20.

- Experimento 5.6 (Modelo sintáctico con 50 de profundidad)

  Basado en el experimento 5.4 pero creando un árbol de profundidad 50.

- Experimento 5.7 (Modelo sintáctico con 100 de profundidad)

  Basado en el experimento 5.4 pero creando un árbol de profundidad 100.

Cuadro 5.24: Comparativa modelos sintácticos

		BLEU		**TER**		**Tiempo (minutos)**
		EN → JA	JA → EN	EN → JA	JA → EN	
Exp. 5.1	Modelo basado en frases	18.98	9.86	0.772	0.820	1372
Exp. 5.4	Modelo sintáctico de tamaño 10	19.59	11.19	0.746	0.825	2329
Exp. 5.5	Modelo sintáctico de tamaño 20	19.43	11.11	0.770	0.815	3098
Exp. 5.6	Modelo sintáctico de tamaño 50	19.53	10.94	0.752	0.820	2528
Exp. 5.7	Modelo sintáctico de tamaño 100	19.53	11.18	0.772	0.815	4322

Tal como podemos observar en el cuadro 5.24 una profundidad de 10 basta para mejorar los resultados, pero a costa de un incremento en el tiempo de un 59 %. El incrementar el tamaño del árbol además de no mejorar los resultado, hace que los tiempos de entrenamiento se dupliquen o tripliquen, haciendo poco viable crear árboles de gran tamaño.

Modelos sintácticos con información lingüística

En estos experimentos vamos a probar los modelos sintácticos en los que añadiremos información lingüística adicional en el corpus durante el preproceso. Por lo que tanto el preproceso como la decodificación se han visto afectados.

La información sintáctica solo se ha podido añadir al inglés (al japonés no hemos podido ya que no hemos encontrado ningún parseador sintáctico del japonés), por lo que los entrenamientos solo serán en el sentido en→ja.

A continuación vamos a detallar los experimentos, su nomenclatura y configuración:

- Experimento 5.4 (Modelo sintáctico con 10 de profundidad)

  Utilizaremos este experimento de modelo sintáctico normal (de la Sección 5.2.4) y lo utilizaremos de referencia para compararlo con los modelos sintácticos que tienen información lingüística en el preproceso. Está entrenado con Moses, con profundidad de árbol de 10, usando Mecab para segmentar el japonés, IRSTLM para los modelos de lenguaje, 5-gramas y con traducción no monótona.

- Experimento 5.9 (Modelo sintáctico con 10 de profundidad y información sintáctica)

  Basado en el experimento 5.4 pero añadiendo información sintáctica del inglés.

Cuadro 5.25: Comparativa modelos sintácticos con información sintáctica

		BLEU	**TER**
		EN → JA	EN → JA
Exp. 5.4	Modelo sintáctico de tamaño 10	19.59	0.746
Exp. 5.9	Modelo sintáctico de tamaño 10 e inf. sintáctica	13.34	0.812

Como podemos observar en el cuadro 5.25, el incluir información lingüística en el preproceso no solo no ha ayudado a los modelos sintácticos, sino que ha hecho que den peores resultados.

6 Conclusiones

En esta tesis de máster se han visto y comparado distintas aproximaciones para solventar distintos problemas (falta de espacios entre las palabras, orden gramatical distinto y suciedad de los datos de entrenamiento) que surgen en la creación de motores de traducción automática entre lenguas distantes.

Por lo que hemos visto en los experimentos, es difícil conseguir buena calidad al traducir automáticamente de lenguas sencillas lingüísticamente (morfológicamente o gramaticalmente) a lenguas más complejas lingüísticamente, por ejemplo de inglés a polaco, de japonés a inglés o de chino a inglés. Esto ocurre porque en la lengua más compleja se tiene que inferir información que no está en la lengua menos rica. El camino contrario es mas sencillo ya que se puede perder información lingüística sin que suponga una pérdida de calidad.

Podríamos llegar a la conclusión que traducir de lenguas analíticas o aislantes (chino o inglés) hacia lenguas sintéticas (japonés o polaco) es más difícil que el camino inverso. Y traducir de lenguas sintéticas fusionantes (polaco) hacia lenguas sintéticas aglutinantes (japonés) o lenguas analíticas (chino o inglés) es más sencillo que el camino contrario.

Uno de los problemas vistos, es que hay lenguas, como el chino y el japonés, que no usan espacios entre las palabras o morfemas, lo cual hace imposible trabajar con ellos directamente, ya que al alinear entre el inglés y dichas lenguas, acabaremos alineando palabras con frases enteras. Esto lo solucionamos tokenizando durante preproceso con las herramientas de análisis morfológica vistas, Mecab y Kytea. Con dichas herramientas insertaremos los espacios necesarios entre las palabras o morfemas, dejando casi la misma cantidad de tokens entre el inglés y dichas lenguas, lo cual favorece una buena alineación.

