Procesamiento del Lenguaje Natural, núm. 35 (2005), pp. 85-92 recibido 29-04-2005; aceptado 01-06-2005

Algoritmo de Decodificación de Traducción Automática Estocástica basado en N-gramas

Josep M. Crego **José B. Mariño** **Adrià de Gispert**

Centro de Investigación TALP
Campus Nord UPC, 08034-Barcelona
{jmcrego,canton,agispert}@gps.tsc.upc.es

Resumen: En esta comunicación se presenta MARIE, un algoritmo de decodificación para un sistema de traducción automática estocástica basado en N-gramas. Para su implementación se utiliza una estrategia de búsqueda en haz, con capacidad para realizar reordenamientos (distorsión). El modelo de traducción está basado en N-gramas bilingües, ampliado para introducir reordenamientos en las cadenas de palabras. La estructura del espacio de búsqueda permite realizar un alto grado de poda, incrementando así la eficiencia del algoritmo.
Palabras clave: traducción automática estocástica, modelos de traducción basados en N-gramas, algoritmos de decodificación

Abstract: In this paper we describe MARIE, an N-gram-based stochastic machine translation decoder. It is implemented using a beam search strategy, with distortion (or reordering) capabilities. The underlying translation model is based on an N-gram approach, extended to introduce reordering at the phrase level. The search graph structure is designed to perform very accurate comparisons, what allows for a high level of pruning, improving the decoder efficiency.
Keywords: stochastic machine translation, N-gram-based translation models, decoding algorithms

1. Introducción

La traducción automática estocástica (TAE), se define como una tarea donde cada oración fuente f_1^J se transforma (o genera) en una oración destino d_1^I, a través de un proceso estocástico.

La traducción de una oración fuente puede formularse como la búsqueda de la oración destino que maximiza la probabilidad condicional $p(d_1^I|f_1^J)$, que usando la regla de Bayes puede reescribirse como:

$$\arg\max_{d_1^I} \left\{ p(f_1^J|d_1^I) \cdot p(d_1^I) \right\} \qquad (1)$$

donde $p(f_1^J|d_1^I)$ representa el modelo de traducción y $p(d_1^I)$ es el modelo de lenguaje del idioma destino.

La descomposición del problema en dos fuentes sigue el enfoque llamado 'modelo de canal ruidoso'. La operación de maximización (argmax) denota el problema de búsqueda.

En este punto podemos describir la tarea de traducción automática como el proceso de búsqueda de las palabras del idioma destino que maximizan conjuntamente dos objetivos: (primero) encontrar las palabras que mejor traducen las palabras de la oración fuente. Y (segundo) encontrar la secuencia de palabras que genera una oración destino correcta. Para el primero se utiliza un modelo de traducción, que indica para cada par de palabras (fuente y destino) la probabilidad de que una sea traducción de la otra. Para el segundo se utiliza un modelo de lenguage (destino), que indica para cada oración (o secuencia) la probabilidad de que ésta pertenezca al idioma destino.

En referencia al modelo de traducción, los primeros sistemas de traducción automática trabajaban a nivel de palabras (Brown et al., 1990) (las unidades bilingües se componian de palabras aisladas). Los algoritmos de decodificación de estos primeros sistemas pueden clasificarse bajo diferentes aproximaciones: búsqueda A^* (Och, Ueffing, y Ney, 2001), programación entera (Germann et al., 2001), algoritmos voraces (Germann, 2003), (Berger et al., 1994), (Wang y Waibel, 1998).

Recientemente, los sistemas de TAE tien-

ISSN: 1135-5948

den a utilizar secuencias de palabras como unidades básicas del modelo de traducción, con el objetivo de introducir el contexto en dicho modelo. Estos sistemas llevan a cabo la traducción mediante la maximización de una combinación lineal de los logaritmos de la probabilidad asignada a la traducción por el modelo de traducción y otras características, siguiendo la aproximación por máxima entropía, (Berger, Della Pietra, y Della Pietra, 1996) (facilitando así la introducción de modelos adicionales):

$$\arg\max_{d_1^I}\{\exp(\sum_i \lambda_i h_i(d,f))\} \qquad (2)$$

donde cada función característica h_i consiste en el logaritmo de la probabilidad asignada por cada uno de los modelos (traducción, lenguaje destino, distorsión, etc.), los pesos λ_i se ajustan optimizando una función de evaluación.

