Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València

# Traducción Automática Interactiva Basada en Segmentos de Palabras

Trabajo Fin de Máster

**Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital**

**Autor**: Torres Badia, Guillem

**Tutores**: Casacuberta Nolla, Francisco
José Miguel Benedí Ruiz

2016/2017

# Resumen

Los sistemas actuales de traducción interactiva están basados en la validación/corrección por parte del humano de sucesivos prefijos de las traducciones y en la generación de los correspondientes sufijos por parte del traductor automático. Esta aproximación tiene el inconveniente de que requiere demasiado esfuerzo por parte del usuario, superando al requerido en enfoques no interactivos

En este trabajo se propone una implementación eficiente de un sistemas de traducción interactivo-predictivo en el que el humano valida todos los segmentos que desee de las traducciones generadas por el traductor automático e introduce una corrección donde crea oportuno y el sistema debe rellenar con nuevas sugerencias los segmentos no validados por el humano.

Esta implementación será validada mediante una serie de experimentos en varias tareas de traducción.

**Palabras clave:** traducción automática; traducción interactiva; modelos jerárquicos; hipergrafos

# Abstract

Current interactive machine translation systems are based on the validation / correction by a human of the successive prefixes of the translations and the generation of the corresponding suffixes by the machine translation systems. This approach has the disadvantage that requires more user effort than non-interactive approaches.

This work presents an efficient implementation of an interactive-predictive machine translation system in which the human validates all desired segments of the automatic generated translations and introduces a correction, the system must fill with new suggestions the segments not validated by the human.

This implementation will be validated through a series of experiments in various translation tasks.

**Keywords:** machine translation, interactive machine translation, hierarchical models, hypergraphs

# Resum

Els sistemes actuals de traducció interactiva estan basats en la validació/correcció per part d'un humà dels successius prefixes de les traduccions i en la generació dels corresponents sufixes per part del traductor automàtic. Aquesta aproximació té l'inconvenient de que requereix massa esforç per part de l'usuari, superant a aproximacions no interactives.

En aquest treball, es proposa una implementació eficient d'un sistema de traducció interactiva-predictiva en el que l'humà valida tots els segments que desitge de les traduccions generades pel traductor automàtic i introdueix una correcció on crega oportú, de manera que el sistema deurà emplenar amb nous suggeriments els segments no validats per l'humà.

Aquesta implementació serà validada mitjançant una sèrie d'experiments en diverses tasques de traducció.

**Paraules clau:** traducció automàtica; traducció interactiva; models jeràrquics; hipergrafs

# Agradecimientos

A los Profesores Francisco Casacuberta y José Miguel Benedí, por brindarme la oportunidad de trabajar con ellos.

Al Dr. Jesús González, por comenzar este proyecto y por todo el apoyo que me ha prestado para poder continuarlo.

A mis compañeros del PRHLT, especialmente a Miguel, por su inestimable ayuda.

A mi familia, por la paciencia.

# Índice de contenidos

# Índice de ilustraciones

# Índice de tablas

# Perspectiva general

En esta tesis de máster presentamos un sistema de Traducción Automática Interactiva que presenta un protocolo de interacción basado en segmentos de palabras. Con ello, pretendemos reducir el esfuerzo humano requerido por otros sistemas interactivos, como por ejemplo los basados en validación de prefijos. La estructura del documento es la siguiente:

- El *Capítulo 1* resume el estado del arte de la Traducción Automática, destacando los paradigmas en los que se enmarca nuestro sistema.

- En el *Capítulo 2* se describe con detalle el protocolo de interacción de nuestro sistema. A continuación, se exponen sus bases teóricas, en lo que respecta al modelo estadístico y al algoritmo de búsqueda de la traducción óptima.

- En el *Capítulo 3* se detalla la experimentación llevada a cabo para evaluar el desempeño del sistema y se analizan los resultados obtenidos.

- Por último, en el *Capítulo 4* exponemos las conclusiones globales y se proponen futuras mejoras a nuestro sistema.

# Capítulo 1. Introducción

El lenguaje es una potente herramienta de comunicación que ha sido determinante para el desarrollo social, cultural y tecnológico del ser humano. Estas cualidades se potenciaron aún más con el surgimiento del lenguaje escrito, que facilitó la comunicación entre personas de diferentes procedencias y permitió la transmisión de conocimiento entre generaciones de forma mucho más precisa que la tradición oral.

No obstante, la inmensa variedad de idiomas que existen –se estima que hay 7015 lenguas vivas en la actualidad– impide la comunicación en ausencia de una *lingua franca*. Por ello, surge la necesidad de traducir los textos entre diferentes idiomas, tarea que, por otra parte, requiere una gran cantidad de esfuerzo humano y recursos económicos. Estas limitaciones se han acentuado en la actualidad, ya que si bien existe una lengua común muy extendida, el volumen de información que se genera diariamente es inabarcable con los medios tradicionales.

Para suplir estos inconvenientes, surge la Traducción Automática o *Machine Translation* (de ahora en adelante MT), que pretende automatizar el proceso de traducción mediante el uso de computadores. Esta tarea se ha abordado desde diversos enfoques, tanto en lo que respecta a la forma de interactuar con el usuario como a la tecnología y fundamentos matemáticos empleados.

Como apunte, la MT no se aplica a textos literarios, ya que la traducción de estos requiere conservar la belleza del texto original en la medida de lo posible, cosa que no es posible realizar de forma automática. Por ello, la MT se limita a textos científicos, políticos, técnicos y similares.

## 1.1 Traducción Automática

Si bien el origen de la MT es un poco difuso, hemos optado por identificarlo con la patente de la primera máquina traductora, de acuerdo con Hutchins (1995). Ésta fue realizada en los años 30 por George Artsrouni y consistía en un diccionario bilingüe basado en cintas perforadas. Otro sistema de traducción más complejo fue el realizado por el ruso Peter Troyanskii, empleaba tanto un diccionario como una referencia gramatical que permitía operar entre diferentes lenguajes. No obstante, este sistema no fue descubierto hasta finales de los años 50, cuando ya existían sistemas de traducción basados en computadoras.

El primero de estos fue el *Translation Memorandum*, realizado por Warren Weaver (*Rockefeller Foundation*) en 1949 y que estaba basado en las máquinas criptográficas de la Segunda Guerra Mundial. A este le siguieron múltiples propuestas que surgieron en diferentes universidades de EEUU, así como una pequeña demostración pública llamada *Georgetown-IBM Experiment* (1954).

No obstante, estos sistemas estaban basados en diccionarios bilingües y reglas escritas a mano, lo que los hacía difíciles de mantener y les impedía tratar adecuadamente con ambigüedades. Estas limitaciones fueron señaladas por ALPAC

en 1964, lo que, a falta de un paradigma más fiable, hizo que las investigaciones se abandonaran casi por completo.

Si bien los adelantos tecnológicos permitieron retomar la actividad en este campo a principios de los 80, los sistemas seguían estando basados en reglas morfológicas, sintácticas y semánticas. No fue hasta finales de esta década que aparecieron métodos como la MT estadística o la MT basada en ejemplos, que empleaban la manipulación de grandes corpus en lugar de reglas escritas mano.

Estos nuevos enfoques, junto con el avance en campos similares (como reconocimiento y síntesis del habla) y la aparición de computadoras más baratas y potentes supuso un *boom* de la MT en los años 90. Tanto es así que múltiples empresas vieron un nicho en este sector, como *Globalink*, *Microtac*, *AltaVista* o *Google*.

En la actualidad existen múltiples frentes de investigación abiertos, tanto en lo que respecta a la tecnología empleada como al ámbito de aplicación.

Respecto a lo primero, podemos destacar las técnicas híbridas –que combinan la MT estadística y la basada en reglas– o el uso de redes neuronales en MT –unas estructuras de nodos por capas que son de gran utilidad en *machine learning* estadístico. Otro hecho destacable es la creación de *Moses* (Kohen, et al., 2007), un *software* de MT estadística de código abierto y gratuito.

También se ha encontrado interés en nuevos ámbitos de aplicación, como ámbitos parlamentarios (que requieren traducción del habla en tiempo real) o Internet (que requiere traducción multi-lenguaje).

### 1.1.1 Traducción Automática Basada en Reglas

Como se ha mencionado con anterioridad, la MT Basada en Reglas o *Rule-Based MT* (de ahora en adelante, RBMT) fue la primera que surgió históricamente. Como su nombre indica, se basa en grandes colecciones de reglas escritas a mano por humanos. Por supuesto, esta definición es muy amplia y abarca varios tipos:

- **RBMT basada en diccionarios bilingües**. Es el enfoque más simplista, ya que solo captura información léxica. No obstante, puede combinarse con otros paradigmas algo más complejos.
- **RBMT mediante transferencia.** Captura la información del texto original y lo traduce al lenguaje objetivo, empleando reglas que relacionan ambos idiomas. Puede hacerse de forma más superficial (nivel sintáctico) o más profunda (nivel semántico).
- **RBMT mediante interlingua.** Es muy similar al RBMT mediante transferencia, solo que es completamente independiente del par de lenguas involucradas. Es el más flexible de los dos, pero también presenta un peor desempeño.

La principal limitación de estos sistemas es que, al estar escritos a mano, requieren un gran esfuerzo por parte del desarrollador, además de ser poco flexibles. Por ello, en la actualidad se han visto desplazados por otros enfoques más adaptables.

### 1.1.2 Traducción Automática Basada en Corpus

La Traducción Automática Basada en Corpus surge para poner fin a los problemas derivados de las variantes basadas en reglas. Este enfoque emplea un corpus de entrenamiento consistente en frases traducidas en ambos lenguajes para aprender a realizar el proceso de traducción sin necesidad de reglas explícitas. Los tipos de MT Basada en Corpus más importantes son la MT Basada en Ejemplos y la MT Estadística.

Inspirada en el proceso de traducción humana, la MT Basada en Ejemplos emplea el corpus de entrenamiento para establecer analogías entre las construcciones semánticas de ambos idiomas. Generalmente, el proceso de traducción tiene dos fases: encontrar una hipótesis (construcción) similar en el corpus de entrenamiento y recombinar la hipótesis en el lenguaje objetivo para generar la traducción.

La MT Estadística, por su parte, emplea el corpus de entrenamiento para entrenar un modelo estadístico que servirá para estimar la traducción más probable para cada oración en concreto. Puesto que el sistema que vamos a desarrollar se clasifica dentro de este paradigma, será explicado con más detalle en la sección 1.2.

## 1.2 Traducción Automática Estadística

La MT Estadística se basa en la idea de que el problema de traducción no es un problema cerrado, en el sentido de que una oración puede tener varias posibles traducciones. Asociando una determinada probabilidad a cada una ellas según su verosimilitud, el problema de la MT Estadística se puede entender como la búsqueda de la hipótesis de traducción más probable.

Formalmente, dada la frase origen $s$, el objetivo es encontrar una traducción $\hat{t}$ que maximice la probabilidad *a posteriori*:

$$\hat{\boldsymbol{t}} = \arg\max_{\boldsymbol{t}} P(\boldsymbol{t}|\boldsymbol{s}) \tag{1}$$

Aplicando la regla de Bayes, esta expresión puede reformularse como:

$$\hat{\boldsymbol{t}} = \arg\max_{\boldsymbol{t}} P(\boldsymbol{t}) \cdot P(\boldsymbol{s}|\boldsymbol{t}) \tag{2}$$

Donde *probabilidad del modelo de lenguaje* $P(\boldsymbol{t})$ representa la verosimilitud *a priori* de $t$, mientras que la *probabilidad del modelo de traducción* $P(\boldsymbol{s}|\boldsymbol{t})$ representa la relación entre la frase origen y su traducción.

De esta forma, los principales retos a la hora de desarrollar un sistema de MT Estadística son estimar las probabilidades mencionadas anteriormente y desarrollar un sistema de búsqueda del valor óptimo.

Para lo primero, es habitual estimar ambas probabilidades con un modelo log-lineal, que combina $N$ características importantes para la traducción, representadas por $f_n(\boldsymbol{t}, \boldsymbol{s})$ y ponderadas con el peso $\lambda_n$:

$$\hat{\boldsymbol{t}} = \arg\max_{\boldsymbol{t}} \frac{\exp \sum_{n=1}^{N} \lambda_n \cdot \log(f_n(\boldsymbol{t}, \boldsymbol{s}))}{\sum_{\boldsymbol{t}'} \exp \sum_{n=1}^{N} \lambda_n \cdot \log(f_n(\boldsymbol{t}', \boldsymbol{s}))} = \arg\max_{\boldsymbol{t}} \left\{ \sum_{n=1}^{N} \lambda_n \cdot \log(f_n(\boldsymbol{t}, \boldsymbol{s})) \right\} \quad (3)$$

Algunas características $f_n(\boldsymbol{t}, \boldsymbol{s})$ que suelen emplearse en sistemas de traducción, (como por ejemplo *Moses*) son, entre otras:

- Probabilidad del modelo de lenguaje ($P(\boldsymbol{t})$), sobre un modelo de n-gramas.
- Probabilidad del modelo de frases bilingües ($P_{PB}(\boldsymbol{t}|\boldsymbol{s})$). Basado en un alineamiento entre las frases de la fuente $\boldsymbol{s}$ con las de la traducción $\boldsymbol{t}$.
- Probabilidad del modelo de frases bilingües inverso ($P_{PB}(\boldsymbol{s}|\boldsymbol{t})$). Similar a la anterior, pero en sentido inverso.
- Probabilidad del modelo de frases bilingües lexicalizado ($P_{LEX}(\boldsymbol{t}|\boldsymbol{s})$). Basado un alineamiento entre las palabras de la fuente $\boldsymbol{s}$ con las de la traducción $\boldsymbol{t}$.
- Probabilidad del modelo de frases bilingües lexicalizado inverso ($P_{LEX}(\boldsymbol{s}|\boldsymbol{t})$). Similar a la anterior, pero en sentido inverso.

