Traducción automática basada en n-gramas conexionistas

Machine Translation based on Neural Network Language Models

Francisco Zamora-Martínez
Univ. CEU-Cardenal Herrera
46115 Alfara del Patriarca, Valencia
fzamora@dsic.upv.es

María José Castro-Bleda
Univ. Politécnica de Valencia
46022 Valencia
mcastro@dsic.upv.es

Resumen: Este trabajo describe un sistema de traducción que integra n-gramas conexionistas en la etapa de decodificación, motivado por los buenos resultados obtenidos en los últimos años usando estos modelos de lenguaje. Hasta el momento todos los resultados publicados delegan el modelo de lenguaje conexionista a una segunda etapa desacoplada en la que se repuntúan listas de N-best o bien se utilizan sobre grafos de palabras que contienen las N-best. Nuestro objetivo es mostrar la viabilidad de utilizar estos modelos de lenguaje dentro de un sistema totalmente acoplado.
Palabras clave: Traducción automática, Modelado de lenguaje conexionista, Traducción basada en n-gramas

Abstract: This paper describes a Machine Translation system that integrates a Neural Network Language Model in the decoding process. This work is motivated by the excellent performance of these connectionist language models. So far, the use of Neural Network Language Models in the translation systems is uncoupled: they are used in a second stage to rerank a N-best hypothesis list or to parse a word graph containing the N-best list. Our goal is to show the feasibility of using these language models within a fully integrated system.
Keywords: Machine Translation, Neural Network Language Models, n-gram-based Machine Translation

1. *Introducción y motivación*

La traducción automática estadística (Brown et al., 1993) ha experimentado un avance muy rápido en los últimos años, en gran parte debido a la generalización al pasar de la traducción basada en palabras a la traducción basada en segmentos (Koehn, Och, y Marcu, 2003), que es el actual estado del arte. Desde el punto de vista estadístico la ecuación fundamental de la traducción es (Brown et al., 1993):

$$\begin{aligned} \hat{t} &= \operatorname*{argmax}_t p(t|s) && (1)\\ &= \operatorname*{argmax}_t p(s|t)p(t) && (2)\\ &= \operatorname*{argmax}_t p(s,t) && (3) \end{aligned}$$

siendo s la frase de entrada en la lengua origen, y $\hat{t}$ la traducción obtenida como salida del sistema en la lengua destino. Se ha aplicado la regla de Bayes para descomponer la probabilidad $p(t|s)$, resultando en la probabilidad del modelo de traducción $p(s|t)$ y la probabilidad del modelo de lenguaje de la lengua destino $p(t)$, o bien la probabilidad conjunta $p(s,t)$. Esta aproximación se ha extendido en los últimos años a través del principio de máxima entropía, permitiendo la combinación log-lineal de modelos diferentes (Och y Ney, 2002):

$$\hat{t} = \operatorname*{argmax}_t \sum_{m=1}^{M} \lambda_m h_m(s,t) \qquad (4)$$

donde λ_m es el coeficiente de mezcla del modelo m y $h_m(s,t)$ es su puntuación (en escala logarítmica), y M es el número de modelos disponibles.

Por otro lado, moviéndose dentro del marco formal de la teoría de lenguajes, existen diversos trabajos donde la traducción es llevada a cabo por modelos de estados finitos (Vidal, 1997; Knight y Al-Onaizan, 1998; Bangalore y Riccardi, 2000; Casacuberta y Vidal, 2004). La metodología más relevante bajo esta aproximación es GIATI (Casacuberta y Vidal, 2004), que ha sido ampliada en diversos trabajos hasta el punto de incorporar su

desarrollo dentro del marco de los modelos de máxima entropía de la Ecuación (4) (González y Casacuberta, 2008). Existen múltiples trabajos describiendo algoritmos basados en esta metodología para estimar modelos de estados finitos, entre los cuales destacaremos lo que se conoce como traducción basada en n-gramas (Mariño et al., 2006), ya que será la idea mediante la cual introduciremos los modelos de lenguaje conexionistas en el ámbito de la traducción automática.