Otro problema visto, ha sido que hay lenguas, como el japonés y el inglés, que entre ellas tienen un orden gramatical bastante distinto, lo cual supone un gran problema para los sistemas de reordenamiento utilizados en sistemas estadísticos como Moses. Dicho problema no se soluciona ni aumentando el tamaño de los n-gramas. Esto lo solucionamos con la hibridación propuesta, que ya fue presentada en el artículo [YE11] de la revista AAMT, y que consiste en reordenar el inglés (niponización) durante preproceso con el sistema basado en reglas de Toshiba, y luego creando el motor con el sistema estadístico basado en frases de PangeaMT. Podemos decir que es realmente muy útil, particularmente en el sentido inglés a japonés ya que la alineación durante el entrenamiento del motor estadístico acaba siendo uno a uno y

esto acaba solventando el problema que tiene el reordenamiento basado en distancias de Moses. Entre lenguas como el chino y el inglés no haría falta dicha reordenación debido a lo parecido que son las estructuras gramaticales de las lenguas.

El último problema visto, ha sido la gran cantidad de valores atípicos que se puede encontrar en los datos de entrenamiento obtenidos a partir de las memorias de traducción de la industria de traducción profesional. La relevancia de dicha detección o limpieza ya fue presentada en la las jornadas de la JTF de Tokyo en noviembre del 2011[1][2] y en el forum "TAUS Tokyo Executive Forum" en abril del 2013[3]. La limpieza por defecto que tiene Moses no se puede solucionar dicho problema, pero con la limpieza propuesta, que tiene más factores en cuenta, además de reducir drásticamente los tiempos de computación, se puede llegar incluso a mejorar la calidad de la traducción y además es independientes de los idiomas tratados.

Por otra parte, los modelos factoriales y sintácticos no son una solución viable para las lenguas distantes. Los modelos sintácticos a pesar que consiguen mejorar los resultados del estado del arte (modelos basados en frases), tienen la contrapartida del incremento bastante significativo del tiempo de entrenamiento, además de requerir mucha más memoria RAM y espacio de disco. Por contrapartida, los modelos factoriales no solo no mejoran los resultados, sino que los empeoran.

[1]http://www.pangeanic.com/news/2011/pangeamt-syntax-based-hybrid-presentation-at-jtf.html
[2]http://www.slideshare.net/manuelherranz/jtf-new
[3]http://www.slideshare.net/manuelherranz/tms-days-04-2012-manuel-herranz-pangea-mt

7 Futuro trabajo

En este capítulo presentamos las futuras direcciones hacia donde se desarrollaran algunos de los puntos presentados en esta tesis de máster.

- Explorar otros sistemas que realicen prereordenación

  Tal como hemos visto la reordenación durante el preproceso del japonés con el sistema de Toshiba ayuda en gran medida a la traducción automática estadística, por lo que suponemos que otras lenguas con problemas similares, como el alemán o el árabe, podrían beneficiarse de aplicar un prereordenamiento similar.

- Realización de un sistema de prereordenamiento propio del japonés

  Durante el reordenamiento del inglés con el sistema de Toshiba, se ha observado que frases de gran complejidad (como oraciones subjuntivas o condicionales que en inglés pueden construirse de distintas manera pero que en japonés solo hay una manera) daban problemas al reordenamiento, por lo cual, se empezó a desarrollar un sistema de reordenamiento propio para lidiar con este tipo de frases. El sistema tiene dos formas de funcionamiento, reordenamiento tras hacer análisis sintáctico y reordenamiento tras hacer análisis gramatical. El primero más eficaz pero más lento, y el segundo más limitado pero más rápido. Hasta el momento, el sistema contiene más de 20 reglas lingüísticas de reordenamiento, pero aún siguen sin ser suficientes para mejorar los resultados, y para conseguir más reglas lingüísticas se requiere un gran esfuerzo para el que hacen falta grandes conocimientos lingüísticos del inglés y del japonés. Dicho sistema fue parcialmente incluido en PangeaMT en una de sus últimas versiones [YHH+12].

- Detección de valores atípicos

  Tal como se ha visto la detección y exclusión de valores atípicos puede llegar a mejorar los resultados, pero no siempre, ya que si se filtran demasiados segmentos se puede llegar a quedarse corto de datos de entrenamiento y entonces perder cobertura lingüística. Por lo que una mejor detección sería conveniente.