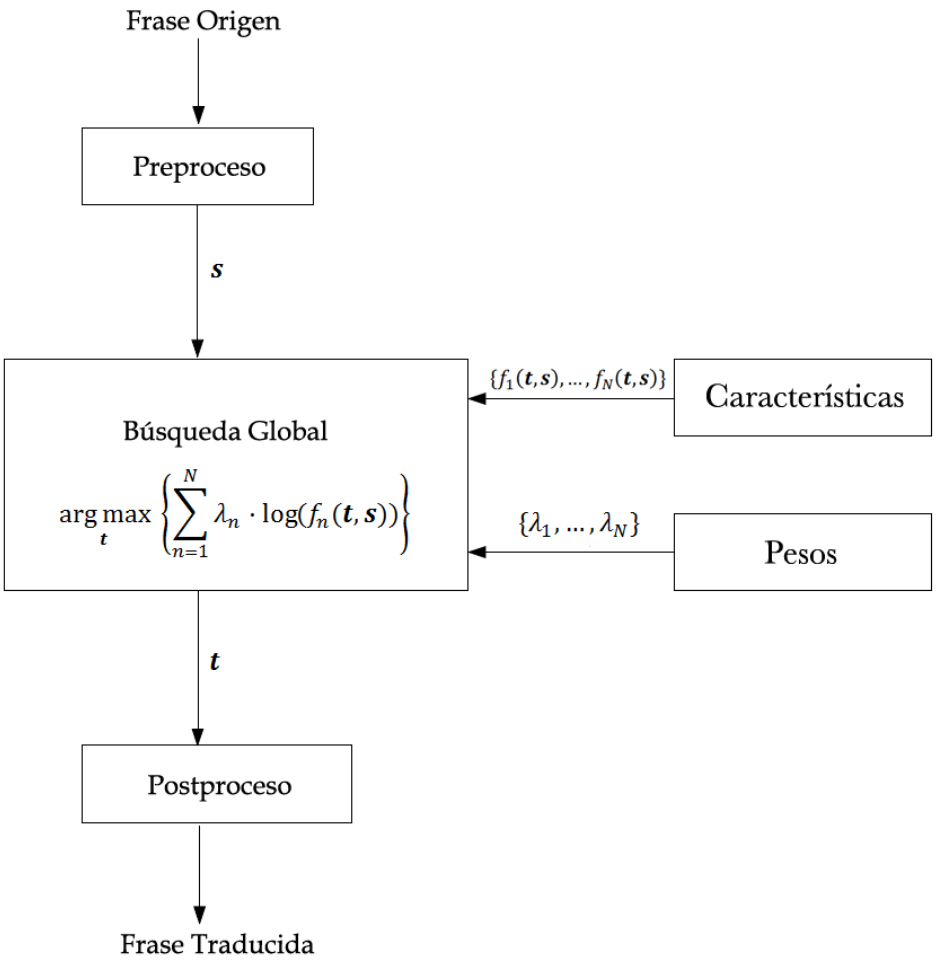


**Ilustración 1.** Arquitectura de un sistema de MT basado en un modelo log-lineal. Adaptación de una ilustración extraída de Zens, et al. (2002).

Aunque en la práctica hemos empleado un modelo log-lineal, todas las ecuaciones de este documento se expresan en términos de probabilidades (de forma similar a las Ecuaciones 1 y 2), ya que consideramos que de esta manera se entiende con más claridad.

Para la búsqueda de la hipótesis más probable pueden emplearse diversas técnicas de acuerdo con el paradigma de MT empleado: alineamientos a nivel de palabra o de frase, Redes Neuronales Artificiales, grafos de palabras o gramáticas, entre otros.

### 1.2.1 Traducción Automática Jerárquica

Aunque el objetivo de la MT Estadística es encontrar la traducción más probable para una oración completa, no es posible hacer esto de forma fiable tratando la oración como una construcción monolítica. Por ello, surgieron varios enfoques según cuál era la unidad en la que se dividían las oraciones, siendo la primera de ellas la traducción palabra a palabra.

Este paradigma pronto demostró no ser adecuado, dando lugar a la traducción basada en frases –entendiendo *frases* como las construcciones semánticas de las que se componen las oraciones. A grandes rasgos, este paradigma se caracterizaba por seguir un proceso generativo de tres pasos: dividir la oración original en fragmentos o frases, traducir dichas frases individualmente y reordenarlas. Si bien este enfoque era adecuado para pares de lenguas similares, también demostró estar muy limitado para lenguajes diferentes entre sí, como el chino y el inglés.

Para solucionar este problema surge la traducción jerárquica (Chiang, 2005), en el que las frases podían poseer segmentos que debían ser reemplazados por subfrases. Por ejemplo:

$$yu\ X_1\ you\ X_2 \rightarrow have\ X_2\ with\ X_1$$

En el que $X_1$ y $X_2$ son símbolos no-terminales que deben sustituirse por frases, que a su vez pueden contener más no-terminales. Como puede verse, éste enfoque posee una mayor potencia expresiva que el basado en frases, ya que permite el reordenamiento dentro de las propias frases y no sólo entre ellas.

Para formalizar este modelo, se emplean Gramáticas Incontextuales (Aho & Ullmann, 1969). Más concretamente, se emplean Gramáticas Incontextuales Síncronas o SynCFG. En ellas, cada correspondencia entre dos frases se expresa mediante una regla de derivación con la siguiente forma:

$$X \rightarrow < \gamma, \alpha >$$

Donde $X$ es un símbolo terminal, $\gamma$ la derivación de la regla en el idioma fuente y $\alpha$ la derivación de la regla en el idioma destino. Tanto $\gamma$ como $\alpha$ son cadenas de símbolos terminales y no terminales. Así, en el ejemplo anterior, $\gamma = "yu\ X_1\ you\ X_2"$ y $\alpha = "have\ X_2\ with\ X_1"$.

A las reglas comunes se añaden las reglas de cohesión, que permiten desarrollar la oración entera, ya que parten del símbolo inicial $S$:

$$S \rightarrow < S_1 X_1, S_1 X_1 > ; S \rightarrow < X_1, X_1 >$$

Para adaptar las SynCFG a la MT Estadística, cada regla de derivación deberá tener asociada una probabilidad. Con ello, obtendremos la mejor traducción mediante la secuencia de reglas de derivación que obtenga la mayor probabilidad acumulada. Esto puede calcularse mediante el algoritmo *inside-outside* (Baker, 1979).

## 1.3 Traducción Automática Interactiva

Si bien la MT fue concebida como un sistema cerrado que se encarga de todo el proceso de traducción, lo cierto es que actualmente aún posee una tasa de error bastante superior a la de un traductor humano. Por ello, excepto en sistemas en los que la velocidad prime sobre la calidad de la traducción –como traducción de habla en tiempo real–, los sistemas de MT interactuarán con un humano que se encargará de subsanar los errores.

Su enfoque más típico es el proceso conocido como post-edición, en el que la interacción entre el traductor humano y el sistema se realiza de forma indirecta. En él, un traductor humano corrige los textos generados por la máquina después de que ésta haya terminado de realizar la traducción. La principal desventaja de este enfoque es que el traductor debe enfrentarse a grandes volúmenes de texto, suponiendo un esfuerzo comparable a traducir el *corpus* en cuestión directamente.

En respuesta a este tedioso proceso, surge la Traducción Automática Interactiva o *Interactive Machine Translation* (de ahora en adelante, IMT), caracterizado por aceptar correcciones del usuario en tiempo real.

El enfoque más típico es el de IMT Basada en Prefijos. En éste, el usuario corregirá una palabra, considerando que el prefijo (palabras que han aparecido antes de ésta) es válido. De este modo, el sistema aceptará dicha palabra y generará un nuevo sufijo que considere más verosímil en relación con el prefijo.

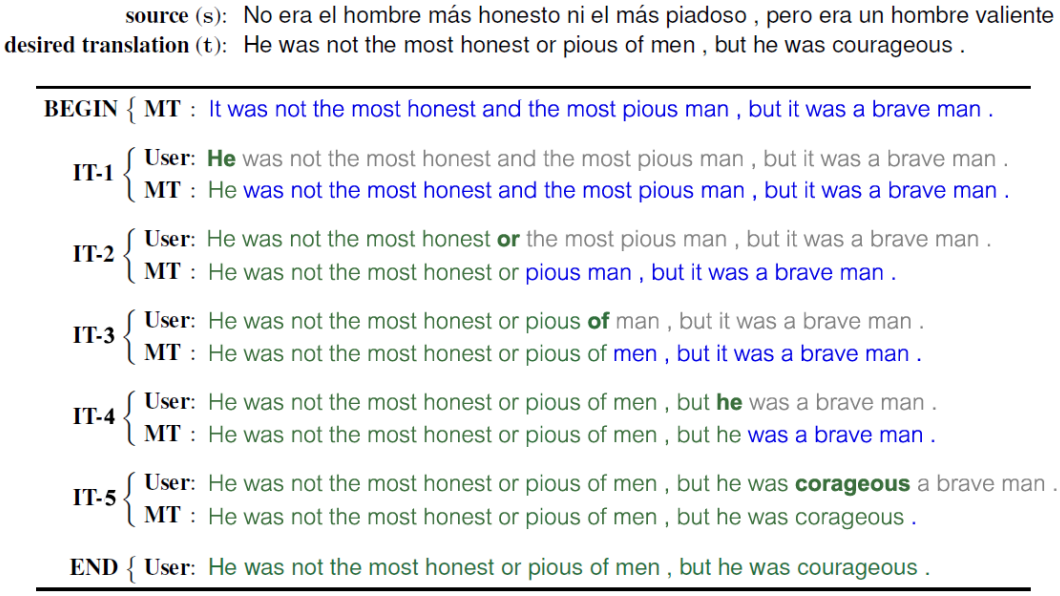


**Ilustración 2.** Ejemplo de IMT Basada en Prefijos del español al inglés. En azul, muestra la corrección propuesta por el sistema. En verde, segmentos validados por el usuario, con las palabras en negrita indicando que han sido introducidas mediante teclado.

La principal ventaja de este paradigma frente a MT convencional es una mejora de la sinergia entre el traductor humano y la máquina, permitiendo el ahorro de esfuerzo humano. Dicha sinergia también acarrea un cambio en la filosofía de estos sistemas, desplazando el foco al traductor humano, que emplearía este tipo de sistemas como un asistente para realizar su labor.

No obstante, los sistemas de IMT actuales que funcionan en tiempo real emplean modelos y protocolos de usuario muy simples, por lo que la generación de traducciones alternativas y la interacción con el usuario están muy limitadas.

El primer enfoque interactivo en MT fue Foster, et al. (1998), posteriormente desarrollado por Langlais & Lapalme (2002) en el proyecto TransType y que estaba basado en predecir una palabra como continuación de un prefijo de la traducción validado por el usuario. Posteriormente en los proyectos TransType2 (Barrachina, et al., 2009) y CasMaCat (Alabau, et al., 2013) las predicciones se extendieron al sufijo completo de la traducción.

A Barrachina, et al. (2009) le siguieron múltiples investigaciones relacionadas con IMT, como Koehn, et al. (2014). Algunas extensiones de su trabajo desarrollaron métodos de generación de sufijos alternativos (Ortiz-Martínez, 2011; Azadi & Khadivi, 2015), el uso de medidas de confianza para ayudar al usuario a validar prefijos válidos (González-Rubio, et al., 2010) o modelos jerárquicos (González-Rubio, et al., 2013) que sustituían al modelo *phrase-based* empleado por Barrachina.

Otros avances en IMT incluyen la adaptación de técnicas de aprendizaje *online*, esto es, que el sistema adapte comportamiento futuro a partir de las respuestas del usuario, pero teniendo en cuenta su conocimiento previo. Esta técnica permite evitar el reentrenamiento completo del sistema, proceso que suele implicar un

coste temporal elevado. Algunos artículos relevantes que estudiaron el impacto de estas técnicas fueron Martínez-Gómez, et al. (2012), que se centró en métricas de adaptación y Ortiz-Martinez (2016), que estudió el impacto sobre medidas clásicas como la tasa de error por palabra (WER) o Bilingual evaluation understudy (BLEU).

## 1.4 Traducción Automática Interactiva Basada en Segmentos de Palabras

Algunos experimentos comparativos entre IMT y post-edición (Koehn, 2014; Sanchis-Trilles, et al., 2014) demostraron que, a pesar de que el paradigma IMT requería menos correcciones por parte del usuario, el esfuerzo percibido y el tiempo empleado tendían a ser mayores. Ello fue achacado a varias limitaciones de la IMT Basada en Prefijos: el hecho de que el usuario está obligado a corregir de izquierda a derecha, al tiempo de aprendizaje requerido y a un mayor esfuerzo cognitivo, un factor que es difícil medir de forma objetiva.

Por ello, González-Rubio, et al. (2016) propuso un protocolo de usuario alternativo, conocido como IMT basada en Segmentos de Palabras. En éste, el usuario deberá marcar como correctos un conjunto de segmentos de la traducción obtenida del sistema, que éste empleará para generar una traducción alternativa acorde con los segmentos válidos. De esta forma, se dotará de mayor libertad al usuario, subsanando la primera limitación mencionada en el párrafo anterior.

El proceso de interacción comienza con el sistema sugiriendo una traducción al usuario. Éste marcará aquellos segmentos que considere correctos además de, opcionalmente, indicar el comienzo de la nueva traducción. A partir de esto, el sistema generará una traducción alternativa que sea compatible con la entrada del usuario, esto es, que contenga los segmentos indicados. Dicho proceso se repetirá de forma iterativa hasta que el usuario considere que la traducción sugerida por el sistema es adecuada.

En teoría, este nuevo paradigma permite aprovechar de forma eficaz las características de cada frase. Con ello, se reducirá el número de interacciones con el usuario y, en extensión, el esfuerzo de éste.

Esto puede apreciarse en la Ilustración 3, que permite traducir la frase de ejemplo con 2 interacciones. Este resultado representaría una mejora sustancial respecto al paradigma basado en prefijos (Ilustración 2), que requería 5 interacciones para obtener el mismo resultado.

**source** (s): No era el hombre más honesto ni el más piadoso , pero era un hombre valiente .
**desired translation** (t): He was not the most honest or pious of men , but he was courageous .

| | | |
|---|---|---|
| **BEGIN** | **MT** : | It was not the most honest and the most pious man , but it was a brave man . |
| **IT-1** | **User**: | It was not the most honest and the most pious **of** man , but it was a brave man . |
| | **MT** : | He was not the most honest or pious of men , but it was a brave man . |
| **IT-2** | **User**: | He was not the most honest or pious of men, but it was **courageous** . |
| | **MT** : | He was not the most honest or pious of men, but he was courageous . |
| **END** | **User**: | He was not the most honest or pious of men , but he was courageous . |

**Ilustración 3.** Ejemplo de IMT Basada en Segmentos de palabras del español al inglés, en el que se ejemplifica la misma frase que en la Ilustración 2. De nuevo, en azul, muestra la corrección propuesta por el sistema. En verde, segmentos validados por el usuario, con las palabras en negrita indicando que han sido introducidas mediante teclado.
Extraída de González-Rubio, et al. (2016).

# Capítulo 2. Fundamentos teóricos

A pesar de que el protocolo de usuario introducido por González-Rubio (2016) conseguía reducir el esfuerzo del traductor humano respecto a otros enfoques de IMT y la post-edición, la implementación del mismo presentaba otra limitación importante: el tiempo de ejecución de calcular traducciones alternativas era demasiado elevado para ser empleado por usuarios humanos. Esto se debía, entre otros factores, al hecho de que el lenguaje de programación empleado era *Python*.

Por ello, en el presente proyecto proponemos la migración del código fuente al lenguaje *C++*. A continuación se detallan las decisiones de diseño del sistema, tanto en lo que respecta a la interacción del usuario como a los fundamentos matemáticos subyacentes.

## 2.1 Aproximación al protocolo de usuario

El paradigma de interacción introducido en la sección 1.4 tiene como principal inconveniente que el coste temporal de buscar la traducción alternativa puede ser muy elevado si el número de palabras marcadas es alto. Esto puede parecer contradictorio *a priori,* ya que a más palabras más limitada está la búsqueda. No obstante, hay que tener en cuenta que el sistema no tiene información sobre qué hay entre los segmentos, lo que hace que el espacio de búsqueda sea más grande.