El uso de redes neuronales artificiales en traducción automática ha recibido poco interés de la comunidad científica, entre otras razones por su alto coste computacional y la difícil escalabilidad de los modelos. Este trabajo se presenta una solución a este problema, y da una alternativa a otras técnicas conexionistas, basadas en redes recurrentes (Castaño y Casacuberta, 1997). A diferencia de las redes recurrentes, la aproximación presentada tiene mejor escalabilidad, y se integra dentro del marco de la traducción estadística basada en n-gramas a través de la Ecuación (4).

2. *Modelado de lenguaje conexionista*

El modelado de lenguaje es una parte esencial dentro de un sistema de traducción automática, y en nuestro caso, dado que la aproximación a la traducción se basa en n-gramas, el modelo de lenguaje (bilingüe) hace la función de modelo de traducción. El modelado de lenguaje estadístico se basa en la predicción de cada unidad lingüística dada la secuencia de todas las palabras anteriores (Bahl, Jelinek, y Mercer, 1983; Jelinek, 1997):

$$p(w) = \prod_{i=1}^{|w|} p(w_i | w_{i-1} w_{i-2} \dots w_1) \qquad (5)$$

donde $w = w_1 w_2 \dots w_{|w|}$ es una secuencia unidades lingüísticas de un determinado vocabulario Ω. Este cálculo se simplifica introduciendo un límite de longitud $n-1$ al contexto utilizado, estableciendo el denominado modelo de n-gramas (Bahl, Jelinek, y Mercer, 1983):

$$p(w) \approx \prod_{i=1}^{|w|} p(w_i | w_{i-1} w_{i-2} \dots w_{i-n+1}) \qquad (6)$$

Este modelo se estima a partir de conteos en un corpus textual, y dada la naturaleza exponencial de la tarea, es necesario aplicar algún tipo de suavizado al modelo resultante de manera que permita calcular la probabilidad para *cualquier* combinación posible de w. Éste es uno de los grandes problemas de esta aproximación, de manera que el resultado final del sistema dependerá, en gran medida, de la calidad del suavizado aplicado.

Varios autores (Bengio et al., 2003; Castro-Bleda y Prat, 2003; Schwenk, 2007; Bengio, 2008) proponen el uso de redes neuronales para estimar los modelos de lenguaje, definiendo el modelo conexionista de n-gramas como una red neuronal que recibe en su entrada el contexto de las $n-1$ palabras anteriores y calcula en su salida la probabilidad $p(\omega | w_{i-1} w_{i-2} \dots w_{i-n+1})$ para toda palabra del vocabulario Ω. El uso de una red neuronal presenta como ventaja la interpolación de forma natural de secuencias de palabras no vistas en el entrenamiento.

Dado que la red neuronal recibe la secuencia de palabras, necesitamos codificarlas de alguna manera para poder darle dicha información. Para ello utilizamos una representación de las palabras en un espacio continuo (Bengio et al., 2003; Schwenk, 2007), que nos sirve de codificación distribuida. La red neuronal aprende durante el entrenamiento a realizar el cómputo de la probabilidad condicional para cada una de las palabras del vocabulario, conocidas las $n-1$ anteriores, y a proyectar cada palabra del vocabulario en dicho espacio continuo.

La topología de la red neuronal, que se puede ver en la Figura 1 (izquierda), constará de una capa de proyección, una capa oculta y una capa de salida. La capa de proyección está formada por I entradas y P neuronas, de tal forma que las neuronas se particionan en $n-1$ conjuntos, cada uno de ellos se corresponde con una palabra de entrada. Cada palabra de entrada está codificada localmente, esto es, una unidad de entrada a uno y las demás a cero. Los pesos entre los conjuntos están ligados, con lo que en realidad no hay $n-1$ conjuntos de pesos, sino, un único conjunto de pesos que es compartido por los $n-1$ conjuntos. Dicho conjunto de pesos actúa como una matriz que proyecta la entrada en una codificación distribuida en un espacio continuo, reduciendo la dimensionalidad de la entrada. La función de activación