Diversos autores han demostrado que la utilización de un modelo de distorsión, el cual permite realizar reordenamientos de secuencias de palabras, permite mejorar los resultados de traducción para algunos pares de lenguas. Estos sistemas se ven forzados a restringir sus habilidades de distorsión debido al alto coste computacional que suponen. En (Knight, 1999), se ha clasificado el problema de la decodificación, cuando se permite distorsión, en el grupo de los problemas NP-completos.

En (Tillmann y Ney, 2000) y (Och y Ney, 2004) se describen dos decodificadores con habilidad para realizar reordenamientos. En (Koehn, 2004) se puede encontrar un decodificador basado en secuencias de palabras que realiza reordenamientos y se puede descargar de internet de manera gratuita.

Esta comunicación enfoca el problema de la decodificación en TAE cuando se permite distorsión, bajo un modelo de traducción basado en N-gramas, ver (de Gispert y Mariño, 2002). Está organizado de la siguiente manera: en la sección 2 se introducen las particularidades del modelado de la traducción utilizado por el decodificador, en la sección 3 se describen las características del decodificador. En la sección 4 se muestran varios experimentos que evalúan el rendimiento del decodificador. Finalmente, en la sección 5 se presentan las conclusiones y se describen líneas de investigación futuras.

2. Modelo de Traducción basado en N-gramas

De acuerdo con la ecuación 1, la traducción puede también verse como el proceso estocástico que maximiza la probabilidad conjunta:

$$\arg\max_{d_1^I}\{\, p(d_1^I, f_1^J)\} \qquad (3)$$

El modelo de traducción puede entenderse como un modelo de lenguaje de unidades bilingües (llamadas tuplas). Dichas tuplas, definen una segmentación monónotona de los pares de oraciones utilizadas en el entrenamiento del sistema (f_1^J, e_1^I), en K unidades $(t_1, ..., t_K)$.

En la extracción de las unidades bilingües, cada par de oraciones da lugar a una secuencia de tuplas que sólo dependerá de los alineamientos internos entre las palabras de la oración. En (Picó, Tomás, y Casacuberta, 2004) se describe una metodologia de traducción por estados finitos basada en alineamientos.

La mayoria de sistemas basados en secuencias de palabras utilizan una unidad bilingüe parecida (habitualmente llamada *'phrase'*, en inglés), que no contempla la idea de segmentación del par de oraciones (Och y Ney, 2004). En la extracción de estas unidades bilingües se utilizan los alineamientos internos entre palabras del par de oraciones con el objetivo de conseguir el máximo número de unidades (*phrases*).

Tal y como se han definido las tuplas y las *phrases*, el conjunto de tuplas consiste en un subconjunto del conjunto de *phrases*.

La figura 1 muestra un ejemplo de extracción de tuplas y *phrases* a partir de un par de oraciones alineadas palabra a palabra.

En la traducción de una oración de entrada, el decodificador debe encontrar la secuencia de tuplas asociada a una segmentación de la oración de entrada que produzca probabilidad máxima. Tal probabilidad máxima, se calcula como combinación lineal de los modelos utilizados en el sistema de traducción (maximización indicada en la ecuación 2).

El modelo de traducción esta implementado utilizando un modelo de lenguaje (bilingüe) basado en N-gramas (B), (con $N = 3$):

$$p(d,f) = Pr(t_1^K) = \prod_{k=1}^{K} p(t_k \,|\, t_{k-2}, t_{k-1}) \quad (4)$$

Usando el enfoque log-lineal de la ecuación 2, el decodificador utiliza cuatro funciones caracteristicas (definidas como probabilidades):

- Un modelo de traducción calculado utilizando las probabilidades lexicas del modelo IBM1, para ambas direcciones (I):

$$Pr(t_1^K) = \prod_{i=1}^{K} pI(d_k \mid f_k)^{\lambda_1} pI'(f_k \mid d_k)^{\lambda_2} \tag{5}$$

  donde cada peso de modelo (λ_1 y λ_2) se ajusta optimizando una función de evaluación, de manera conjunta con el resto de coeficientes de los modelos.