Por ello, siguiendo los pasos de González-Rubio (2016), presentaremos un modelo aproximado, en el que el usuario está limitado a marcar como correctas un número máximo de palabras total $M$, dándole libertad de repartirlas en tantos segmentos como considere.

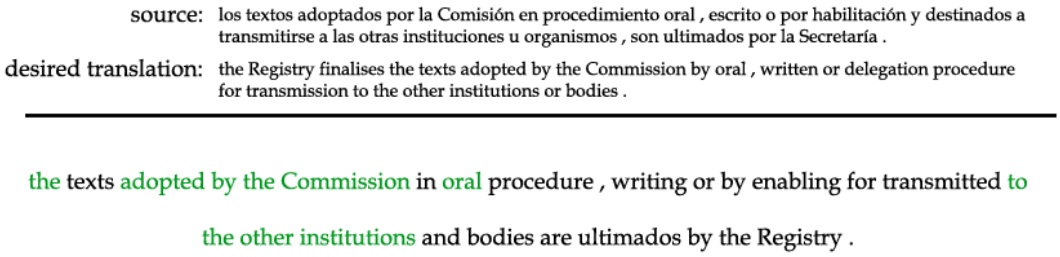


**Ilustración 4.** Muestra un ejemplo del protocolo de interacción descrito con $M = 10$.

No obstante, este paradigma está muy limitado, llegando a ser más rígido que la IMT basada en prefijos (sección 1.3) en determinadas circunstancias. Esto es especialmente relevante cuando el máximo de palabras $M$ es bajo. Además, presenta dificultades cuando aparecen palabras desconocidas para el sistema.

Para solucionar esto, combinaremos el paradigma basado en prefijos con el que hemos descrito. Así, el usuario deberá seguir los siguientes pasos en cada interacción.

- Marcar el prefijo correcto de máxima longitud.
- Insertar la palabra siguiente al prefijo.
- Marcar como correctas un máximo de $M$ palabras.

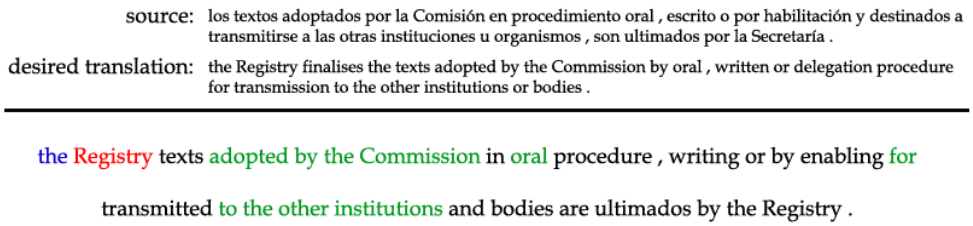


**Ilustración 5.** Muestra el ejemplo de la Ilustración 4, adaptado al protocolo de interacción extendido. En azul se marca el prefijo, en rojo la corrección y en verde el resto de segmentos marcados.

Este protocolo puede verse como una generalización del protocolo basado en prefijos, correspondiéndose este con $M = 0$.

## 2.2 Modelo estadístico

Como la mayoría de sistemas de MT actuales, la aplicación que hemos desarrollado se clasifica dentro de la Traducción Estadística, por lo que la traducción obtenida será aquella que el sistema considere más probable. No obstante, al tratarse de un sistema interactivo, es necesario reflejar la realimentación $f$, esto es, los segmentos que el usuario haya marcado. Así, la expresión a maximizar será la siguiente:

$$\hat{\boldsymbol{t}} = \arg\max_{\boldsymbol{t}} P(\boldsymbol{t}|\boldsymbol{s},\boldsymbol{f}) \tag{4}$$

Aplicando *naïve* Bayes –esto es, asumiendo que $\boldsymbol{s}$ y $\boldsymbol{f}$ son variables estadísticamente independientes–, aproximamos la expresión anterior como:

$$\hat{\boldsymbol{t}} = \arg\max_{\boldsymbol{t}} P(\boldsymbol{t}) \cdot P(\boldsymbol{s}|\boldsymbol{t}) \cdot P(\boldsymbol{f}|\boldsymbol{t}) \tag{5}$$

Esta expresión es muy similar a la Ecuación 2, añadiendo la probabilidad del modelo de corrección $P(\boldsymbol{f}|\boldsymbol{t})$, que representa el grado de verosimilitud entre la entrada del usuario y la traducción.

No obstante, puesto que $f$ y $t$ siempre son compatibles hasta cierto punto, $P(\boldsymbol{f}|\boldsymbol{t})$ nunca tendrá un valor nulo. Esto implica que el sistema puede generar traducciones que no contengan la realimentación $f$ por completo.

Para corregir este problema, definimos un alineamiento monótono $\boldsymbol{a} = a_1 \dots a_{|\boldsymbol{f}|}$ entre los segmentos validados por el usuario $\boldsymbol{f} = \boldsymbol{f}_1 \dots \boldsymbol{f}_{|\boldsymbol{f}|}$ y los segmentos de la traducción $\tilde{\boldsymbol{t}} = \tilde{t}_1 \dots \tilde{t}_{|\boldsymbol{f}|}$ que maximicen el número de coincidencias con $\boldsymbol{f}$. De este modo, cada enlace del alineamiento $a_k$ significará que el subsegmento $\tilde{t}_k$ debe ser sustituido por $f_k$.

Finalmente, incluyendo el alineamiento en la ecuación anterior obtenemos:

$$\hat{\boldsymbol{t}} = \underset{\boldsymbol{t}}{\operatorname{argmax}} \max_{\boldsymbol{a}} P(\boldsymbol{t}) \cdot P(\boldsymbol{s}|\boldsymbol{t}) \cdot P(\boldsymbol{f}, \boldsymbol{a}|\boldsymbol{t}) \tag{6}$$

Así pues, será necesario realizar una estimación de $P(\boldsymbol{t})$, $P(\boldsymbol{s}|\boldsymbol{t})$ y $P(\boldsymbol{f}, \boldsymbol{a}|\boldsymbol{t})$. Las dos primeras son aproximadas mediante un modelo log-lineal proporcionado por *Moses*, desarrollado por Koehn (2009). Pondremos más énfasis, pues, en la probabilidad del alineamiento $P(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{t})$.

De acuerdo con González-Rubio, et al. (2013), modelamos $P(\boldsymbol{f}, \boldsymbol{a}|\boldsymbol{t})$ con un modelo de corrección de errores basado en la distancia de edición de Levenshtein (1966). Dada una hipótesis $\boldsymbol{t}$ y la realimentación del usuario $\boldsymbol{f}$, asumimos que el número de ediciones $\delta$ necesario para adaptar $\boldsymbol{t}$ a $\boldsymbol{f}$ sigue una distribución binomial $\delta \sim \mathrm{B}(l_{\boldsymbol{t}}, p_e)$, donde $l_{\boldsymbol{t}}$ es la longitud de $\boldsymbol{t}$ y $p_e$ es la probabilidad *a priori* de error, el único parámetro libre del modelo estadístico.

Así pues, aproximamos la probabilidad de corrección de error como:

$$P(\boldsymbol{f}, \boldsymbol{a} \mid \boldsymbol{t}) \approx \prod_{k=1}^{|a|} P_E(\tilde{f}_k, a_k) \tag{7}$$

Donde $P_E(\tilde{f}_k, a_k)$ es la probabilidad de corrección de error del k-ésimo enlace del alineamiento $\boldsymbol{a}$:

$$P_E(\tilde{f}_k, a_k) = \binom{l_k}{\delta_k} p_e^{\delta_k} (1 - p_e)^{\left(l_{t_k} - \delta_k\right)} \tag{8}$$

Así pues, esta se calcula como un coeficiente binomial dependiente de $l_{\boldsymbol{t}_k}$ (longitud en palabras del k-ésimo segmento de $\tilde{\boldsymbol{t}}$) y $\delta_k$ (distancia de edición entre $\tilde{f}_k$ y el segmento de la hipótesis que se alinea con él $\tilde{t}_k$).

## 2.3 Búsqueda

A continuación abordaremos el problema de la búsqueda de la hipótesis más probable. Para representar el espacio de búsqueda se ha empleado una estructura de datos llamada hipergrafo, que almacena las diferentes alternativas de traducción para una frase dada.

### 2.3.1 Hipergrafo: concepto matemático

El concepto de hipergrafo surge de otra estructura matemática más sencilla: el grafo, propuesto por Euler en 1736 para resolver el problema de los puentes de Königsberg. Éstos se definen como un conjunto de nodos $N$ conectados por un conjunto de aristas $E$, de modo que cada arista conecta un par de nodos. Los grafos han demostrado ser muy potentes para diversos dominios de búsqueda, gracias a sus múltiples variantes, como los grafos ponderados (cuyas aristas tienen asociado un valor numérico) o los grafos dirigidos (en los que una conexión del nodo $a$ al $b$ no asegura la conexión recíproca).

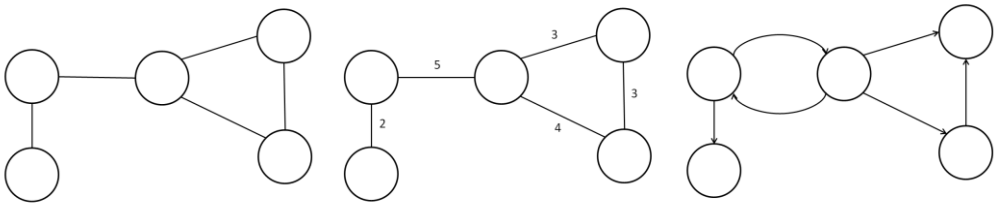


**Ilustración 6.** Muestra diferentes tipos de grafos. A la izquierda, un grafo básico. Al centro, un grafo ponderado, cuyas aristas tienen un coste asociado. A la izquierda, un grafo dirigido, en el que se pueden observar conexiones recíprocas y no recíprocas.

Los hipergrafos pueden entenderse como una generalización del concepto de grafo en la que las aristas (ahora hiperaristas) conectan más de dos nodos (ahora hipernodos).

Matemáticamente, definimos un hipergrafo como un par $H = \{N, E\}$, donde $N = \{n_1, n_2, \dots, n_{|N|}\}$ es el conjunto de nodos y $E = \{e_1, e_2, \dots, e_{|E|}\}$ es el conjunto de aristas. A su vez, definimos cada arista como $e_i \subseteq N$, representando así el subconjunto de nodos que conecta.

Introducimos también el concepto de aridad, una propiedad de las aristas que indica el número de nodos que éstas conectan. La aridad también puede aplicarse a los hipergrafos, siendo ésta equivalente a la máxima aridad de una arista perteneciente a él. Por ello, podemos decir que un grafo es un caso particular de hipergrafo con aridad 2.

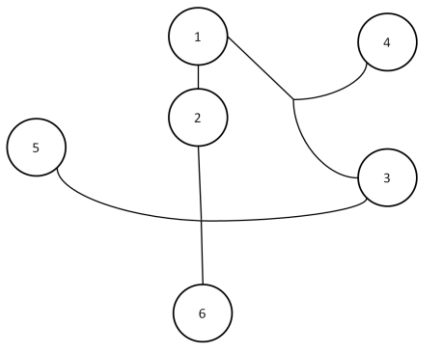


**Ilustración 7.** Hipergrafo de ejemplo. Su aridad es equivalente a 4, debido a la arista que comunica los nodos 2,3, 5 y 6.

Por supuesto, también es posible aplicar a los hipergrafos las modificaciones que se han mencionado para los grafos: la ponderación y la dirección. Si bien la naturaleza de los hipergrafos ponderados es bastante evidente, los dirigidos requieren alguna consideración adicional.

Los hipergrafos dirigidos contienen hiperaristas que comunican un conjunto de nodos origen con un conjunto de nodos destino. Una de las particularidades de estas estructuras es que para explorar los nodos destino de una arista es necesario haber visitado previamente sus nodos origen. Estableciendo una equivalencia con lógica binaria, los hipergrafos dirigidos permiten expresar tanto conjunciones como disyunciones, mientras que los grafos dirigidos sólo permiten expresar disyunciones.

Existen dos casos particulares de hipergrafo dirigido de especial interés: los hipergrafos con conexiones *forward* (o f-grafos) y los hipergrafos con conexiones tipo *backward* (o b-grafos). Ambos limitan uno de los conjuntos de cada arista: mientras que los f-grafos sólo permiten un nodo origen por arista, los b-grafos limitan a uno los nodos destino.

**Ilustración 8.** A la izquierda, una conexión tipo *backward* (muchos-a-uno); a la derecha, una conexión tipo *forward* (uno-a-muchos). Extraída de Gallo, et al. (1993).

Un tipo importante de hipergrafo dirigido son los llamados *árboles*, que se definen por la ausencia de ciclos. Esto significa que ningún nodo será ancestro de sí mismo. También implica necesariamente que en el hipergrafo existen nodos raíz (sin ningún nodo padre) y nodos hoja (sin ningún nodo hijo).

### 2.3.2 Hipergrafo de traducción

Una vez introducido el concepto de hipergrafo y algunas de sus particularidades, describiremos las características que tienen los que hemos empleado para representar el espacio de búsqueda del sistema de traducción que nos ocupa.

Así, cada oración a traducir se corresponde con un hipergrafo-árbol tipo *forward* de f-aridad 2. De este modo, cada arista tendrá exactamente un nodo cabeza (de ahora en adelante, padre) y entre cero y dos nodos cola (de ahora en adelante, hijos).

Cada nodo del hipergrafo representa una hipótesis de traducción de un fragmento de la oración completa. Las aristas, por otra parte, sirven para determinar cómo se combinan estos fragmentos en otros más grandes. Este anidamiento continúa hasta obtener la frase entera, representada mediante el nodo raíz, que no está conectado a ningún nodo padre.

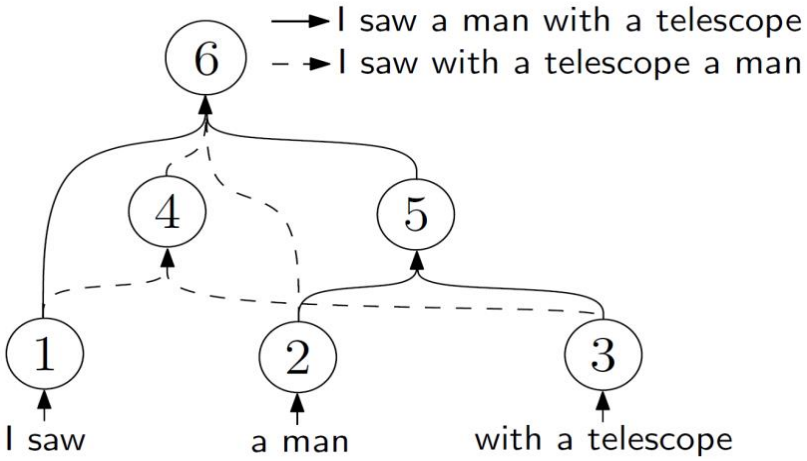


**Ilustración 9.** Hipergrafo de traducción que almacena dos posibles traducciones para la frase en español "Vi un hombre con un telescopio". Extraída de González-Rubio, et al. (2016).