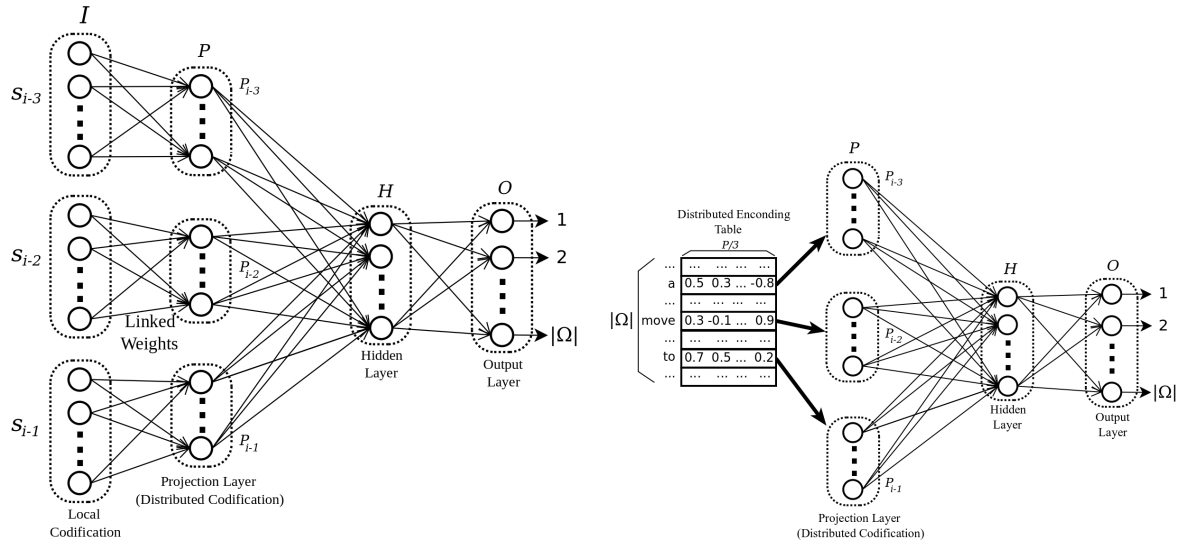


Figura 1: Arquitectura de un modelo de lenguaje de 4-gramas conexionista (izquierda), y el mismo modelo durante la fase de test tras eliminar la entrada de la red (derecha).

de esta capa de proyección puede ser lineal o tangente hiperbólica. Esta capa se descarta después del entrenamiento, convirtiéndose en una tabla del tamaño del vocabulario ($|\Omega|$), como muestra la Figura 1 (derecha). La capa oculta contiene H neuronas con activación tangente hiperbólica. La capa de salida está formada por $O = |\Omega|$ neuronas con función de activación softmax. La red neuronal se entrena usando el algoritmo de retropropagación del error incorporando término de regularización ("weight decay"). Como función objetivo se utiliza la entropía cruzada.

Los modelos de lenguaje conexionistas se han utilizado con exito en diversas tareas (Schwenk y Gauvain, 2002; Schwenk, Costa-jussà, y Fonollosa, 2007; Khalilov et al., 2008), entre ellas traducción automática estadística. En todos los casos la aplicación del modelo conexionista ha sido en una etapa desacoplada del sistema, trabajando con listas de N-best, o bien aplicándolos sobre grafos de palabras que contienen las N-best. El objetivo de este trabajo se basa en (Zamora-Martínez, Castro-Bleda, y España-Boquera, 2009; Zamora-Martínez et al., 2010), donde ya se ha presentado un sistema de reconocimiento que incorpora modelos conexionistas de forma acoplada.

3. Aceleración de la evaluación de los modelos conexionistas

El cuello de botella de la red neuronal cuando se trabaja con grandes vocabularios se encuentra en su capa de salida, ya que ésta tiene tantas unidades como palabras en el vocabulario. Además, la función de activación de esta capa es la softmax, que sigue esta fórmula:

$$o_i = \frac{exp(a_i)}{\sum_{j=1}^{|\Omega|} exp(a_j)}, \qquad (7)$$

siendo a_i el resultado del producto escalar de la i-ésima neurona de salida y o_i su activación tras aplicar la normalización softmax. Esta función precisa del cómputo de la activación de *todas* las neuronas de salida para poder calcular la constante de normalización de la softmax que normaliza los valores $exp(a_i)$ entre 0,0 y 1,0. Consecuentemente, la red neuronal calcula todas las salidas aunque solamente se vayan a necesitar unas pocas. En la literatura (Schwenk, 2007) se pueden encontrar soluciones al problema de esta función de activación. Todas ellas se basan en la idea de usar el modelo de lenguaje conexionista en una etapa posterior, a diferencia de la aproximación que estamos presentando que nos permitirá aprovechar las ventajas de los modelos conexionistas durante la etapa de búsqueda.