Bibliografía

[AOCBFZ+07] Carme Armentano Oller, Antonio Miguel Corbí Bellot, Mikel L Forcada Zubizarreta, Mireia Ginestí Rosell, Marco A Montava Belda, Sergio Ortiz Rojas, Juan Antonio Pérez Ortiz, Gema Ramírez Sánchez, Felipe Sánchez Martínez, et al. Apertium, una plataforma de código abierto para el desarrollo de sistemas de traducción automática. 2007.

[AOP06] Yaser Al-Onaizan and Kishore Papineni. Distortion models for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 529–536. Association for Computational Linguistics, 2006.

[AT03] Juan A Alonso and Gregor Thurmair. The comprendium translator system. In *Proceedings of the Ninth Machine Translation Summit*, 2003.

[AU69] Alfred V. Aho and Jeffrey D. Ullman. Syntax directed translations and the pushdown assembler. *Journal of Computer and System Sciences*, 3(1):37–56, 1969.

[Axe06] Amittai Axelrod. Factored language model for statistical machine translation. *Master of Science Thesis*, 2006.

[BCP+90] Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Fredrick Jelinek, John D Lafferty, Robert L Mercer, and Paul S Roossin. A statistical approach to machine translation. *Computational linguistics*, 16(2):79–85, 1990.

[BK03] Jeff A Bilmes and Katrin Kirchhoff. Factored language models and generalized parallel backoff. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume of the Proceedings of HLT-NAACL 2003–short papers-Volume 2*, pages 4–6. Association for Computational Linguistics, 2003.

[Chi05] David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270. Association for Computational Linguistics, 2005.

[CKK05] Michael Collins, Philipp Koehn, and Ivona Kučerová. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 531–540. Association for Computational Linguistics, 2005.

[Eis07] Andreas Eisele. Hybrid machine translation: Combining rule-based and statistical mt systems. *First Machine Translation Marathon*, pages 16–20, 2007.

[Elm08] Jakob Elming. Syntactic reordering integrated with phrase-based smt. In *Proceedings of the Second Workshop on Syntax and Structure in Statistical Translation*, pages 46–54. Association for Computational Linguistics, 2008.

[GGK+06] Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 961–968. Association for Computational Linguistics, 2006.

[GHKM04] Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. What's in a translation rule? In *HLT-NAACL*, pages 273–280, 2004.

[GM08] Michel Galley and Christopher D Manning. A simple and effective hierarchical phrase reordering model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 848–856. Association for Computational Linguistics, 2008.

[Hab07] Nizar Habash. Syntactic preprocessing for statistical machine translation. *Proceedings of the 11th MT Summit*, 2007.

[HC07] Liang Huang and David Chiang. Forest rescoring: Faster decoding with integrated language models. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 45, page 144, 2007.

[KAM+05] Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. Edinburgh system description for the 2005 iwslt speech translation evaluation. In *International Workshop on Spoken Language Translation*, 2005.

[KHB+07] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics, 2007.

[KOM03] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics, 2003.

[LCBD+09] Zhifei Li, Chris Callison-Burch, Chris Dyer, Juri Ganitkevitch, Sanjeev Khudanpur, Lane Schwartz, Wren NG Thornton, Jonathan Weese, and Omar F Zaidan. Joshua: An open source toolkit for parsing-based machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 135–139. Association for Computational Linguistics, 2009.

[LLZ+07] Chi-Ho Li, Minghui Li, Dongdong Zhang, Mu Li, Ming Zhou, and Yi Guan. A probabilistic approach to syntax-based reordering for statistical machine translation. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 45, page 720, 2007.

[Nag84] Makoto Nagao. A framework of a mechanical translation between japanese and english by analogy principle. 1984.

[Neu11] Graham Neubig. The kyoto free translation task. *Available on line at http://www. phontron. com/kftt*, 2011.

[Och02] Franz Josef Och. *Statistical machine translation: from single-word models to alignment templates*. PhD thesis, Bibliothek der RWTH Aachen, 2002.

[ON03] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51, 2003.

[ON04] Franz Josef Och and Hermann Ney. The alignment template approach to statistical machine translation. *Computational linguistics*, 30(4):417–449, 2004.

[PRWZ02] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[QMC05] Chris Quirk, Arul Menezes, and Colin Cherry. Dependency treelet translation: Syntactically informed phrasal smt. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 271–279. Association for Computational Linguistics, 2005.

[Sad89] Victor Sadler. Working with analogical semantics: disambiguation techniques in dlt. 1989.