- Un modelo de lenguaje basado en N-gramas del idioma destino (T), (con $N = 3$):

$$Pr(d_1^I) = \prod_{i=1}^{I} p(d_i \mid d_{i-2}, d_{i-1}) \tag{6}$$

- Una penalización basada en el número de palabras de la traducción, usada para compensar la preferencia del decodificador por las traducciones cortas (P):

$$Pr(d_1^I) = \exp(I) \tag{7}$$

- Un modelo de distorsión basado en la distancia entre palabras (R):

$$Pr(t_1^K) = \exp(-\sum_{k=1}^{K} dist_k) \tag{8}$$

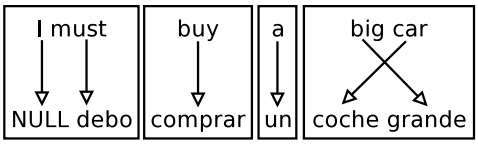
  donde $dist_k$ indica la distancia entre la primera palabra de la tupla K, y la última palabra $+1$ de la tupla $K-1$ (distancias medidas en número de palabras). Véase que para dos tuplas consecutivas la distancia resultante es $dist = 0$.

Se pueden introducir nuevos modelos extendiendo el sumatorio de la ecuación 2 con funciones caracteristicas adicionales.

3. Decodificador

En TAE, los decodificadores construyen la oración destino (traducción) de manera incremental (de izquierda a derecha) en forma de hipótesis, permitiendo discontinuidades en la oración fuente.

En el proceso de búsqueda del camino óptimo, se utiliza un algoritmo de búsqueda



PHRASES:
I must # debo
I must buy # debo comprar
must # debo
must buy # debo comprar
must buy a # debo comprar un
buy # comprar
buy a # comprar un
a # un
a big car # un coche grande
big # grande
big car # coche grande
car # coche

TUPLAS:
I must # debo
buy # comprar
a # un
big car # coche grande

Figura 1: *Extracción de tuplas y phrases a partir de un par de oraciones alineadas palabra a palabra. La segmentación de cada par de oraciones se lleva a cabo utilizando el alineamiento de Viterbi, ver (Crego, Mariño, y de Gispert, 2004). Como puede verse, el conjunto de tuplas resultantes son un subconjunto del conjunto de phrases.*

'en haz', con poda. Los algoritmos de búsqueda 'en haz' suelen ser los mas utilizados en el problema de la decodificación en TAE.

La búsqueda se realiza construyendo traducciones parciales (hipótesis), que se conservan en una o varias listas. Las listas son posteriormente podadas, de acuerdo con la probabilidad acumulada de las hipótesis que contienen. Las peores hipótesis, con menor probabilidad, se descartan con el objetivo de hacer eficiente la búsqueda.

Uno de los mayores problemas al construir un decodificador de TAE consiste en la obligatoriedad de podar la búsqueda. Algunos de los decodificadores que utilizan un algoritmo de busqueda 'en haz' utilizan una sola lista, donde todas las hipótesis compiten para sobrevivir al proceso de poda. Utilizar una sola lista hace que la manera como se comparan (ordenan) las hipótesis en la lista tenga una importancia capital, dado que la búsqueda tiende a mantener vivas aquellas hipótesis (traducciones parciales) que en los primeros pasos de la búsqueda obtienen mayor probabilidad (a pesar de no ser siempre las mejores).

Para evitar este problema suele utilizarse una función heurística, que para cada hipótesis ofrece una estimación de la probabilidad

del mejor camino que completa la traducción, permitiendo descartar aquellas hipótesis con valoración pesimista respecto al camino futuro y transformando el algoritmo en una búsqueda de tipo A^*. Un problema asociado a la utilización de función heurística consiste en la dificultad de encontrar buenas estimaciones, con coste computacional aceptable, ver (Och, Ueffing, y Ney, 2001), (Och y Ney, 2004) and (Koehn, 2004).

Varias listas, en vez de una única, pueden utilizarse, ayudando al proceso de poda (siendo más justa la comparación efectuada). Normalmente, las hipótesis se guardan en diferentes listas, dependiendo del número de palabras fuente o destino que la traducción lleva acumuladas. Nuestro decodificador sigue la aproximación multi-lista sin utilizar función heurística.