Para lograr este objetivo vamos a extender un trabajo anterior (Zamora-Martínez, Castro-Bleda, y España-Boquera, 2009) donde se presentó la idea de precalcular las constantes de normalización softmax que más probablemente se necesitarán durante la ejecución del sistema. De esa manera se consigue reducir significativamente el coste computacional dado que, buscar estas constantes en una tabla, comparado con el coste de calcular-

las, es *casi* despreciable. Cuando una constante no se encuentra en la tabla se pueden hacer dos cosas: o bien se calcula "al vuelo", o bien se utiliza algún tipo de suavizado. La primera propuesta conlleva un incremento del coste computacional, la segunda propuesta permite mantener el coste computacional en cotas reducidas, incluso con redes de gran tamaño, a costa de una pequeña pérdida en la calidad del modelo. Seguiremos la segunda idea, cuando una constante no se encuentre, se utilizará un modelo de lenguaje más simple (bajando la n del n-grama, o usando un modelo de lenguaje estadístico estándar).

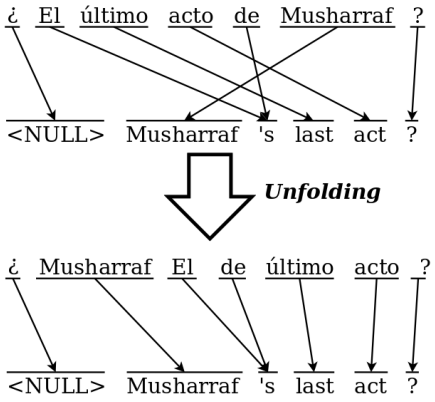
4. Segmentación del corpus en tuplas bilingües

Basándonos en las ideas de (Mariño et al., 2006), la unidad básica de nuestro modelo de traducción es la tupla bilingüe. Éstas son extraídas de un corpus bilingüe alineado palabra a palabra entre ambas lenguas. Dicho alineamiento se calcula mediante la herramienta GIZA++ (Och y Ney, 2003) que implementa diversos modelos IBM de alineamiento y traducción (Brown et al., 1993).

Las tuplas se extraen asumiendo alineamiento muchos-a-muchos, en contra de otras aproximaciones similares que asumen uno-a-uno (Bangalore y Riccardi, 2000), o bien uno-a-muchos (Casacuberta y Vidal, 2004). Además el proceso se enriquece mediante el reordenamiento de la frase de entrada (Banchs et al., 2005; Sanchis y Casacuberta, 2006; Costajussà y Fonollosa, 2006), de tal forma que la extracción de tuplas se realiza estableciendo como objetivo el orden de las palabras en la frase destino, haciendo que la frase de entrada tenga el orden adecuado para la dirección de la traducción que se está trabajando. Para extraer las tuplas seguimos las restricciones de (Banchs et al., 2005). La Figura 2 muestra un ejemplo de este proceso a partir de dos frases alineadas. Tendremos en cuenta las consideraciones explicadas en (Banchs et al., 2005; Mariño et al., 2006) para atacar el problema de las palabras sin traducción y de la búsqueda con reordenamiento.

5. Traductor basado en n-gramas

La traducción basada en n-gramas (Mariño et al., 2006) es un caso específico de la traducción con modelos de estados finitos (Vidal, 1997), en la cual el transductor estimado para realizar el proceso de traducción se construye a partir de un modelo de lenguaje de n-gramas. Denotaremos la frase de entrada como s, la función $r(s_i)$ nos dirá la posición donde se coloca la palabra de entrada s_i, después de reordenarla. La frase de salida será t, el conjunto de tuplas dadas como respuesta del sistema será T, $s(T_i)$ será la parte correspondiente a la lengua origen de la tupla T_i y $t(T_i)$ la parte correspondiente a la lengua destino de la tupla T_i.



Tuplas extraídas		
¿	→	<NULL>
Musharraf	→	Musharraf
El de	→	's
último	→	last
acto	→	act
?	→	?

Figura 2: Ejemplo de alineamiento entre dos frases y el conjunto de tuplas extraídas. <NULL> representa la palabra vacía.