[SDS+06] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, pages 223–231, 2006.

[Til04] Christoph Tillmann. A block orientation model for statistical machine translation. In *Proceedings of HLT-NAACL*, 2004.

[WCK07] Chao Wang, Michael Collins, and Philipp Koehn. Chinese syntactic reordering for statistical machine translation. In *EMNLP-CoNLL*, pages 737–745, 2007.

[Wu97] Dekai Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational linguistics*, 23(3):377–403, 1997.

[XKRO09] Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. Using a dependency parser to improve smt for subject-object-verb languages. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, pages 245–253. Association for Computational Linguistics, 2009.

[XLL06] Deyi Xiong, Qun Liu, and Shouxun Lin. Maximum entropy based phrase reordering model for statistical machine translation. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 44, page 521, 2006.

[XM04] Fei Xia and Michael McCord. Improving a statistical mt system with automatically learned rewrite patterns. In *Proceedings of the 20th international conference on Computational Linguistics*, page 508. Association for Computational Linguistics, 2004.

[YE11] Helle A. Suzuki H. Yuste E., Herranz M. Going hybrid: Pangeanic's and toshiba's first steps toward enjp mt hybridization. *In AAMT Journal No.50 Nov. 2011*, 2011.

[YHH+12] Elia Yuste, Manuel Herranz, Alexandre Helle, Antonio L. Lagarda, Mercedes Garcia-Martinez, Jeronimo Pla-Civera, Maria Blasco, Antonio Morella, and Jordi Mallach. Pangeanic's do-it-yourself machine translation: User empowerment and user-driven mt processing. *Asia-Pacific Association for Machine Translation Journal*, (52):36–50, 2012.

[YHL+10] Elia Yuste, Manuel Herranz, Antonio L. Lagarda, Lionel Tarazon, Isaias Sanchez-Cortina, and Francisco Casacuberta. Pangeamt - putting open standards to work... well. In *Proceedings of The Ninth Conference of the Association for Machine Translation in the Americas (AMTA2010)*. 2010. http://www.mt-archive.info/AMTA-2010-Yuste.pdf.

[YK01] Kenji Yamada and Kevin Knight. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 523–530. Association for Computational Linguistics, 2001.

[ZN06] Richard Zens and Hermann Ney. Discriminative reordering models for statistical machine translation. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 55–63. Association for Computational Linguistics, 2006.

[ZV04] Ying Zhang and Stephan Vogel. Measuring confidence intervals for mt evaluation metrics. *TMI 2004*, 2004.

[ZV06] Andreas Zollmann and Ashish Venugopal. Syntax augmented machine translation via chart parsing. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 138–141. Association for Computational Linguistics, 2006.

[ZVOP08] Andreas Zollmann, Ashish Venugopal, Franz Och, and Jay Ponte. A systematic comparison of phrase-based, hierarchical and syntax-augmented statistical mt. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 1145–1152. Association for Computational Linguistics, 2008.

[ZZN07] Yuqi Zhang, Richard Zens, and Hermann Ney. Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation. In *Proceedings of the NAACL-HLT 2007/AMTA Workshop on Syntax and Structure in Statistical Translation*, pages 1–8. Association for Computational Linguistics, 2007.

Abreviaciones

API	Application Programming Interface
BBDD	Base de Datos
BLEU	Bilingual Evaluation Understudy
CAT	Computer Assisted Translation
CKY	Cocke-Younger-Kasami
DNT	Do Not Translate
EBMT	Example Based Machine Translation
FTP	File Transfer Protocol
GPL	General Public License
HLT	Human Language Technology
HMT	Hybrid Machine Translation
HTML	HyperText Markup Language
KyTea	Kyoto Text Analysis
LGPL	Lesser General Public License
LSP	Language Service Provider
MBMT	Memory Based Machine Translation
MT	Machine Translation
POS	Part Of Speech
RAM	Random Access Memory
RBMT	Rule Based Machine Translation
REST	Representational state transfer

RTF	Rich Text Format
SCFG	Stochastic context-free grammar
SDLXLIFF	SDL XML-based Localization Interchange File Format
SDTS	Syntax-Directed Translation Schemes
SMT	Statistical Machine Translation
SOAP	Simple Object Access Protocol
SOV	Sujeto-Objeto-Verbo
SVM	Support Vector Machines
SVO	Sujeto-Verbo-Objeto
TER	Translation Edit Rate
TM	Translation Memory
TMX	Translation Memory eXchange
TU	Translation unit
XLIFF	XML Localisation Interchange File Format
XML	eXtensible Markup Language

Índice de figuras

Índice de cuadros