3.1. Algoritmo y estructura del grafo de búsqueda

La búsqueda empieza añadiendo un estado inicial, donde ninguna de las palabras fuente estan aún cubiertas (traducidas).

Se van añadiendo nuevos estados (hipótesis), utilizando las tuplas disponibles que cubren las palabras de la oración fuente aun sin cubrir. Cada hipótesis utiliza un vector de cobertura para indicar qué palabras de la oración fuente estan ya traducidas (por ejemplo, una hipótesis con vector de cobertura '11000' indica que las dos primeras palabras de la oración fuente ya se han traducido).

La expansión de una hipótesis permite cubrir (traducir) cualquiera de las palabras de la oración fuente (con la restricción de que las palabras cubiertas en la expansión considerada sean consecutivas y que no hayan sido ya traducidas). Las palabras destino de la tupla utilizada en una expansión se añaden secuencialmente a la oración destino, construyendo la traducción de manera monótona.

El coste de cada nuevo estado resulta de sumar al coste del estado predecesor, el coste derivado de los diferentes modelos.

La Tabla 1 indica la información contenida en cada estado.

La figura 2 muestra un ejemplo de la estructura del grafo de búsqueda. El grafo puede descomponerse en tres niveles:

- Hipótesis. En la figura 2, representados mediante el símbolo '*'.
- Listas. En la figura 2, las cajas con la etiqueta correspondiente al vector de cobertura. Cada lista contiene un conjunto ordenado de hipótesis (todas las hipótesis de una lista han traducido las mismas palabras de la oración fuente).
- Grupos (de listas). En la figura 2, delimitadas con una linea discontinua. Cada grupo contiene un conjunto ordenado de listas, correspondiente a las listas de hipótesis que han traducido el mismo número de palabras de la oración fuente (para ordenar las listas se utiliza el coste de la mejor hipótesis de cada una de ellas). Cuando la búsqueda esta restringida a traducciones monótonas, sólo se permite una lista para cada grupo.

Un enlace al estado predecesor
Las últimas N_1 tuplas
Las últimas N_2 palabras destino
El vector de cobertura
Posiciones cubiertas por la última tupla
El coste asociado al estado

Cuadro 1: *Cada hipótesis se representa utilizando seis campos de información. Cada N_i se fija de acuerdo a los N-gramas usados en los modelos de traducción (B) y de lenguaje destino (T).*

En la figura 2, las flechas unen a cada hipótesis con su predecesora. De manera que cada hipótesis tendrá una única flecha de salida (que la unirá a su predecesora) y varias hipótesis de entrada (sucesoras).

La búsqueda itera expandiendo las hipótesis disponibles. La expansión se produce de manera incremental, empezando por el grupo de listas que cubren una palabra de la oración fuente, acabando en el grupo de listas que cubren $J-1$ palabras (J es el número de palabras de la oración fuente).

Entre las hipótesis guardadas en el último grupo (aquellas que cubren todas las palabras de la oración fuente), se elige la de menor coste (mayor probabilidad), siendo la hipótesis que contiene la traducción ganadora.

3.2. Poda del grafo de búsqueda

Aquellas hipótesis que coinciden en las últimas N_1 tuplas, el vector de cobertura y las últimas N_2 palabras destino son recombinadas (N_1 y N_2 se fijan de acuerdo con el orden N de los modelos N-gramas de traducción (B) y lenguaje del idioma destino (T)),

de manera que se mantienen sólo las mejores hipótesis. La recombinación consiste en una técnica de poda sin riesgo para la búsqueda, ya que aquellos estados que se van a recombinar no pueden distinguirse en futuros pasos del proceso de búsqueda.

Para cada lista, sólo sus mejores hipótesis son expandidas:

- Las (b) hipótesis con menor coste (poda por histograma);
- con un coste comprendido entre el coste de la mejor hipótesis y un margen (t) (poda por umbral).

Las mismas estrategias de poda se utilizan en la expansión de las listas de cada grupo:

- Se expandirán las (B) listas con menor coste (poda por histograma);
- con un coste comprendido entre el coste de la mejor lista y un margen (T) (poda por umbral).