5.1. Combinación de modelos

El sistema presentado implementa la búsqueda de la mejor traducción dentro de una combinación log-lineal de diversos modelos: modelo de reordenamiento, modelos léxicos, modelo de lenguaje de la lengua destino, modelo de lenguaje de tuplas bilingües (modelo de traducción), penalización del número de tuplas, y penalización del número de palabras en la respuesta del sistema. Los modelos de la combinación log-lineal reciben como entrada el conjunto de las tuplas. Extenderemos la Ecuación (4) de esta manera:

$$\hat{T} = \underset{T}{\operatorname{argmax}} \sum_{m=1}^{M} \lambda_m h_m(T) \qquad (8)$$

Modelo de reordenamiento Este modelo permite puntuar el reordenamiento aplicado a las palabras de la frase de entrada, penalizan-

do el cambio de orden de las palabras de la frase de entrada:

$$h_r(T) = h'_r(s(T)) = \sum_{i=1}^{|s|} abs(r(s_i) - i)$$

Modelo IBM-1 léxico directo e inverso Permiten establecer una penalización a priori de cada una de las tuplas (Mariño et al., 2006). Para cada tupla del sistema de traducción se precalcula su puntuación y se guarda en una tabla. Está basado en las probabilidades de los modelos IBM-1 (Brown et al., 1993):

$$\begin{aligned} h_{s2t}(T) &= \sum_{i=1}^{|T|} h'_{s2t}(s(T_i), t(T_i)) \\ h'_{s2t}(x,y) &= \log \frac{1}{(|x|+1)^{|y|}} \prod_{j=1}^{|y|} \sum_{i=0}^{|x|} q(y_j|x_i) \\ h_{t2s}(T) &= \sum_{i=1}^{|T|} h'_{t2s}(s(T_i), t(T_i)) \\ h'_{t2s}(x,y) &= \log \frac{1}{(|y|+1)^{|x|}} \prod_{i=1}^{|x|} \sum_{j=0}^{|y|} q(x_i|y_j) \end{aligned}$$

donde $q(t_j|s_i)$ es la probabilidad de traducción de s_i en t_j, y $q(s_i|t_j)$ la contraria.

Penalización del número de tuplas y palabras Sirve para penalizar la longitud de la respuesta del sistema (*wip* de "Word Insertion Penalty" y *tip* de "Tuple Insertion Penalty"):

$$\begin{aligned} h_{wip}(T) &= \sum_{i=1}^{|T|} |t(T_i)| = |t| \\ h_{tip}(T) &= |T| \end{aligned}$$

Modelo de lenguaje bilingüe (modelo de traducción) Establece la probabilidad de cada una de las tuplas del sistema condicionada a la secuencia de las $n-1$ tuplas anteriores y se utiliza para aproximar la probabilidad conjunta $p(s,t)$. De este modelo tendremos más de uno, típicamente al menos el modelo estadístico y el modelo conexionista, ambos combinados log-linealmente siguiendo la Ecuación (8):

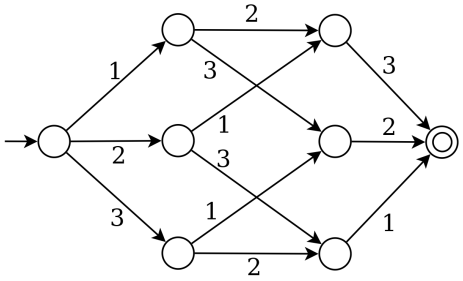


Figura 3: Ejemplo de grafo generado para realizar búsqueda sobre todos los posibles reordenamientos de una frase de $|s| = 3$ palabras. En el grafo se indica la posición en la frase de entrada.

$$h_{TM}(T) = \log \prod_{i=1}^{|T|} p(T_i|T_{i-1} \dots T_{i-n+1})$$

siendo n la longitud del n-grama.

Modelo de lenguaje de la lengua destino Sirve para puntuar la calidad de la frase resultante en la lengua destino. Calcula la probabilidad de la frase dada como respuesta por el sistema, $p(t)$. De nuevo, de este modelo podemos tener más de uno. En nuestra aproximación tendremos el modelo de n-gramas estadístico estándar y el modelo conexionista. Ambos son combinados siguiendo la Ecuación (8):

$$\begin{aligned} h_{LM}(T) &= \log p(t(T_1)t(T_2) \dots t(T_{|T|})) \\ &= \log \prod_{i=1}^{|t|} p(t_i|t_{i-1} \dots t_{i-n+1}) \end{aligned}$$

siendo n la longitud del n-grama.