Este paradigma se engloba dentro de la MT estructurada, ya que su potencia expresiva es equivalente a la de una gramática incontextual, sustituyendo los nodos por símbolos terminales y las aristas por reglas de derivación. Por ejemplo, la Ilustración 9 es equivalente a la gramática de la Ilustración 10.

$$S \rightarrow X_6$$
$$X_6 \rightarrow X_1 X_5$$
$$X_6 \rightarrow X_4 X_2$$
$$X_4 \rightarrow X_1 X_3$$
$$X_5 \rightarrow X_2 X_3$$
$$X_1 \rightarrow \text{"}I\ saw\text{"}$$
$$X_2 \rightarrow \text{"}a\ man\text{"}$$
$$X_3 \rightarrow \text{"}with\ a\ telescope\text{"}$$

**Ilustración 10.** Gramática incontextual equivalente al hipergrafo de la Ilustración 9.

Además de la información estructural mostrada en la Ilustración 9, tanto las aristas como los nodos almacenan información relativa a la hipótesis parcial de traducción que almacenan.

#### *Aristas*

Como se ha indicado anteriormente, al establecer la gramática equivalente a un hipergrafo, cada una de sus aristas se sustituye por una regla de derivación. Así pues, la parte derecha de cada regla o *right-hand side* (RHS) será almacenada en la arista correspondiente.

La principal restricción para ello es que las RHS deben tener tantos símbolos no terminales como nodos hijo tenga la arista a la que pertenece. De esta forma, podemos obtener la hipótesis más probable asociada a una arista sustituyendo cada uno de sus no terminales por la hipótesis más probable de cada uno de sus nodos hijo.

Las aristas también tienen asociada una probabilidad, que indica la verosimilitud de que la hipótesis más probable de su nodo-padre se calcule a través de ella.

Así pues, definimos las aristas de traducción como:

$$e = \{n_p, N_s, r, P\} \tag{9}$$

Siendo $n_p$ el nodo padre en el que incide la arista, $N_s$ los nodos hijos de los que parte, $r$ la RHS de la regla equivalente y $P$ su probabilidad. De ahora en adelante, nos referiremos a $r$ como $\mathrm{RHS}(e)$.

#### *Nodos*

En un hipergrafo de traducción, los nodos sirven para representar las hipótesis parciales candidatas a formar parte de la traducción completa. Esta hipótesis se será almacenada en el nodo y se calcula como la hipótesis asociada a la arista incidente en el nodo más probable, descrita en el subapartado anterior.

Cada nodos también tienen asociada la probabilidad *inside* (Baker, 1979) del símbolo no terminal equivalente a él en la gramática incontextual probabilística equivalente al hipergrafo. En la práctica, la probabilidad *inside* se calcula como el producto de las probabilidades de las aristas incidentes en el nodo.

Los nodos de traducción se definen matemáticamente como:

$$n = \{E_p, E_s, t, P_I\} \tag{10}$$

Con $E_p$ siendo el conjunto de aristas padre salientes del nodo, $E_s$ el conjunto de aristas hijo incidentes en él, $t$ la hipótesis parcial más probable y $P_I$ la probabilidad *inside*.

### 2.3.3 Algoritmo de búsqueda

Como se ha abordado en la sección 2.2, el objetivo del sistema es encontrar la hipótesis $\hat{\boldsymbol{t}}$ que maximice la expresión $P(\boldsymbol{t}) \cdot P(\boldsymbol{s}|\boldsymbol{t}) \cdot P(\boldsymbol{f}, \boldsymbol{a}|\boldsymbol{t})$ (Ecuación 4).

Este problema puede abordarse mediante programación dinámica, un esquema algorítmico empleado para resolver problemas de optimización. Más concretamente, es una estrategia algorítmica en la que se fragmenta el problema global en problemas locales de escala cada vez más reducida y cuyas soluciones se almacenan para reutilizarla más adelante si es necesario. Este esquema tiene dos particularidades que lo hacen adecuado a nuestro problema:

- **Optimalidad**. Siempre y cuando el problema sea divisible en problemas locales, un algoritmo de programación dinámica es capaz de encontrar la solución óptima. Nuestro problema es divisible, dado que la solución de un nodo se calcula empleando las soluciones de sus nodos-hijo.
- **Ahorro de coste temporal**. Gracias al almacenamiento de soluciones locales, la programación dinámica es menos costosa temporalmente que otros esquemas óptimos. Esto es de gran relevancia, al tratarse de una aplicación cuyo objetivo es funcionar en tiempo real.

Así pues, la búsqueda se realizará recorriendo los nodos en orden topológico, desde los nodos hojas hasta la raíz. Para cada uno de ellos se calcula la hipótesis más probable teniendo en cuenta la realimentación $\boldsymbol{f}$ del usuario.

La solución óptima de cada nodo, por su parte, se obtiene también de forma incremental, calculando la solución óptima teniendo en cuenta subconjuntos de segmentos marcados por el usuario que cumplan una condición determinada.

Los subconjuntos válidos para esta exploración reciben el nombre de *coberturas* de segmentos. Si definimos la realimentación como $\boldsymbol{f} = f_1 \dots f_{|\boldsymbol{f}|}$, de modo que $f_i$ es el segmento de palabras i-ésimo que ha marcado el usuario, la cobertura $\boldsymbol{c}_{ij}$ sería equivalente al conjunto de palabras entre la $i$-ésima y la $j$-ésima, ambas incluidas:

$$\boldsymbol{c}_{ij} = f_i \dots f_j \mid 0 < i, j \leq |\boldsymbol{f}|;\ \ i \leq j \tag{11}$$

Para ilustrar esto tomaremos como ejemplo la frase "*It was not the most honest and the most pious of man , but it was a brave man*", mostrada en la Ilustración 11. En ella, vemos cómo el usuario ha marcado los segmentos "*was not the most honest*", "*pious of*", "*but*" y "*was*".


**Ilustración 11.** Ejemplo de traducción obtenida por el sistema. Los segmentos validados e introducidos por el usuario están marcados en verde.

Algunas posibles coberturas para estos segmentos aparecen en la Tabla 1. Ésta también contiene ejemplos de coberturas mal formadas con una justificación de por qué no son válidas.

| Segmentos cubiertos | ¿Es válida? | Justificación |
|---|---|---|
| - | Sí | Cobertura λ (vacía) |
| *was not the most honest* | Sí | Cobertura (1, 1) |
| *was not the most honest , pious of* | Sí | Cobertura (1, 2) |
| *was not the most honest, pious of, but, was* | Sí | Cob. completa (1, 4) |
| *the most honest, pious of* | No | El segmento 1 está incompleto |
| *was not the most honest, but* | No | No cubre el segmento 2, pero sí el 1 y el 3 |
| *but, pious of* | No | Invierte el orden de los segmentos 2 y 3 |

**Tabla 1.** Diversas coberturas correctas e incorrectas para la realimentación del usuario que aparece en la Ilustración 11.

El algoritmo consistirá, pues, en rellenar progresivamente una matriz bidimensional $\boldsymbol{M}$ de tamaño $|N| \times |C|$, siendo $N$ el conjunto de los nodos y $C$ el conjunto de coberturas válidas. De esta forma, $\boldsymbol{M}(n, c_{ij})$ almacenará la hipótesis más probable (y su probabilidad) del nodo $n$ si sólo se tuvieran en cuenta los segmentos de palabras cubiertos por la cobertura $c_{ij}$. Una vez finalizado el algoritmo, el resultado final se almacenará en la celda $(|N|, |C|)$, con el nodo $|N|$ siendo la raíz y la cobertura $|C|$ la completa.

Para cada nodo $n$, el algoritmo atravesará varias fases:

1. **Coberturas vacía**. Inserta en la celda correspondiente a $n$ y a la cobertura vacía $c_\lambda$ la hipótesis *a priori* $\boldsymbol{t}(n)$.
2. **Coberturas de un solo segmento**. Calcula hipótesis de traducción teniendo en cuenta un solo segmento, que será equivalente a $\boldsymbol{t}(n)$ con su prefijo sustituido por el segmento cubierto $\boldsymbol{f}_i$. La longitud del prefijo sustituido se calcula mediante la distancia de Levenshtein.
3. **Combinaciones de coberturas exploradas por los hijos**. Para cada arista $e$ incidente en $n$, combinará las coberturas exploradas por los nodos hijos $N_S(e) = \{n_{si}(e) \mid \forall i, 1 \leq i \leq |N_S(e)|\}$ de $e$ de todas las maneras posibles que el resultado también sea una cobertura válida. Definimos cada combinación de coberturas como $\boldsymbol{c}_{ij} = \{c_1, \dots, c_{|N_S(e)|}\}$.
4. **Para cada combinación de coberturas $\boldsymbol{c}_{ij}$** obtenida en el paso 3, derivará la RHS de la arista $e$, sustituyendo cada uno de los $|N_S(e)|$ símbolos no terminales por las cadenas almacenadas en las celdas de la matriz $\boldsymbol{M}(n_{si}(e), c_i)| \; \forall i, 1 \leq i \leq |N_S(e)|$.
5. **Extensión a izquierda** para cada combinación de coberturas $\boldsymbol{c}_{ij}$ obtenida en el paso 3. Si $\boldsymbol{c}_{ij}$ no cubre el primer segmento $(i > 1)$, calculamos el alineamiento entre $\boldsymbol{t}(n)$ y $\boldsymbol{f}_{i-1}$, de modo similar al descrito en el paso 2.
6. **Extensión a derecha** para cada combinación de coberturas $\boldsymbol{c}_{ij}$ obtenida en el paso 3. De forma similar al paso 5, si $\boldsymbol{c}_{ij}$ no cubre el último segmento $(j < |\boldsymbol{f}|)$, calculamos el alineamiento entre $\boldsymbol{t}(n)$ y $\boldsymbol{f}_{j+1}$. En esta ocasión, el fragmento de $\boldsymbol{t}(n)$ que sustituiremos por $\boldsymbol{f}_{j+1}$ es el sufijo, a diferencia de los pasos 2 y 5 donde lo hacíamos con el prefijo.

En caso de que una de estas fases explore un par $(n, c)$ que ya haya sido explorado anteriormente, la matriz almacenará el que tenga una mayor probabilidad asociada.

Para consultar la definición matemática del algoritmo con más detalle junto con el coste temporal, véase el Anexo II (más adelante).

# Capítulo 3. Experimentación

En este capítulo se detallan los procedimientos que se han seguido para evaluar el rendimiento y corrección del sistema desarrollado.

En primer lugar, introduciremos el *software*, métricas y *corpus* empleados. A continuación, describiremos el objetivo de los diferentes experimentos, así como la configuración empleada para cumplirlos. Por último, presentaremos los resultados y extraeremos conclusiones al respecto.

## 3.1 Software

A continuación se describirá el *software* de apoyo que se ha empleado.

***Moses***

Moses (Kohen, et al., 2007) es un *toolkit* de código abierto de traducción que permite entrenar modelos estadísticos entre el lenguaje origen y el lenguaje destino. Incluye un algoritmo decodificador que, una vez entrenado el modelo, permite obtener la traducción más probable para un conjunto de frases. También puede determinar la calidad de la traducción obtenida si se posee el conjunto de frases traducidas por un humano.

El decodificador que se ha empleado en nuestro caso concreto es *chart*, que se enmarca dentro de la traducción jerárquica. Este decodificador puede obtener como salida hipergrafos correspondientes a las frases traducidas, que servirán como entrada a nuestro sistema.

***SRILM***

SRILM (Stolcke, 2002) es un *toolkit* diseñado para estimar modelos de lenguaje de gran tamaño. Además de en traducción automática, esta herramienta se emplea en otros campos de *machine learning* relacionados con el lenguaje, como reconocimiento del habla, reconocimiento de texto manuscrito o etiquetado.

***MGIZA++***

MGIZA++ (Gao & Stephan, 2008) es una implementación multi-hilo de GIZA++ (Och & Ney, 2003), un *toolkit* que calcula alineamientos de palabras entre oraciones de un *corpus* bilingüe empleando métodos estadísticos independientes del par de lenguas involucrado.

## 3.2 Métricas

En este apartado se detallarán las métricas que se han empleado para evaluar el desempeño de nuestro sistema, atendiendo a distintas facetas del mismo.

***BiLingual Evaluation Understudy (BLEU)***

Propuesto por Papineni, et al. (2002), el BLEU es uno de los métodos más usados para la evaluación automática de *Machine Translation*. Esta medida calcula la media de precisión para cada par de n-gramas entre los idiomas origen y destino

($p_n$). También incorpora el factor $BP$, que evita dar demasiado peso a las oraciones cortas. Su valor oscila entre 0 y 100, con éste último representando una traducción perfecta:

$$BLEU = BP \cdot \exp(\sum_{n=1}^{N} \frac{\log p_n}{N}) \quad (12)$$

*Moses* permite calcular esta métrica para un conjunto de test bilingüe. En el caso que nos ocupa, el BLEU se obtiene al mismo tiempo que se genera el hipergrafo.

Así pues, usaremos esta medida para estimar la calidad de la traducción inicial, es decir, la que se calcula sin tener en cuenta al usuario. Esto depende, sobretodo, de la similitud de las lenguas implicadas y de su ambigüedad, especialmente de la de la lengua destino.

En relación con nuestro caso, el BLEU podría actuar como indicador de la calidad y complejidad de los hipergrafos, si bien su fiabilidad para este menester aún está por determinar. Así pues, en los experimentos compararemos la evolución de esta métrica con el tiempo de respuesta, que está estrechamente relacionado con la complejidad de las estructuras.

### *Word-Stroke Ratio (WSR)*

El *Word-Stroke Ratio* (Tomás & Casacuberta, 2006) es una métrica que se emplea para estimar el esfuerzo del usuario en un sistema de IMT. Ésta se calcula como el número de palabras introducidas por teclado partido por el número total de palabras en el conjunto de frases traducidas. Cuanto mayor es esta métrica, más esfuerzo realiza el usuario.

$$WSR = \frac{número\ de\ palabras\ tecleadas}{número\ total\ de\ palabras} \quad (13)$$

### *Mouse-Action Ratio (MAR)*

Propuesta por Barrachina, et al. (2009), el *Mouse-Action Ratio* es una medida similar al WSR en el sentido de que ambas se emplean para evaluar el esfuerzo del usuario. En este caso, se calcula como el número de clics partido por el total de palabras en el conjunto de frases traducidas. Al igual que en el WSR, cuanto mayor es esta métrica, peores resultados indica.

$$MAR = \frac{número\ de\ clics\ realizados}{número\ total\ de\ palabras} \quad (14)$$

Conceptualmente, la diferencia fundamental entre el WSR y el MAR es que el primero tiene en cuenta el esfuerzo físico del usuario (teclear las palabras necesarias), mientras que el MAR estima el esfuerzo cognitivo del usuario (buscar visualmente los segmentos correctos). Por tanto, consideramos ambas métricas como complementarias a la hora de calcular el esfuerzo humano necesario para la tarea.