Cuando en la búsqueda se utiliza la caracteristica de distorsión, el decodificador sufre de la aparición de una gran cantidad de listas (con límite superior 2^J, siendo J el número de palabras de la oración fuente). No sólo los estados de cada lista, sino que también las listas deben ser podadas. Para reducir el número de listas, se utilizan dos restricciones, que se detallan en la siguiente subsección.

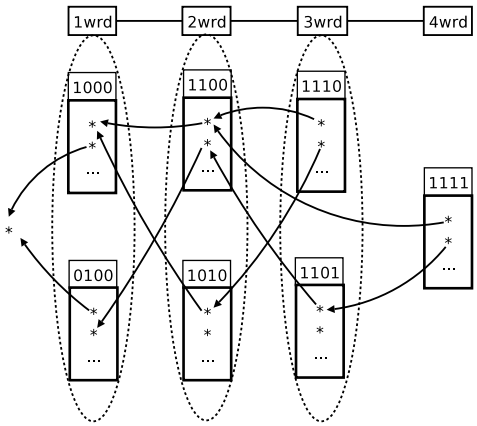


Figura 2: *Grafo de búsqueda correspondiente a una oración fuente con cuatro palabras. El grafo esta limitado por el número máximo de distorsiones permitidas en cada oración ($j = 1$), así como la distancia máxima que se permite reordenar una palabra (o secuencia de palabras ($m = 1$). Los detalles de estas restricciones pueden encontrarse en las siguientes secciones.*

3.3. Distorsión

Una buena estrategia para limitar los reordenamientos resulta clave para reducir el problema derivado de la explosión combinatoria del espacio de búsqueda. En (Zens, Och, y Ney, 2004), se muestra una comparativa entre varias restricciones de reordenamiento, llamadas restricciones ITG (Wu, 1995) y IBM (Berger, Della Pietra, y Della Pietra, 1996).

Nuestro decodificador implementa dos restricciones con el objetivo de reducir las listas de la estructura del espacio de búsqueda:

- Un límite de distorsión (m). Una palabra de la oración fuente (o secuencia de palabras) sólo podra ser reordenada una distancia máxima (medida en palabras).
- Un limite de reordenamientos (j). Una traducción sólo puede contener un número máximo de reordenamientos j.

4. Experimentos

En esta sección se muestran varios experimentos que evalúan el rendimiento del decodificador, usando BLEU (Papineni et al., 2001) y WER (Word Error Rate). Inicialmente se detallan las condiciones en que los experimentos fueron realizados.

4.1. Corpus

Los experimentos se efectuaron utilizando dos bases de datos:

- El corpus TC-Star[1] (español-inglés) y
- el corpus IWSLT 2004 BTEC[2] (chino-inglés).

Con el corpus TC-Star, se utilizo la versión de texto FTE (Final Text Edition), empleada en la primera evaluación SLT (Spoken Language Translation) del proyecto TC-Star.

Las tablas 2 y 3 muestran las estadísticas básicas de dichos corpus, es decir, número de oraciones (N_{orac}), palabras (N_{pal}), vocabulario ($|V|$) y longitudes media para cada idioma (L). M y k indican millones y miles de unidades respectivamente.

[1] www.tc-star.org
[2] www.slt.atr.jp/IWSLT2004

Conj	Leng	N_{orac}	N_{pal}	$\|V\|$	L
entr	en	1,2M	33,4M	105k	27.3
	es		34,7M	168k	28.4
test	en	1,008	26k	3,2k	25.8
	es		25k	3,9k	25.4

Cuadro 2: *Corpus TC-Star. Para el conjunto de test de ambos idiomas se utilizaron dos referencias.*

Conj	Leng	N_{orac}	N_{pal}	$\|V\|$	L
entr	en	20k	188k	8,1k	9.4
	zh		183k	7,6k	9.1
test	en	500	4,1k	2,4k	8.4
	zh		3,7k	893	7.6

Cuadro 3: *Corpus BTEC. Para el conjunto de test en inglés se utilizaron 16 referencias.*

4.2. Entrenamiento de los modelos

En el alineamiento palabra a palabra del corpus de entrenamiento utilizamos *giza++*, las dos direcciones fueron alineadas para posteriormente utilizar la unión de ambos alineamientos, ver (Och y Ney, 2004). Los pares de oraciones del conjunto de entrenamiento fueron segmentadas, extrayendo las unidades bilingües (tuplas) usando el método *extract-tuples* descrito en (Crego, Mariño, y de Gispert, 2004).