5.2. Etapas del proceso de traducción

La idea del sistema completo se basa en una solución por programación dinámica al problema del viajante de comercio aplicada al problema del reordenamiento (Held y Karp, 1971; Tillmann y Ney, 2003), junto con el algoritmo de Viterbi para buscar el camino de mejor probabilidad, aplicando poda global tanto estática ("histogram pruning") como dinámica ("beam search") en todas las fases.

Reordenamiento de la entrada A partir de la frase de entrada se va generando poco a poco un grafo con los mejores reordena-

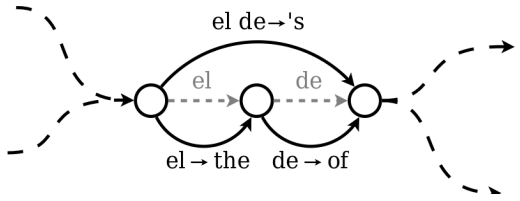


Figura 4: Ejemplo de transformación de un segmento del grafo de palabras de entrada (lengua origen), en un segmento de grafo de tuplas bilingües (vocabulario bilingüe).

mientos, siguiendo una solución por programación dinámica al problema del viajante de comercio (Held y Karp, 1971). Dicho grafo es serializado siguiendo las ideas de trabajos previos (Boquera, Moya, y Martínez, 2007), de manera que el grafo se emite siguiendo un orden topológico, lo que permite incorporar en el algoritmo de generación del mismo la puntuación obtenida por el sistema completo de traducción para tener un criterio de poda global. En este punto las aristas del grafo incorporan la puntuación del modelo de reordenamiento. La Figura 3 ilustra un ejemplo de grafo generado en esta etapa.

Generación del grafo de tuplas A partir del grafo de palabras reordenadas de la frase de entrada se genera un grafo de tuplas bilingües. Dicho grafo incorporará la puntuación de los modelos léxicos IBM-1 directo e inverso, que se combina con la puntuación de la etapa anterior. La Figura 4 ilustra un ejemplo de cómo se genera un trozo del grafo de tuplas bilingües a partir del grafo de palabras de la lengua origen.

Búsqueda de la mejor traducción Cuando llegamos a esta última etapa, la decodificación se reduce a la búsqueda del camino de máxima probabilidad en un grafo dirigido y sin ciclos, que se resuelve por programación dinámica mediante el algoritmo de Viterbi. En este punto se añade el modelo de lenguaje bilingüe así como el modelo de lenguaje de la lengua destino, que se combinan entre sí y con la puntuación dada por las etapas anteriores.

6. Experimentación

El sistema está todavía en desarrollo y es muy preliminar, pero podemos presentar resultados que corroboran la posibilidad de utilizar los modelos conexionistas durante la etapa de decodificación. El experimento se ha realizado con el corpus descrito en el Cuadro 1, escogiendo la dirección de traducción español-inglés, correspondiente al corpus NewsCommentary10 distribuido para la campaña de evaluación del "Workshop of Machine Translation" del año 2010 (WMT'10). El alineamiento palabra a palabra, en ambas direcciones de la traducción, se ha calculado mediante la herramienta Gizza++, y se ha obtenido un único alineamiento a partir de los dos anteriores mediante la opción grow-diag-final-and de Moses (Koehn et al., 2007).. El corpus ha sido segmentado en tuplas, y se ha entrenado un modelo de 4-gramas estándar con la herramienta SRI LM (Stolcke, 2002), que servirá de baseline. También se ha entrenado un modelo de 4-gramas para el inglés, con el corpus NewsCommentary monolingüe suministrado de nuevo durante la campaña de evaluación del WMT'10, con 125 879 frases y 2 973 711 palabras. Para ambos modelos se ha utilizado la técnica de descuento "modified Knesser-Ney", configurando la herramienta para entrenar el modelo interpolando la probabilidad de todos los niveles de n-gramas.

El *modelo de traducción conexionista* está compuesto por redes entrenadas para 4-gramas, 3-gramas y 2-gramas, siguiendo la técnica de suavizado descrita en (Zamora-Martínez, Castro-Bleda, y España-Boquera, 2009), entrenadas con el mismo corpus de tuplas que el baseline, con 100 unidades por palabra en la capa de proyección y 150 unidades en la capa oculta. El vocabulario de tuplas se ha restringido, entrenando la red con las 20K tuplas más frecuentes.