Otra consideración importante entre ambas medidas es determinar a cuál daremos prioridad en caso de conflicto entre ambas. Dado que la principal ventaja que un sistema IMT ofrece frente a la traducción tradicional es la reducción del número de palabras introducidas por teclado, consideraremos al WSR como más importante, ya que es la que refleja esta mejora.

#### *Tiempo de respuesta*

Esta métrica mide el tiempo medio que el usuario debe esperar desde que introduce una corrección hasta que el sistema produce una traducción alternativa. Esto se corresponde, mayoritariamente, con el tiempo de ejecución del algoritmo de búsqueda (apartado 2.3.3) y también, en menor medida, con el tiempo necesario para leer el hipergrafo del fichero.

Al tratarse de un sistema interactivo, esta métrica es sumamente relevante, ya que un tiempo demasiado elevado se traduce en que el sistema es inutilizable. También es útil para compararlo con González-Rubio, et al. (2016), al ser la única mejora prevista respecto a éste.

De acuerdo con Nielsen (1993), el tiempo que debe tardar una interacción para considerarse *tiempo real* es de 0,1 segundos o menos, dando al usuario la impresión de que la espera ha sido instantánea. Una vez superado este umbral, consideraremos como aceptables aquellas interacciones de menos de 1 segundo, ya que a partir de este punto se suele romper el flujo del pensamiento del usuario.

El principal problema de esta métrica es la existencia de *outliers* o valores anómalos. Mientras que la inmensa mayoría de las interacciones se sitúan dentro de lo que consideraríamos aceptable, pueden existir casos en los que el tiempo se dispare, superando el umbral de 1 segundo por varios órdenes de magnitud.

El valor de estos *outliers* es tan extremo que desvirtúan totalmente métricas como la desviación típica, que adquiere valores superiores a la media. Por ello, reportaremos tanto la media como la mediana. Así pues, la media estará muy influenciada por los valores extremos, mientras que la mediana consistirá en una estimación de la mayoría de los casos.

Definimos, en relación a esto, la métrica $\hat{t}$, que mide la relación entre la media y la mediana, sirviendo como indicador de la varianza:

$$\hat{t} = \frac{tiempo\ medio - tiempo\ mediano}{tiempo\ medio} \tag{15}$$

## 3.3 Corpus

Para realizar los experimentos, se han empleado un total de seis *corpus* distintos, que pertenecen a dos tareas de diferentes dominios: *Boletín de la Unión Europea* (*EU*) y *Manuales de impresoras Xerox* (*Xerox*), introducidos por Barrachina, et al. (2009). Ambos vienen en tres pares de lenguas: español-inglés (Es-En), alemán-inglés (De-En) y francés-inglés (Fr-En).

Tanto *EU* como *Xerox* se han dividido en tres conjuntos: *training*, *development* y *test*. Emplearemos el conjunto de *training* para entrenar el modelo de lenguaje y *development* para ajustar los pesos de éste. El conjunto de *test* se emplea para evaluar el rendimiento del sistema y, en este caso, para generar los hipergrafos.

| | | **EU** | | |
|---|---|---|---|---|
| | | **Es-En** | **Fr-En** | **De-En** |
| **Training** | **Frases** | 214K | 983K | 989K |
| | **Palabras** | 6,0M / 5,3M | 20M / 19M | 18M / 19M |
| | **Vocabulario** | 84K / 70K | 161K / 150K | 242K / 152K |
| **Development** | **Frases** | 400 | 400 | 400 |
| | **Palabras** | 12K / 10K | 12K / 10K | 10K / 10K |
| | **Vocabulario** | 3,1K / 2,8K | 2,9K / 2,6K | 3,1K / 2,6K |
| **Test** | **Frases** | 800 | 800 | 800 |
| | **Palabras** | 23K / 20K | 22K / 20K | 19K / 20K |
| | **Vocabulario** | 4,7K / 4,2K | 4,5K / 3,9K | 5,0K / 3,9K |

**Tabla 2.** Datos de los que disponemos de la tarea *EU* en cada par de lenguas. Estos se expresan como el número de frases, el total de palabras y el tamaño del vocabulario para cada uno de los tres conjuntos: *training*, *development* y *test*. K y M representan miles y millones, respectivamente.

Estas tareas poseen naturalezas muy diferentes, como se refleja en la Tabla 2 y la Tabla 3. En primer lugar, *EU* posee un número de frases de entrenamiento significativamente más alto que *Xerox*, por lo que sus resultados serán más fiables. No obstante, *EU* tiene menos frases en español que en francés y alemán, por lo que la comparativa entre lenguas puede no ser justa.

Otro factor de divergencia es la longitud de las frases: mientras que en *EU* la media es de aproximadamente 20 palabras por fase, en *Xerox* es de 12.

Respecto a la densidad de vocabulario (proporción de palabras vistas por primera vez respecto al total), parece ser que es menor en *EU*. Sin embargo, este hecho parece achacable al volumen de los datos, ya que dentro de *EU*, observamos una menor densidad en las frases inglesas de *Es-En* que en sus análogas de *Fr-En* y *De-En*.

| | | **Xerox** | | |
|---|---|---|---|---|
| | | **Es-En** | **Fr-En** | **De-En** |
| **Training** | **Frases** | 56K | 52K | 49K |
| | **Palabras** | 749K / 665K | 667K / 615K | 539K / 593K |
| | **Vocabulario** | 17K / 14K | 16K / 14K | 25K / 14K |
| **Development** | **Frases** | 1,0K | 994 | 964 |
| | **Palabras** | 16K / 14K | 12K / 11K | 11K / 11K |
| | **Vocabulario** | 1,8K / 1,6K | 1,9K / 1,8K | 1,7K / 1,5K |
| **Test** | **Frases** | 1,1K | 984 | 996 |
| | **Palabras** | 10K / 8,3K | 12K / 11K | 12K / 13K |
| | **Vocabulario** | 1,9K / 1,9K | 1,8K / 1,7K | 2,2K / 1,8K |

**Tabla 3.** Datos de los que disponemos de la tarea *Xerox* en cada par de lenguas. Estos se expresan como el número de frases, el total de palabras y el tamaño del vocabulario para cada uno de los tres conjuntos: *training*, *development* y *test*. K y M representan miles y millones, respectivamente.

Además de sobre las tareas, los datos de los *corpus* también proporcionan información de los cuatro lenguajes implicados. Para hacer esta comparación, estudiaremos los datos de *Xerox*, ya que *EU* puede estar condicionada por el menor número de frases en español.

En esta tarea, la longitud media de las frases parece variar entre los idiomas. Ordenadas de mayor a menor, tenemos: 13,38 en español, 12,82 en francés, 11,92 en inglés y 11,00 en alemán.

También podemos apreciar que, mientras que la densidad de vocabulario tiene valores similares en español, francés e inglés (2,27%, 2,4% y 2,25%), en alemán es drásticamente mayor (4,64%). Probablemente, esto se debe a la naturaleza aglomerante del alemán, ya que unir nombres y adjetivos crea nuevas palabras.

## 3.4 Simulación del usuario

Al estar enmarcado dentro de la Traducción Automática Interactiva, el objetivo final del sistema que nos ocupa es ser utilizado por traductores humanos. No obstante, realizar las pruebas sobre un usuario real hubiera sido un proceso costoso, tanto económica como temporalmente. Además, si dichas pruebas no se realizaran sobre un conjunto muy extenso de sujetos, éstas estarían muy condicionadas por las habilidades de estos.

Por ello, se ha desarrollado un algoritmo que simula el comportamiento que tendría un usuario humano, de acuerdo con el protocolo especificado en la sección 2.1. Para ello, se emplean las frases del conjunto de test en el idioma destino, que se considerarán como la salida deseada por el usuario.

Cada vez que el sistema genere una traducción potencial, el usuario simulado calculará la distancia de edición de Levenshtein (1966) entre ésta y la traducción deseada. Así pues, las palabras que estén alineadas consigo mismas, se marcarán como correctas, mientras que las ediciones serán consideradas para ser introducidas por el usuario.

El comportamiento del usuario simulado está determinado por dos características fundamentales:

- **Voracidad**: el usuario hace las correcciones en orden de aparición, por lo que siempre introducirá por teclado la primera palabra que no esté alineada. Además, si el parámetro *M* (sección 2.1) no permitiera marcar todas las palabras alineadas, dará prioridad a las que aparezcan antes en la oración.
- **El usuario sólo marcará segmentos en la primera iteración**. En el resto, se limitará a marcar el prefijo común más largo e introducir una nueva palabra.

Esto tiene como objetivo simplificar su comportamiento, evitando situaciones indeseables en las que realice interacciones contraproducentes entre ellas. Por tanto, al presentar un comportamiento determinado por estas limitaciones, los resultados serán pesimistas en comparación con el esperado de un usuario real.

## 3.5 Configuración de los experimentos

En esta sección se describe los parámetros iniciales de los dos experimentos realizados, así como el objetivo de cada uno de ellos.

### 3.5.1 Experimento 1

Puesto que nuestro sistema es equivalente lógicamente al desarrollado por González-Rubio, et al. (2016), nuestra primera tarea consistirá en replicar el experimento realizado por éste. Dicho experimento consiste en la traducción de 800 frases de la tarea *EU* del español al inglés. Para esto se realizará un barrido del parámetro *M* (sección 2.1) con valores entre 0 y 40, observando la relación entre el tiempo de espera y los parámetros de esfuerzo.

En este experimento, se ha realizado un pre-proceso de categorización. Esto significa que ciertas palabras, como los números o las fechas, se han sustituido por etiquetas correspondientes a la categoría a la que pertenecen. De este modo, se reduce complejidad a la tarea.

La métrica más relevante en este experimento es el tiempo de espera, al ser la única discrepancia esperada entre ambos sistemas. No obstante, también se reportarán los resultados atendiendo al resto de métricas y se estudiará su evolución en función de *M*.

### 3.5.2 Experimento 2

El objetivo del segundo experimento es evaluar el desempeño del sistema con todos los *corpus*, estudiando el impacto de las tareas y los lenguajes involucrados.

Con un valor de *M* bastante elevado (100), se realizará la traducción de las frases de test de las tareas *EU* y *Xerox* en los pares español-inglés, alemán-inglés y francés-inglés. Esto se realizará en ambas direcciones, por lo que se ejecutarán un total de 12 subtareas. No se ha realizado categorización para realizar este experimento.

## 3.6 Resultados

A continuación se muestran los resultados de los experimentos descritos en la sección 3.5, además de extraer conclusiones sobre los mismos.

### 3.6.1 Experimento 1

En este apartado se muestran los resultados del primer experimento. A tenor de estos, se determinará si nuestro sistema es equivalente al de González-Rubio, et al. (2016), así como la mejora de tiempo de ejecución respecto a éste.

#### *BLEU*

El BLEU obtenido en esta tarea es de 44,2. Es necesario remarcar que este valor hace referencia a la tarea *EU* con el par de lenguas Es-En, sobre el que se ha aplicado en este un pre-proceso de categorización.

### *Métricas de esfuerzo*

En la Tabla 4 se muestran los valores de WSR y MAR que se han obtenido para cada valor de $M$. Estos resultados se han obtenido tanto en nuestro sistema como en el desarrollado por González-Rubio, et al. (2016), por lo que podemos afirmar que ambos tienen un comportamiento equivalente.

| *M* | WSR (%) | MAR (%) |
|---|---|---|
| 0 | 49,3 | 14,2 |
| 1 | 48,1 | 17,3 |
| 2 | 47,2 | 19,3 |
| 3 | 46,2 | 21,1 |
| 4 | 45,2 | 22,8 |
| 6 | 43,1 | 25,7 |
| 8 | 41,3 | 28,3 |
| 10 | 39,6 | 30,5 |
| 20 | 34,6 | 35,9 |
| 30 | 33,5 | 36,9 |
| 40 | 33,3 | 36,9 |

**Tabla 4.** Valores de las métricas de esfuerzo WSR y MAR, expresadas en tanto por cien, para cada valor del parámetro $M$. Estos resultados han sido obtenidos tanto con nuestro sistema como con el de González-Rubio, et al. (2016).

Como se puede apreciar, el WSR disminuye con el valor de $M$, ya que cuanto mayor es éste más informada está la búsqueda, lo que produce un aumento en la calidad de la traducción alternativa. Si comparamos los resultados basados en prefijos ($M = 0$) con el que más se aproxima al protocolo sin límite de palabras ($M = 40$), la mejora total es de 16 puntos de WSR.

La relación entre el WSR y $M$ es constante, hasta llegar a $M = 10$. A partir de este valor, cada vez hay menos frases que se beneficien de validar tantas palabras, con lo que el impacto del parámetro es cada vez menor. Esto se aprecia más los valores más altos, con $M = 30$ y $M = 40$ produciendo resultados casi idénticos.

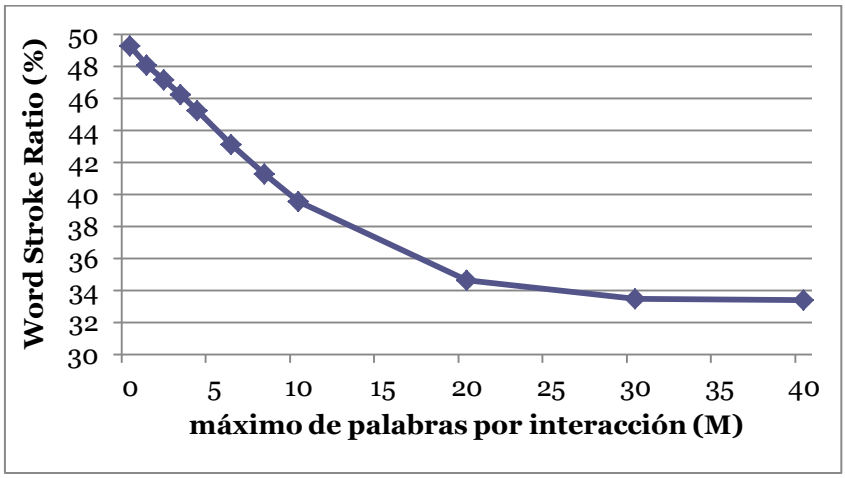


**Ilustración 12.** Progresión del WSR en ambos sistemas de acuerdo con el parámetro $M$.

No obstante, la reducción del número de palabras introducidas por teclado es a costa del aumento de palabras validadas mediante el ratón. Esto se puede apreciar con el aumento del MAR (Ilustración 13).

Si bien en la sección 3.2 hemos establecido que el WSR es más prioritario que el MAR, la mejora del primero también repercute negativamente en el tiempo de espera, como veremos más adelante. Un posible valor de compromiso sería $M = 10$, ya que la mejora en términos de WSR a partir de este valor es bastante reducida en comparación con el aumento de coste temporal (Tabla 5).