Inicialmente, el conjunto de oraciones de entrenamiento del corpus TC-Star fue podado bajo los siguientes criterios:

- Las oraciones estan limitadas a un número máximo de palabras (fijado a 100).
- La diferencia en número de palabras entre cada oración fuente y su correspondiente traducción no puede superar un umbral determinado (fijado a 2.5).

Posteriormente, se realizó una nueva poda sobre el vocabulario de tuplas derivado del corpus TC-Star, considerando el criterio de limitar el número de traducciones para cada secuencia de palabras del idioma fuente (fijado a 20 en el caso de la traducción inglés a español, y a 30 en el caso de la traducción español a inglés).

Las diferentes podas realizadas, benefician la estimación de los modelos de traducción así como a la eficiencia de la búsqueda. Los criterios seguidos para establecer los valores de la poda han sido elegidos empiricamente.

Pueden utilizarse otras técnicas de poda sobre el conjunto de unidades bilingües de entrenamiento, basadas en limitar dichas unidades a aquellas:

- Que ocurren un mínimo número de veces en el conjunto de entrenamiento.
- Que no exceden un umbral de tamaño (número de palabras en cada idioma de la unidad).
- Que no exceden un umbral de fertilidad (diferencia en el número de palabras de cada idioma).

Estas últimas técnicas de poda no fueron utilizadas en los experimentos presentados en esta comunicación.

Para entrenar los modelos N-grama, usamos la herramienta *SRILM* (Stolcke, 2002). El tipo de algoritmo de descuento utilizado fue el Kneser-Ney modificado (Chen y Goodman, 1996), combinando las estimaciones a traves de interpolación.

Los pesos λ_i de la combinación log-lineal de modelos fueron ajustados minimizando el estimador BLEU, usando el algoritmo simplex (Nelder y Mead, 1965).

4.3. Resultados

La tabla 4 muestra los diferentes resultados obtenidos por el decodificador En la dirección chino a inglés, usando el corpus BTEC, y ambas direcciones de traducción para el corpus TC-Star.

Los resultados obtenidos con el decodificador se encuentran entre los del estado del arte para las tareas evaluadas (Akiba et al., 2004) y (TC-Star, 2005).

Tarea	BLEU	WER
zh2en mon	0.331	51.5
zh2en reord	0.363	49.68
es2en mon	0.545	34.4
en2es mon	0.472	41.4

Cuadro 4: *Las primeras dos filas muestran los resultados en la tarea chino a inglés (en la primera fila, el decodificador realiza traducciones monótonas, en la segunda fila se permiten reordenamientos). Las últimas dos filas muestran los resultados del decodificador realizando traducciones monótonas en ambas direcciones.*

Sólo se recomienda la utilización de reordenamientos cuando la pareja de idiomas que

intervienen en la traducción así lo requieren.

Por ejemplo, los reordenamientos requeridos por la pareja español-inglés estan limitados a reordenamientos de corta distancia, dichos reordenamientos estan suficientemente bien capturados por el uso de unidades bilingües formadas por secuencias de palabras.

5. Conclusiones y trabajo futuro

Se ha descrito MARIE un decodificador para sistemas de traducción automática estocástica basado en N-gramas con habilidad para realizar reordenamientos. Permite facilmente la incorporación de nuevos modelos gracias a seguir un enfoque log-lineal.

A pesar de la preferencia por las traducciones monótonas que muestran los modelos basados en N-gramas, los resultados indican la posibilidad de realizar reordenamientos mejorando los resultados de traducción.

La estructura del grafo de búsqueda facilita la utilización de altos niveles de poda gracias a que las hipótesis que compiten en el proceso de poda son comparables.

Las restricciones utilizadas en el espacio de búsqueda permiten reducir su explosión combinatoria, haciendo útil el proceso de búsqueda con reordenamiento.