El *modelo de lenguaje conexionista para la lengua destino (inglés)* está compuesto por redes neuronales entrenadas para 4-gramas, 3-gramas y 2-gramas, con 80 unidades como proyección de cada palabra y 100 en la capa oculta para bigramas y trigramas, y 120 para el 4-grama. Las redes neuronales se han entrenado utilizando como vocabulario las 15K palabras más frecuentes en el corpus del Cuadro 1, entrenándose con el mismo corpus que el modelo de lenguaje del baseline.

Se ha utilizado el proceso iterativo conocido como MERT, descrito en (Och, 2003), para estimar los coeficientes de la combinación log-lineal. Los resultados obtenidos se pueden ver en el Cuadro 2. El baseline utiliza únicamente modelos de n-gramas estándar (7 parámetros en la combinación log-lineal). El sistema NN LM combina los modelos del baseline jun-

	NewsCommentary10		Test08+Test09		Test10	
	ES	EN	ES	EN	ES	EN
Líneas	80 943	80 943	4 576	4 576	2 489	2 489
Palabras	1 823 915	1 625 144	120 689	115 360	65 500	61 924
Vocabulario	53 543	38 788	15 349	12 731	10 782	8 905
OOV	0	0	4 128	3 318	2 404	2 004

Cuadro 1: Estadísticas del corpus utilizado para entrenar el sistema de traducción. Como conjunto de desarrollo se han escogido los tests de la WMT'10 de los años 2008 y 2009 concatenados, y como conjunto de test el oficial del año 2010.

Sistema	Test08+Test09	Test10
Baseline	19.6/62.5	21.7/58.7
NN LM	20.4/61.9	22.5/58.1
Moses	20.3/62.0	22.5/58.1

Cuadro 2: Resultados sobre el conjunto de validación y de test. Se muestran en BLEU/TER.

to a los modelos conexionistas descritos anteriormente (9 parámetros en la combinación log-lineal). Como referencia se ha entrenado también un sistema de traducción basada en segmentos con la herramienta Moses (Koehn et al., 2007), utilizando el mismo modelo de lenguaje del inglés que el baseline (14 parámetros en la combinación log-lineal).

En el Cuadro 2 se puede observar que el sistema de traducción conexionista mejora el baseline en 0,8 puntos de BLEU y 0,6 puntos de TER. El sistema NN LM logra alcanzar los resultados de Moses, que es el estado del arte actual, aun partiendo de un baseline todavía en desarrollo y que está en proceso de mejora.

7. Conclusiones

En este trabajo se presenta un sistema completo de traducción donde el modelo de lenguaje conexionista se utiliza de forma acoplada en la búsqueda del decodificador. Bajo el marco de la traducción basada en n-gramas (Mariño et al., 2006), y extendiendo las ideas presentadas en (Zamora-Martínez, Castro-Bleda, y España-Boquera, 2009) para acelerar el uso de modelos de lenguaje conexionistas, se ha logrado integrar las redes neuronales en el decodificador. Todos los trabajos realizados hasta el día de hoy que han utilizado modelos de lenguaje conexionistas, han delegado su aplicación a una segunda etapa, donde se utilizan para repuntuar listas de las N-best hipótesis dadas como respuesta por un sistema de traducción automática base. La experimentación presentada reafirma la validez de la aproximación propuesta. Como trabajo futuro se plantea realizar un análisis más profundo de hasta dónde pueden llegar los modelos conexionistas cuando se integran en el proceso de decodificación, en lugar de utilizarse en una segunda etapa desacoplada.

Bibliografía

Bahl, L. R., F. Jelinek, y R. L. Mercer. 1983. A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE TPAMI*, 5(2):179–190.

Banchs, R, JM Crego, A Gispert, P Lambert, J.A.R. Fonollosa, M.R. Costa-jussà, y J Mariño. 2005. Bilingual N-gram Statistical Machine Translation. En *10th Machine Translation Summit , MT-Summit*, páginas 275–282.

Bangalore, Srinivas y Giuseppe Riccardi. 2000. Stochastic finite-state models for spoken language machine translation. En *NAACL-ANLP 2000 Workshop on Embedded machine translation systems - Volume 5*, páginas 52–59.