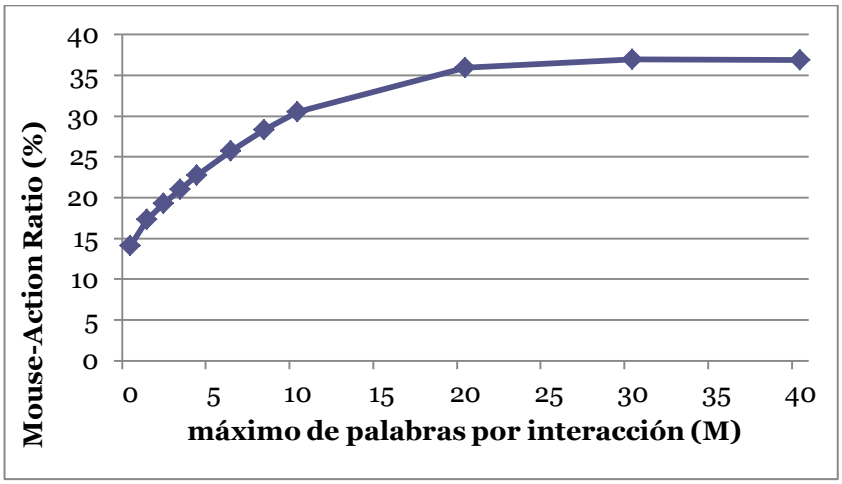


**Ilustración 13.** Progresión del MAR en ambos sistemas, de acuerdo con el parámetro *M*.

### *Tiempo de respuesta*

La Tabla 5 muestra el tiempo de respuesta obtenido con nuestro sistema y el de González-Rubio, et. al. (2016) para cada valor de *M* (número de palabras que el usuario puede marcar fuera del prefijo).

| | **Tiempo de respuesta (s)** | | | |
|---|---|---|---|---|
| ***M*** | Nuestra aproximación (C++) | | González-Rubio, et al. (2016) (Python) | |
| | Media | Mediana | Media | Mediana |
| 0 | 0,20 | 0,20 | 0,87 | 0,85 |
| 1 | 0,22 | 0,21 | 0,96 | 0,94 |
| 2 | 0,25 | 0,22 | 1,16 | 0,96 |
| 3 | 0,29 | 0,23 | 1,54 | 1,04 |
| 4 | 0,37 | 0,24 | 2,11 | 1,10 |
| 6 | 0,64 | 0,26 | 4,04 | 1,25 |
| 8 | 1,06 | 0,30 | 7,03 | 1,47 |
| 10 | 1,62 | 0,34 | 10,98 | 1,75 |
| 20 | 6,25 | 1,18 | 40,97 | 11,30 |
| 30 | 8,03 | 1,74 | 52,17 | 15,90 |
| 40 | 8,17 | 1,80 | 53,61 | 16,58 |

**Tabla 5.** Tiempo de respuesta medio y mediano obtenido con ambos sistemas para cada valor de *M*. Se expresa en segundos.

Si bien no hemos alcanzado tiempo real en ninguna de las ejecuciones, la mejora que hemos obtenido es bastante significativa. Si en el sistema de González-Rubio, et. al. (2016), el tiempo medio sólo es aceptable hasta $M = 1$, mientras que en el no deja de serlo hasta $M = 6$. En el caso de la mediana, esta diferencia se agudiza aún más, siendo los últimos valores de $M$ que obtienen resultados aceptables 2 y 10, respectivamente.

Atendiendo a la Ilustración 14, el factor de mejora es más relevante para valores de $M$ elevados. Esto se debe a que la mayoría de optimizaciones se han llevado a cabo sobre el algoritmo de búsqueda, por lo que la mejora será menor en casos en los que el tiempo de carga de los hipergrafos representa un porcentaje importante del coste computacional.

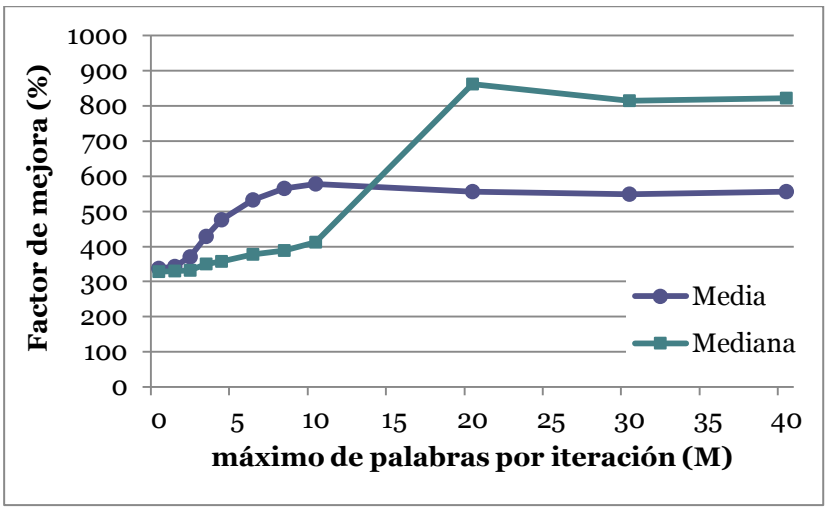


**Ilustración 14.** Mejora porcentual de los tiempos de respuesta medios y medianos obtenida sobre el sistema de González-Rubio, et al. (2016).

Respecto a la varianza de los datos, en ambos sistemas se observa que las diferencias entre la media y la mediana son cada vez más elevadas: con $M = 0$ ambas tienen valores casi idénticos, mientras que en $M = 40$ la media es 4,5 veces más grande. Esta progresión se aprecia con claridad en la Ilustración 15.

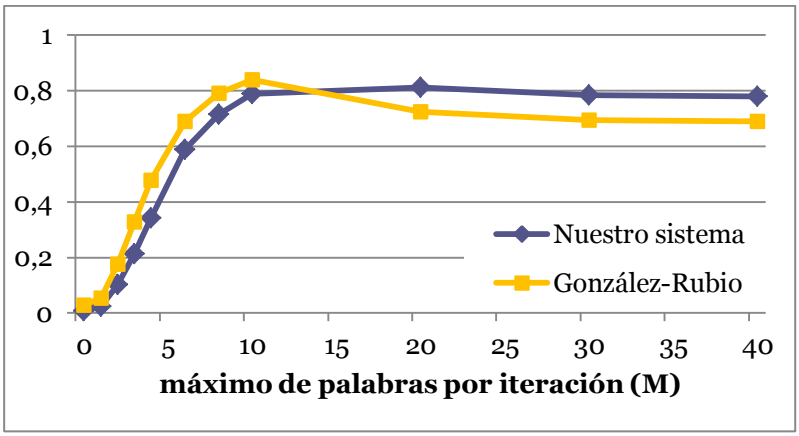


**Ilustración 15.** Progresión de la métrica $\hat{t}$ con el valor de $M$. Refleja la diferencia entre la media y la mediana, lo que constituye una aproximación de la varianza.

### 3.6.2 Experimento 2

A continuación se muestran los resultados del segundo experimento, cuyo objetivo es comparar cómo el comportamiento del sistema con distintos pares de lenguas: español-inglés, alemán-inglés y francés-inglés.

#### *BLEU*

En la Tabla 6 podemos observar el BLEU obtenido para cada *corpus*.

En primer lugar, atendiendo a los resultados de la tarea *EU* con las lenguas *Es-En*, observamos que el resultado está por debajo de los 44,2 puntos de BLEU del Experimento 1 (apartado 3.6.1), que también usaba *EU Es-En*. Esto es principalmente achacable a la falta de categorización del experimento que nos ocupa, aunque también puede deberse a los parámetros del entrenamiento.

| | **BLEU (%)** | |
|---|---|---|
| | **EU** | **Xerox** |
| En-Es | 46,4 | 58,5 |
| Es-En | 40,5 | 44,4 |
| En-Fr | 48,3 | 35,4 |
| Fr-En | 43,7 | 33,5 |
| En-De | 33,9 | 22,7 |
| De-En | 39,5 | 32,3 |

**Tabla 6.** BLEU obtenido al entrenar los hipergrafos correspondientes a cada corpus.

Si comparamos el desempeño para cada par de lenguas, en la tarea de *EU* el *corpus* que mejores resultados obtiene es el francés, mientras que en *Xerox* es el español. Esta discrepancia podría explicarse tanto por el número de frases de entrenamiento significativamente menor que tiene el español en *EU* como por el bajo número de frases de *Xerox* (Tabla 2 y Tabla 3).

No obstante, ambas tareas coinciden en el hecho de que el alemán es el que peores resultados obtiene, lo que puede ser sorprendente debido a la similitud entre éste y el inglés. No obstante, estas similitudes son sobretodo léxicas y las reglas sintácticas suelen desempeñar un papel más importante que éstas. Ergo, las particularidades gramaticales del alemán (aglomeración, género neutro, declinaciones) hacen que, a efectos de la MT, se la considere una lengua más diferente del inglés de lo que lo son el español o el francés.

Respecto a la dirección de los pares de lenguas, tanto el español como el francés obtienen mejores resultados cuando actúan como lengua destino, mientras que el alemán tiene un mejor desempeño como lengua origen. Puesto que el BLEU está más influido por la ambigüedad de la lengua destino, podemos considerar que el inglés es más ambiguo que el francés y el español, pero menos que el alemán.

Nótese que esta ordenación coincide con la media de longitud por frase que observamos en la sección 3.3, con lo que los resultados parecen indicar que la ambigüedad está directamente relacionada con la longitud de las frases.

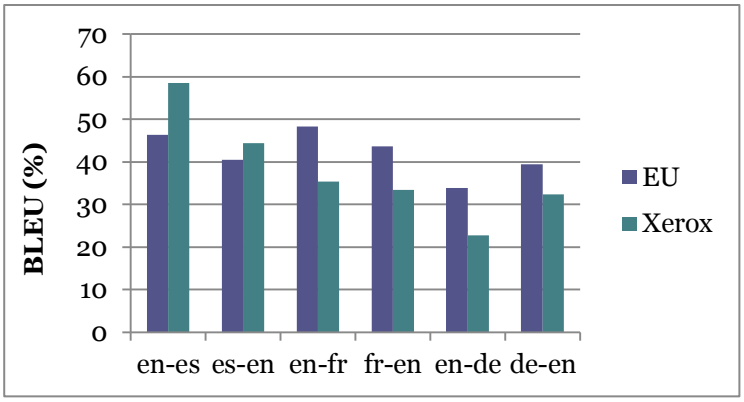


**Ilustración 16.** BLEU para las tareas *EU* y *Xerox* en cada par de lenguas.

A pesar de que ambas tareas a conclusiones similares, *Xerox* presenta un grado de varianza mucho mayor, con una diferencia de 35,8 puntos entre el mejor y el peor caso, frente a los 14,2 puntos de *EU*. Esto puede deberse al hecho de que *Xerox* consiste en manuales de impresoras, por lo que tiene muchas frases con estructuras similares. Por tanto, sus resultados están muy condicionados por el modo en el que cada idioma trata dichas estructuras.

### *Métricas de esfuerzo*

A diferencia de lo extraído en el Experimento 1 (Tabla 4), los resultados mostrados en la Tabla 7 no presentan una correlación inversa entre el WSR y el MAR. Así pues, dicha correlación sólo parece aplicarse cuando variamos el valor de *M* con los mismos idiomas, pero no cuando comparamos diferentes pares de lenguas.

| | **EU** | | **Xerox** | |
|---|---|---|---|---|
| | **WSR (%)** | **MAR (%)** | **WSR (%)** | **MAR (%)** |
| En-Es | 34,7 | 33,9 | 25,1 | 32,3 |
| Es-En | 32,9 | 41,8 | 29,5 | 42,9 |
| En-Fr | 32,6 | 32,0 | 42,4 | 39,0 |
| Fr-En | 29,1 | 41,0 | 42,9 | 42,6 |
| En-De | 45,9 | 35,1 | 52,4 | 39,1 |
| De-En | 40,0 | 36,8 | 44,9 | 40,0 |

**Tabla 7.** Métricas de esfuerzo WSR y MAR obtenidas para cada *corpus* empleando nuestro sistema.

Si observamos los WSR de la Tabla 7 y los comparamos con los BLEU mostrado en la Tabla 6, podemos establecer una leve correlación entre ambos valores. Esto se debe a que el BLEU muestra la calidad de la primera traducción y cuanto mayor sea esta, menos interacciones se requerirán con el usuario.

No obstante, esta correlación no es muy fuerte, por lo que entendemos que el WSR también depende de factores que no están reflejados en el BLEU.

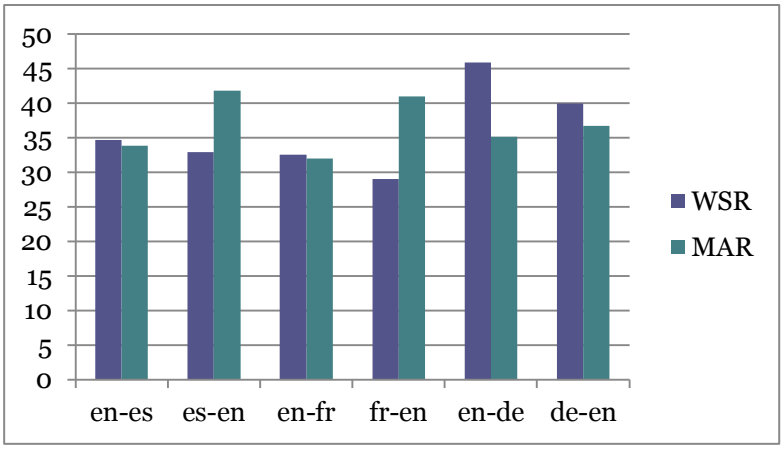


**Ilustración 17.** WSR y el MAR obtenidos para la tarea *EU*.

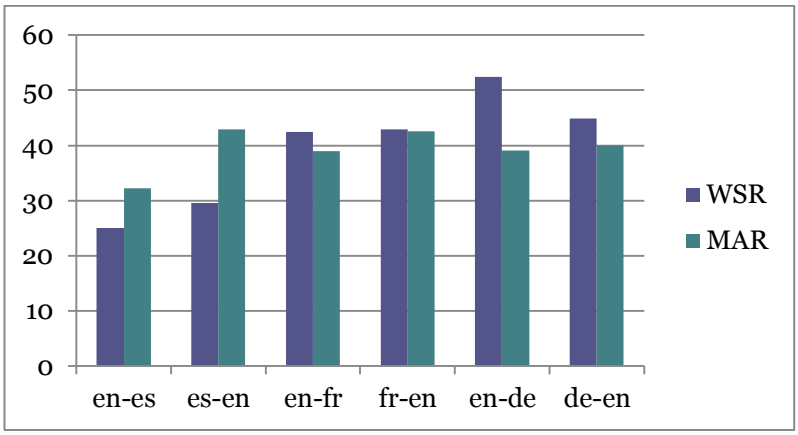


**Ilustración 18.** WSR y el MAR obtenidos para la tarea *Xerox*.

### *Tiempo de respuesta*

Si los comparamos con el Experimento 1 (Tabla 5), los tiempos de la Tabla 8 son significativamente más bajos. No obstante, muchos de ellos siguen siendo inaceptables, especialmente los de *EU*. Por ello, si se quisiera usar el sistema en la práctica habría que fijar un valor de $M$ de compromiso.

| | **Tiempo de respuesta (s)** | | | |
|---|---|---|---|---|
| | **EU** | | **Xerox** | |
| | **Media** | **Mediana** | **Media** | **Mediana** |
| En-Es | 2,47 | 0,90 | 0,61 | 0,04 |
| Es-En | 2,68 | 0,87 | 0,55 | 0,04 |
| En-Fr | 2,12 | 0,69 | 1,01 | 0,19 |
| Fr-En | 2,47 | 0,79 | 1,24 | 0,21 |
| En-De | 1,73 | 0,53 | 0,62 | 0,21 |
| De-En | 1,97 | 0,64 | 0,67 | 0,19 |

**Tabla 8.** Tiempo de respuesta medio y mediano para cada *corpus* empleando nuestro sistema y con $M = 100$. Se expresa en segundos.