Se prevé continuar la investigación para mejorar la eficiencia del decodificador mediante mejorar la poda del espacio de búsqueda añadiendo nuevas restricciones de distorsión.

6. Agradecimientos

Esta comunicación ha sido parcialmente subvencionada por el gobierno español, TIC-2002-04447-C02 (proyecto Aliado), la Unión Europea, FP6-506738 (proyecto TC-STAR) y la Universidad Politècnica de Catalunya (beca UPC-RECERCA).

Bibliografía

Akiba, Y., M. Federico, N. Kando, H.Ñakaiwa, M. Paul, y J. Tsujii. 2004. Overview of the IWSLT04 Evaluation Campaign. *IWSLT 04*, october.

Berger, A., P. Brown, S. Della Pietra, V. Della Pietra, y J. Gillet. 1994. The candide system for machine translation. *Proceedings of the Arpa Workshop on Human Language Technology*, March.

Berger, A., S. Della Pietra, y V. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–72, March.

Brown, P., J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, J.D. Lafferty, R. Mercer, y P.S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Chen, S. y J. Goodman. 1996. An Empirical Study of Smoothing techniques for Language Modeling. En *Proceedings of 34th ACL*, páginas 310–318, San Francisco, July.

Crego, J.M., J. Mariño, y A. de Gispert. 2004. Finite-state-based and phrase-based statistical machine translation. *Proc. of the 8th Int. Conf. on Spoken Language Processing, ICSLP'04*, páginas 37–40, October.

de Gispert, A. y J. Mariño. 2002. Using X-grams for speech-to-speech translation. *Proc. of the 7th Int. Conf. on Spoken Language Processing, ICSLP'02*, September.

Germann, U. 2003. Greedy decoding for statistical machine translation in almost linear time. *HLT-NAACL-2003*, May.

Germann, U., M. Jahr, K. Knight, D. Marcu, y K. Yamada. 2001. Fast decoding and optimal decoding for machine translation. *39th Annual Meeting of the Association for Computational Linguistics*, páginas 228–235, July.

Knight, K. 1999. Decoding complexity in word replacement translation models. *Computational Linguistics*, 26(2):607–615.

Koehn, P. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. *Proc. of the 6th Conf. of the Association for Machine Translation in the Americas*, páginas 115–124, October.

Nelder, J.A. y R. Mead. 1965. A simplex method for function minimization. *The Computer Journal*, 7:308–313.

Och, F.J. y H.Ñey. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449, December.

Och, F.J., N. Ueffing, y H.Ñey. 2001. An efficient A* search algorithm for statistical machine translation. *Data-Driven Machine Translation Workshop, 39th Annual Meeeting of the Association for Computational Linguistics (ACL)*, páginas 55–62, July.

Papineni, K., S. Roukos, T. Ward, y W. Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. Informe Técnico RC22176 (W0109-022), IBM Research Division, Thomas J. Watson Research Center.

Picó, David, Jesús Tomás, y Francisco Casacuberta. 2004. Giati: A general methodology for finite-state translation using alignments. En *SSPR/SPR*, páginas 216–223.

Stolcke, A. 2002. Srilm - an extensible language modeling toolkit. *Proc. of the 7th Int. Conf. on Spoken Language Processing, ICSLP'02*, September.

TC-Star. 2005. Deliverable D5:SLT progress report. Informe técnico, http://www.tc-star.org/documents/deliverable/Deliv_D5_Total_21May05.pdf.

Tillmann, C. y H.Ñey. 2000. Word reordering and dp-based search in statistical machine translation. *Proc. of the 18th Int. Conf. on Computational Linguistics, COLING'00*, páginas 850–856, July.

Wang, Y. y A. Waibel. 1998. Fast decoding for statistical machine translation. En *ICSLP98*, December.

Wu, D. 1995. Stochastic inversion transduction grammars, with application to segmentation, bracketing, and alignment of parallel corpora. *Proc. of the 14th International Joint Conf. on Artificial Inteligence (IJCAI)*, páginas 1328–1334, August.

Zens, R., F.J. Och, y H.Ñey. 2004. Improvements in phrase-based statistical machine translation. *Proc. of the Human Language Technology Conference, HLT-NAACL'2004*, páginas 257–264, May.