Bengio, Y. 2008. Neural net language models. *Scholarpedia*, 3(1):3881.

Bengio, Y., R. Ducharme, P. Vincent, y C. Jauvin. 2003. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3(2):1137–1155.

Boquera, Salvador España, Jorge Gorbe Moya, y Francisco Zamora Martínez. 2007. Semiring Lattice Parsing Applied to CYK. En *Pattern Recognition and Image Analysis*, volumen 4477 de *LNCS*, páginas 603–610.

Brown, Peter F., Vincent J. Della Pietra, Stephen A. Della Pietra, y Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 19(2):263–311.

Casacuberta, Francisco y Enrique Vidal. 2004. Machine translation with inferred stochastic finite-state transducers. *Comput. Linguist.*, 30:205–225.

Castaño, M.A. y F. Casacuberta. 1997. A Connectionist Approach to Machine Translation. En *Eurospeech*, páginas 91–94.

Castro-Bleda, M.J. y F. Prat. 2003. New Directions in Connectionist Language Modeling. En *Computational Methods in Neural Modeling*, volumen 2686 de *LNCS*, páginas 598–605.

Costa-jussà, Marta R. y José A. R. Fonollosa. 2006. Statistical machine reordering. En *EMNLP*, páginas 70–76, Morristown, NJ, USA. Association for Computational Linguistics.

González, Jorge y Francisco Casacuberta. 2008. A finite-state framework for log-linear models in Machine Translation. En *EAMT*, páginas 22–23.

Held, M y RM Karp. 1971. The traveling-salesman problem and minimum spanning trees: Part II. *Mathematical Programming*, 1:6–25.

Jelinek, F. 1997. *Statistical Methods for Speech Recognition*. Language, Speech, and Communication.

Khalilov, Maxim, José A. R. Fonollos, F. Zamora-Martínez, María J. Castro-Bleda, y S. España-Boquera. 2008. Neural Network Language Models for Translation with Limited Data. En *20th ICTAI*, páginas 445–451.

Knight, Kevin y Yaser Al-Onaizan. 1998. Translation with finite-state devices. En *AMTA*, páginas 421–437.

Koehn, Philipp, Franz Josef Och, y Daniel Marcu. 2003. Statistical phrase-based translation. En *NAACL*, páginas 48–54.

Koehn et al., P. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. En *Proc. of the ACL Demo and Poster Sessions*, páginas 177–180.

Mariño, José B., Rafael E. Banchs, Josep M. Crego, Adrià de Gispert, Patrik Lambert, José A. R. Fonollosa, y Marta R. Costa-jussà. 2006. N-gram-based machine translation. *Comput. Linguist.*, 32:527–549.

Och, F. y H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. En *Proc. of the ACL'02*, páginas 295–302.

Och, F.J. 2003. Minimum Error Rate Training in Statistical Machine Translation. En *Proc. of ACL*, páginas 160–167.

Och, Franz Josef y Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Sanchis, Germán y Francisco Casacuberta. 2006. N-best reordering in statistical machine translation. En *IV JTH*, páginas 99–104.

Schwenk, Holger. 2007. Continuous space language models. *Comput. Speech Lang.*, 21(3):492–518.

Schwenk, Holger, Marta R. Costa-jussà, y José A. R. Fonollosa. 2007. Smooth bilingual n-gram translation. En *EMNLP*, páginas 430–438.

Schwenk, Holger y Jean-Luc Gauvain. 2002. Connectionist language modeling for large vocabulary continuous speech recognition. En *ICASSP*, páginas 765–768.

Stolcke, A. 2002. SRILM: an extensible language modeling toolkit. En *ICSLP*, páginas 901–904.

Tillmann, Christoph y Hermann Ney. 2003. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Comput. Linguist.*, 29(1):97–133.

Vidal, E. 1997. Finite-state speech-to-speech translation. *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, 1:111.

Zamora-Martínez, F, MJ Castro-Bleda, S España Boquera, y J Gorbe-Moya. 2010. Unconstrained Offline Handwriting Recognition using Connectionist Character N-grams. En *IJCNN*.

Zamora-Martínez, F., M.J. Castro-Bleda, y S. España-Boquera. 2009. Fast Evaluation of Connectionist Language Models. En *IWANN*, volumen 5517 de *LNCS*, páginas 33–40.