Comparando las tareas, *Xerox* obtiene unos tiempos claramente más bajos a los de *EU* (Tabla 8), aunque su varianza también es significativamente más alta en algunos pares de lenguas (Ilustración 19). Esto puede achacarse al menor número de frases de entrenamiento de *Xerox*. También puede deberse a la menor longitud media por frase: tal y como se ha extraído del BLEU, la longitud parece estar relacionada con la ambigüedad de los idiomas, así que es posible que también afecte a los *corpus*.

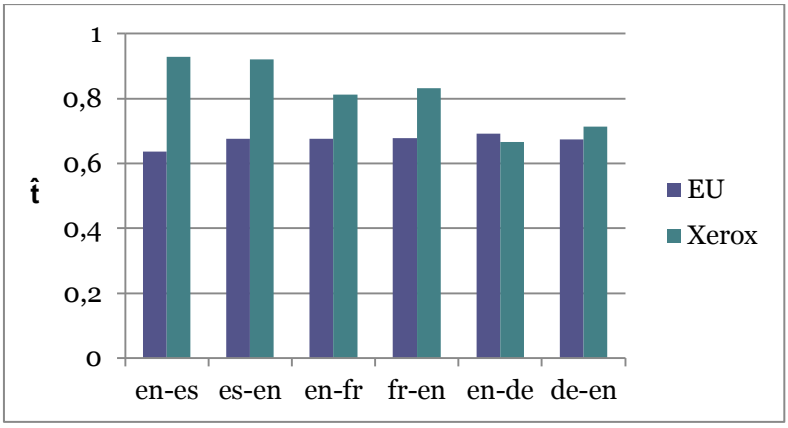


**Ilustración 19.** Métrica $\hat{t}$ para cada par de lenguas en los *corpus EU* y *Xerox*. Con ella, pretendemos estimar la varianza de tiempo de espera para cada una de estas ejecuciones.

Otra consideración interesante sobre el tiempo de respuesta es que está estrechamente relacionado con la complejidad de los hipergrafos, ya el algoritmo de búsqueda es más costoso si se aplica a estructuras complejas. Como se ha indicado en la sección 3.2, uno de los objetivos de la experimentación estudiar si el BLEU es adecuado para representar la complejidad, por lo que compararemos los valores de ambas métricas para este menester.

Como se aprecia en la Tabla 6, los BLEU obtenidos en el Experimento 2 están por debajo de los 44,2 puntos del Experimento 1 (apartado 3.6.1). Esto también se aplica a los tiempos: casi todos los del Experimento 2 (Tabla 8) están por debajo del tiempo que el Experimento 1 obtiene con $M = 40$ (Tabla 5), lo que parece indicar una correlación entre ambas métricas.

No obstante, si observamos los resultados dentro del Experimento 2, se pueden apreciar diferencias. Si bien los tiempos de *EU* corresponden a grandes rasgos con el BLEU, existen algunas discrepancias severas. La más notable es el hecho de que el par de lenguas *En-Fr* sea el que mejor BLEU obtiene, pero el cuarto más costoso temporalmente.

Esto es aún más notable en *Xerox*, en los que no parece haber ninguna relación entre ambas métricas salvo por el hecho de que ambas sitúan a los pares de lenguas alemanes por debajo del resto.

Así pues, los resultados indican que el BLEU no es una medida fiable para representar la complejidad del hipergrafo, si bien parece existir cierta relación entre ambas.

## 3.7 Discusión de los resultados

Para concluir este capítulo, se realizará una pequeña síntesis de los resultados mostrados en la sección 3.6 y se extraerán conclusiones globales.

A pesar de no haber obtenido tiempo real, los resultados del Experimento 1 indican una clara mejora con respecto a González-Rubio, et al. (2016), estando comprendida entre los 340 y los 560 puntos de mejora porcentual. Esta mejora ha sido bastante menos pronunciada en el tiempo de carga de los hipergrafos que en la ejecución del algoritmo de búsqueda. Por ello, de cara a futuras mejoras sería conveniente reducir el tiempo de carga, ya que de no hacerlo podría llegar a convertirse en el cuello de botella.

En el Experimento 1 también hemos observado que el comportamiento del sistema cuando se varía el valor de *M* para un mismo *corpus* es bastante previsible. Según indican los resultados, en estos casos existe una relación inversa entre el WSR por un lado y el tiempo y el MAR por otro.

No obstante, cuando comparamos pares de lenguas distintos (Experimento 2), estas características desaparecen, ya que la relación entre el WSR y el MAR deja de apreciarse por completo. Y si existe una relación inversa entre el WSR por un lado y el tiempo y el BLEU por otra, esta es muy leve.

La debilidad de esta correlación indica que el BLEU no es una métrica fiable de la calidad del hipergrafo cuando comparamos entre *corpus* distintos. No obstante, parece que sí lo es cuando comparamos el mismo *corpus* con parámetros de entrenamiento diferentes, de acuerdo con el BLEU, WSR y tiempo del Experimento 1 y del *corpus EU Es-En* del Experimento 2. Sería necesario estudiar esto con más profundidad, ya que estos resultados podrían no ser significativos, al haber sido realizados sobre una muestra de solo dos casos.

# Capítulo 4. Conclusiones

## 4.1 Conclusiones

A diferencia de los sistemas de Traducción Automática convencional, los sistemas interactivos pueden entenderse como un asistente de un traductor humano. Es su labor, por tanto, reducir el esfuerzo necesario por el usuario en esta tarea. Esto aplica especialmente a las palabras introducidas mediante teclado, ya que es la acción principal que desempeñaría un traductor sin uno de estos sistemas.

Otra consideración importante es el tiempo de respuesta del sistema, ya que de ser demasiado elevado, puede entorpecer la tarea del usuario en el mejor de los casos o volverlo totalmente inutilizable en el peor de estos. Además, el tiempo es también la principal mejora del sistema propuesto en nuestra tesis respecto al presentado por González-Rubio, et al (2016).

Comparado con la interacción basada en prefijos (el enfoque más tradicional de la IMT), el protocolo de interacción basado en Segmentos de Palabras puede interpretarse como una mejora o como un empeoramiento según cuál de estos factores consideremos. Por un lado, la libertad de validar palabras aunque no pertenezcan al prefijo reduce significativamente el esfuerzo de teclado del usuario. Pero por otro lado, complica los cálculos del sistema, aumentando el tiempo de respuesta.

A pesar de que el tiempo de espera se ha reducido considerablemente respecto al de González-Rubio, et al (2016), lo cierto es que si damos al usuario completa libertad para marcar palabras, éste sigue siendo inaceptable. No obstante, los experimentos han demostrado que es posible llegar a un punto de compromiso entre el tiempo y el esfuerzo, dependiendo del número máximo de palabras que dejemos validar al traductor humano.

Una consideración importante a la hora de encontrar el punto de compromiso es que este no puede determinarse *a priori*, ya que depende de diversos factores, los que destacamos la calidad del hipergrafo de traducción. Ésta se representará mediante el BLEU y afecta tanto al tiempo de espera (porque un hipergrafo menos complejo se explora con más rapidez) como al esfuerzo del usuario (ya que la baja calidad del hipergrafo repercute en la adecuación de las traducciones que éste obtenga).

A grandes rasgos, podemos concluir que, si bien no se ha alcanzado tiempo real con el protocolo sin límite de palabras, sí hemos logrado una reducción considerable del tiempo de espera. Así pues, empleando un límite de palabras dependiente de la facilidad de la tarea, podremos obtener un tiempo instantáneo y reducir el esfuerzo respecto al protocolo basado en prefijos.

## 4.2 Trabajo futuro

De acuerdo con los resultados extraídos, la mejora más evidente que se podría implementar en el sistema es reducir el tiempo de respuesta. Puesto que la principal sobrecarga viene dada por el algoritmo de búsqueda, la mayoría de mejoras deberían aplicarse en este ámbito.

Una posible mejora en este aspecto podría consistir en paralelizar el algoritmo, de modo que el algoritmo exploraría varios nodos a la vez. Para hacer esto, sería necesario explorar las dependencias entre ellos, de modo que no se exploren dos nodos con relación de parentesco al mismo tiempo. Una aproximación que permitiría cumplir esta restricción consistiría en crear un mapa de dependencias al cargar el hipergrafo, lo cual puede hacerse en $O(E \cdot N)$.

Otra forma de reducir el tiempo del algoritmo sería aplicando podas, esto es, dejar de explorar caminos que no sean prometedores. Un método que suele aplicarse en grafos es la técnica de *ramificación y poda*, a la que sería necesario realizar una serie de modificaciones para adecuarla a la estructura de hipergrafo.

Además de mejoras relacionadas con la búsqueda, también podría ser interesante reducir el tiempo de carga de los hipergrafos, ya que si bien son eventos más escasos que las interacciones, suelen presentar valores por encima de la mediana.

Para reducir este tiempo, sería posible implementar la lectura del fichero de hipergrafos mediante una Máquina de Estados Finitos, en lugar de mediante una expresión regular como se hace actualmente.

Por último, sería conveniente estudiar si el BLEU es una medida de calidad adecuada cuando comparamos el mismo *corpus* con parámetros de entrenamiento distintos. También convendría buscar métricas alternativas cuando la comparación se realice entre *corpus* distintos, ya que en este caso ha demostrado no ser fiable.

# Bibliografía

Aho, A. V. & Ullmann, J. D., 1969. Syntax directed translations and the pushdown assembler. *Journal of Computer and Systems Science,* pp. 37-56.

Alabau, V. y otros, 2013. *User evaluation of advanced interaction features for a computer-assisted translation workbench*. Nice, Machine Translation Summit XIV.

Azadi, F. & Khadivi, S., 2015. Improved Search Strategy for Interative Machine Translation in Computer-Assisted Translation. *Proceedings of Machine Translation Summit XV*.

Baker, J. K., 1979. Trainable grammars for speech recognition. *Speech Comunication ASA'97,* pp. 547-550.

Barrachina, S. y otros, 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics,* Issue 35, pp. 3-28.

Chiang, D., 2005. A Hierarchical Phrase-Based Model for Statistical Machine Translation. pp. 263-270.

Chiang, D., 2005b. An Introduction to Synchronous CFGs.

Foster, G., Isabelle, P. & Plamondon, P., 1998. Target-text mediated interactive machine translation. *Machine Translation,* I(12), pp. 175-194.

Gallo, G., Longo, G., Pallottino, S. & Nguyen, S., 1993. Directed hypergraphs and applications. *Discrete Applied Mathematics,* II(42), pp. 263-270.

Gao, Q. & Stephan, V., 2008. Parallel implementations of word alignment tool. *Proceedings of the ACL 2008 Software Engineering, Testing and Quality Assurance Workshop,* pp. 49-57.

González-Rubio, J., Ortiz-Martinez, D., Benedí, J. M. & Casacuberta, F., 2013. Interactive machine translation using hierarchial translation models. *Procedings of the conference on Empirical Methods in Language Processing,* pp. 144-254.

González-Rubio, J., Ortiz-Martínez, D., Benedí, J. M. & Casacuberta, F., 2016. Beyond Prefix-Based Interactive Machine Translation. *Proceedings of the 20th SIGNLL Conference in Computational Natural Language Learning (CoNLL),* pp. 7-12.

González-Rubio, J., Ortiz-Martínez, D. & Casacuberta, F., 2010. Balancing user effort and translation error in interactive machine translation via confidence mesures. *Procedings of the ACL 2010 Conference,* pp. 173-177.

Hutchins, J., 1995. *Machine Translation: A Brief History. Concise history of the language sciences: from Sumerians to the cognitivists.*. s.l.:Pergamon Press.

Koehn, P., 2009. A Process of Study of Computer Aided Translation. *Machine Translation,* IV(23), pp. 241-263.

Koehn, P., 2014. Refinements to interactive translation prediction based on search graphs. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics,* pp. 574-578.

Koehn, P., Cjara, T. & Saint-Amand, H., 2014. Refinements to interactive translation prediction based on search graphs. *Procedings of the 52nd Annual Meeting of the Association for Computational Linguistics,* pp. 574-578.

Kohen, P. y otros, 2007. Moses: Open Source Toolkit for Statistical Machine Translation. *Proceedings of the 45th Annual Meeting of the ACL,* pp. 177-180.

Langlais, P. & Lapalme, G., 2002. TransType: development-evaluation cycles to boost translator's productivity. *Machine Translation,* II(17), pp. 77-98.

Levenshtein, V., 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady,* 8(10), pp. 707-710.

Martínez-Gómez, P., Sanchís-Trilles, G. & Casacuberta, F., 2012. Online adaptation strategies for statistical machine translation in. *Pattern Recognition.*

Nielsen, J., 1993. *Usability Engineering*. s.l.:Morgan Kaufmann Publishers, Inc..

Och, F. J. & Ney, H., 2003. A systematic comparison of various statistical alignment models. *Computational Linguistic,* I(29), pp. 19-51.

Ortiz-Martínez, D., 2011. *Advances in Fully-Automatic and Interactive Phrase-Based Statistical Machine Translation*. València: Universitat Politècnica de València.

Ortiz-Martinez, D., 2016. Online Learning for Statistical Machine Translation. *Computational Linguistics,* 42(1), pp. 121-161.

Papineni, K., Roukos, S., Ward, T. & Zhu, W.-J., 2002. BLEU: A Method for Authomatic Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics,* pp. 311-318.

Sanchis-Trilles, G. y otros, 2014. Interactive translation prediction versus conventional post-editing in practice: a study with the casmacat workbench. *Machine Translation,* III(28), pp. 217-235.

Sanchis-Trilles, G. y otros, 2008. Improving Interactive Machine Translation via Mouse Actions. *Proceedings of the Conference on Empirical Methods in Natural Language Processing,* pp. 485-494.

Stolcke, A., 2002. SRILM - an extensible language modeling toolkit. *Speech Technology and Research Laboratory SRI International,* pp. 257-286.

Tomás, J. & Casacuberta, F., 2006. Statistical phrase-based models for interactive computer-assisted translation. *Proceedings of the COLING/ACL 2006,* pp. 835-841.

Zens, R., Och, F. J. & Ney, H., 2002. Phrase-Based Statistical Machine Translation. *Advances in Artificial Inteligence,* 2479(25), pp. 18-32.

# Anexo I. Ficheros de hipergrafos

El *software* de traducción *Moses* tiene la opción de generar hipergrafos correspondientes a cada una de las frases traducidas. Dado que es una opción sencilla y eficaz para obtener estas estructuras, hemos decidido emplear este *software* para ello.

Para generar un fichero de hipergrafos correspondientes a las frases de test, será necesario entrenar un modelo jerárquico (*Moses Chart*). Tras ello, se añade la opción *–output-search-graph* (también abreviada como *–osg*) a la instrucción para realizar la fase de test.

Los hipergrafos se almacenan en ficheros de texto, en los que cada línea representa una arista. Puesto que el formato también sirve para representar grafos de búsqueda, éstas contienen algunos datos que deben ser modificados o incluso ignorados.

El formato introduce dos tipos de línea: las tipo *IDX* y las tipo *IDX->REC*. El primer tipo introduce un nuevo nodo *IDX* junto con la arista, siendo éste la cabeza de la misma. El segundo tipo inserta una nueva arista a un nodo *REC* ya existente.

```
0 259238 X  -> amount of :: term=0-0 1-1 : nonterm=: c=-4.13657 core=(0,-2,1,-5.18019,-5.7831,-5.8306,-6.48508,0,
0)  [1..2] [total=-4.13657] core=(0,-2,1,-5.18019,-5.7831,-5.8306,-6.48508,0,-10.3091)
0 259311->259238 X  -> amount X :1-1 : term=0-0 : nonterm=1-1 : c=-3.90673 core=(0,-1,1,-6.01595,-5.50058,-6.45
835,-5.60642,0,0)  [1..2] 249735 [total=-4.53624] core=(0,-2,2,-6.15641,-5.7831,-6.83069,-6.48508,0,-10.3091)
```

**Ilustración 20.** Ejemplos de líneas tipo *IDX* (superior) y tipo *IDX->REC* (inferior).

La información útil que contiene cada línea es la siguiente:

- Índice del hipergrafo al que pertenece. Útil para determinar cuándo se cambia de frase.
- Identificador del nodo padre de la arista. En los nodos *IDX* se corresponde con *IDX*. En los nodos *IDX->REC* se corresponde con *REC*, mientras que *IDX* debe ser ignorado.
- Identificador de los nodos hijo de la arista, hasta un máximo de dos. Esto incluye a las aristas sin hijos, por ejemplo las que inciden en nodos hoja. En cualquier caso, estos nodos siempre han sido introducidos previamente en una línea *IDX*.
- Parte derecha de la regla de la gramática equivalente (ver apartado 2.3.2). Las cadenas *S* y *X* están reservadas para símbolos no terminales, esto es, que pueden derivarse mediante otras reglas.
- *Inside log-score* de los nodos nuevos, en el caso de las líneas *IDX* (ver apartado 2.3.2). También sirve para calcular el log-score de la arista, restándole el *inside log-score* de los hijos.

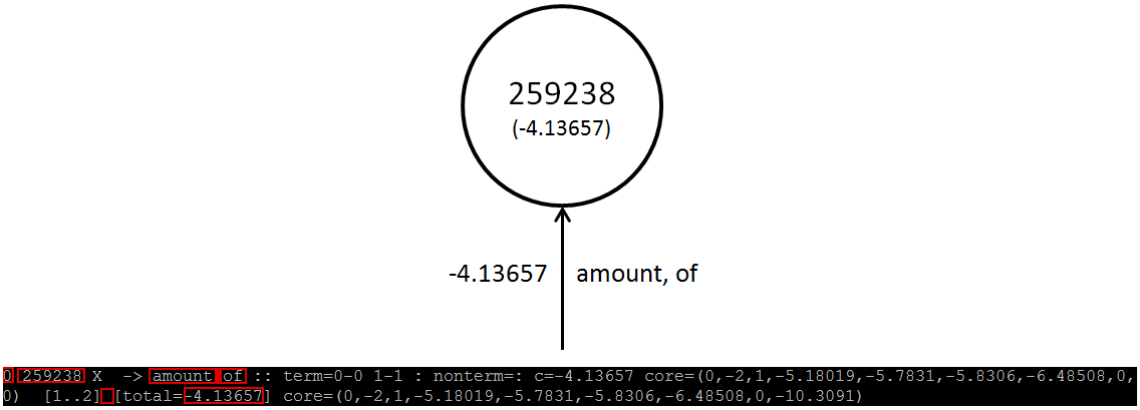


**Ilustración 21.** Línea tipo *IDX* que aparece en la Ilustración 20 en la que se resalta su información útil, junto una ilustración de lo que ésta representa.
En este caso, añade al hipergrafo con índice 0 el nodo con índice 259238 y una arista sin hijos incidente en él. También añade una arista cuya RHS es *amount of*. Tanto el *inside log-score* del nodo y el *log-score* de la arista equivalen a −4,13657, equivalencia que se produce porque la arista carece de hijos.

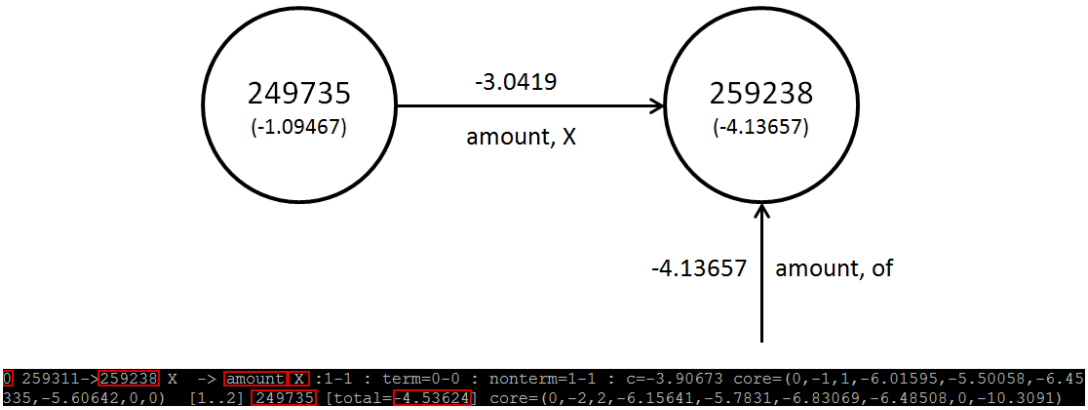


**Ilustración 22.** Línea tipo *IDX -> REC* que aparece en la Ilustración 20 en la que se resalta su información útil, junto una ilustración de lo que ésta representa.
Esta línea crea una arista en el nodo introducido en la Ilustración 21. Ésta parte del nodo 249735, que también ha sido introducido con anterioridad. El RHS de esta es *amount X*, siendo *X* un símbolo no terminal.
El *log-score* de la arista se calcula como el valor que figura tras la cadena "*total*" menos el *log-score* del nodo hijo; en este caso, es: $-4.54 - (-1.09) = -3.04$.

Así pues, los pasos del algoritmo para procesar un fichero completo son los siguientes:

1. Crear una estructura de hipergrafo vacía.
2. Procesar las líneas del fichero, añadiendo la información de cada una de ellas a la estructura creada en el paso uno. Este paso se repetirá hasta llegar a una línea perteneciente a otro hipergrafo o al final del fichero.
3. Realizar el proceso de IMT con la frase correspondiente al hipergrafo.
4. Borrar el hipergrafo.
5. Si no se ha llegado al final del fichero, volver al paso 1.

# Anexo II. Algoritmo de Búsqueda

En este anexo se adjunta la definición matemática del algoritmo. También incluye aclaraciones sobre el mismo, así como una estimación de su complejidad temporal.

Procedimiento NuevaHipótesis($H, \boldsymbol{f}$):
  $\boldsymbol{M}_t \coloneqq \boldsymbol{\lambda}_{|N|,|C|}$
  $\boldsymbol{M}_P \coloneqq \boldsymbol{0}_{|N|,|C|}$
  para cada $n \in N$:
    $\boldsymbol{M}_t(n, c_\lambda) \coloneqq \boldsymbol{t}(n)$
    $\boldsymbol{M}_P(n, c_\lambda) \coloneqq P_I(n)$
    para cada $\boldsymbol{f}_i \in \boldsymbol{f}$:
      $\boldsymbol{a} \coloneqq$ Levenshtein$(\boldsymbol{t}(n), \boldsymbol{f}_i)$
      si $P(\boldsymbol{f}_i, \boldsymbol{a} | \boldsymbol{t}(n)) > \boldsymbol{M}_P(n, c_{ii})$:
        $\boldsymbol{M}_t(n, c_{ii}) \coloneqq$ Sustituir$(\boldsymbol{a}, \boldsymbol{t}(n), \boldsymbol{f}_i)$
        $\boldsymbol{M}_P(n, c_{ii}) \coloneqq P(\boldsymbol{f}_i, \boldsymbol{a} | \boldsymbol{t}(n))$
      finsi
    finpara
    para cada $e \in \mathrm{E}_s(n) \mid S_e > 0$:
      para cada $\boldsymbol{c}_{ij} = \{c_1, \dots, c_{S_e}\} \in$ CombinaciónCoberturas$(N_s(e))$:
        $\boldsymbol{T}_{\boldsymbol{c}_{ij}} \coloneqq \{\boldsymbol{M}_t(n_{s,1}(e), c_1), \dots, \boldsymbol{M}_t(n_{s,S_e}(e), c_{S_e})\}$
        $P_e \coloneqq P(e) \prod_{k=1}^{S_e} \boldsymbol{M}_P(n_{s,k}(e), c_k)$
        si $P_e > \boldsymbol{M}_P(n, c_{ij})$:
          $\boldsymbol{M}_t(n, c_{ij}) \coloneqq$ DerivarRegla(RHS$(e)$, $\boldsymbol{T}_{\boldsymbol{c}_{ij}}$)
          $\boldsymbol{M}_P(n, c_{ij}) \coloneqq P_e$
        finsi
        si $i > 1$:
          $\boldsymbol{a}_i \coloneqq$ Levenshtein$(\boldsymbol{t}(n), \boldsymbol{f}_{i-1})$
          si $P(\boldsymbol{f}_{i-1}, \boldsymbol{a}_i | \boldsymbol{t}(n)) > \boldsymbol{M}_P(n, c_{i-1,j})$:
            $\boldsymbol{M}_t(n, c_{i-1,j}) \coloneqq$ Sustituir$(\boldsymbol{a}_i, \boldsymbol{t}(n), \boldsymbol{f}_{i-1})$
            $\boldsymbol{M}_P(n, c_{i-1,j}) \coloneqq P(\boldsymbol{f}_{i-1}, \boldsymbol{a}_i | \boldsymbol{t}(n))$
          finsi
        finsi
        si $j < |\boldsymbol{f}|$:
          $\boldsymbol{a}_j \coloneqq$ Levenshtein$(\boldsymbol{t}^b(n), \boldsymbol{f}^b_{j+1})$
          si $P(\boldsymbol{f}_{j+1}, \boldsymbol{a}_j | \boldsymbol{t}(n)) > \boldsymbol{M}_P(n, c_{i,j+1})$:
            $\boldsymbol{M}_t(n, c_{i,j+1}) \coloneqq$ Sustituir$(\boldsymbol{a}_j, \boldsymbol{t}(n), \boldsymbol{f}_{j+1})$
            $\boldsymbol{M}_P(n, c_{i,j+1}) \coloneqq P(\boldsymbol{f}_{j+1}, \boldsymbol{a}_j | \boldsymbol{t}(n))$
          finsi
        finsi
      finpara
    finpara
  finpara
  devolver $(\boldsymbol{M}_t(|N|, |C|), \boldsymbol{M}_P(|N|, |C|))$

Las variables y funciones que se muestran en el algoritmo, por orden de aparición son:

- Hipergrafo $H$. Sobre él se realiza la búsqueda
- Realimentación del usuario $\boldsymbol{f}$. Se define como $\boldsymbol{f} = \{\boldsymbol{f}_1, \dots, \boldsymbol{f}_{|f|}\}$, donde cada $\boldsymbol{f}_i$ es un segmento de palabras.
- Matriz de traducciones $\boldsymbol{M}_t$. Almacena la traducción más probable para nodo $n$ teniendo en cuenta los segmentos cubiertos por cada cobertura $c$.
- Matriz de probabilidades $\boldsymbol{M}_P$. Para cada celda $(n, c)$, almacena la probabilidad asociada a la traducción de $\boldsymbol{M}_t(n, c)$.
- Conjunto de nodos $N$ pertenecientes a $H$. Sobre este conjunto, iteraremos para todos los nodos $n$.
- Cadena de texto $\boldsymbol{t}(n)$. Es la hipótesis más probable *a priori* del nodo $n$.
- Cobertura vacía $c_\lambda$.
- Alineamiento $\boldsymbol{a}$ entre $\boldsymbol{t}(n)$ y un segmento validado $\boldsymbol{f}_i$. Se calcula mediante la distancia de edición de Leveshtein.
- Conjunto de aristas hijas $E_s(n)$ incidentes en el nodo $n$. Sobre este conjunto, iteraremos para todas sus aristas $e$.
- Nodos hijo $N_S(e)$ de la arista $e$. La talla de esta se representa como $S_e$. El i-ésimo hijo de esta se denota como $n_{si}(e)$.
- Lista de coberturas $\boldsymbol{c}_{ij}$. Contiene una cobertura $c_i$ explorada por cada hijo $n_{si}(e)$. El resultado de combinarlas debe ser una cobertura válida.
- Lista de cadenas de texto $\boldsymbol{T}_{\boldsymbol{c}_{ij}}$. Para cada $i \mid 1 \leq i \leq N_S(e)$ Almacena la traducción más probable de cada nodo hijo, $n_{si}(e)$ teniendo en cuenta los segmentos cubiertos por la cobertura $c_i \in \boldsymbol{c}_{ij}$.
- Probabilidad $P_e$. Está asociada a la cadena resultante de combinar la RHS de $e$ con las cadenas almacenadas en $\boldsymbol{T}_{\boldsymbol{c}_{ij}}$ (procedimiento DerivarRegla).
- Alineamiento $\boldsymbol{a}_i$ entre $\boldsymbol{t}(n)$ y el segmento extendido a izquierda $\boldsymbol{f}_{i-1}$.
- Alineamiento $\boldsymbol{a}_j$ entre $\boldsymbol{t}^b(n)$ y el segmento extendido a derecha $\boldsymbol{f}^b_{j+1}$. El superíndice $b$ indica que las cadenas están en orden inverso.

Los procedimientos auxiliares empleados son los siguientes:

- $\mathrm{Leveshtein}(t, f)$. Calcula un alineamiento entre las cadenas $t$ y $f$, de acuerdo con la distancia de edición de Levenshtein.
- $\mathrm{Sustituir}(a, t, f)$. Devuelve la cadena de texto resultante de sustituir por $f$ el fragmento de $t$ indicado por el alineamiento $a$.
- $\mathrm{CombinaciónCoberturas}(N)$. Obtiene una lista de combinaciones de coberturas exploradas por los nodos pertenecientes a $N$. Cada combinación contendrá $|N|$ coberturas parciales, de modo que al unirlas en orden de aparición se formará una cobertura válida (definida en el apartado 2.3.3).
- $\mathrm{DerivarRegla}(r, T)$. Sustituye los símbolos no terminales de la regla $r$ por cada una de las cadenas contenidas en $T$. Requiere que $r$ tenga exactamente $|T|$ símbolos no terminales.