ARTÍCULO

Generación automática de codificaciones de vocabularios para Traducción Automática mediante Redes Neuronales

Gustavo A. Casañ, Mª Asunción Castaño
Dpto. Ingeniería y Ciencia de los Computadores
Universitat Jaume I
Avda. Sos Baynat s/n
Castellón, 12071
{ncasan,castano}@icc.uji.es

Recientemente los autores han desarrollado un método que, utilizando un sencillo modelo conexionista, genera codificaciones de palabras de un vocabulario a partir de ejemplos de frases del lenguaje implicado. Estas codificaciones han demostrado experimentalmente que son útiles para abordar tareas de Traducción Automática en dominios semánticos restringidos. Sin embargo, el método no es completamente automático, dado que depende de la elección del tamaño a priori de las codificaciones. La incorporación de un sencillo mecanismo de poda de las unidades del modelo conexionista ha transformado este método en completamente automático.

Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. Vol. VIII, No.22 (2004),pp. 175-186
ISSN: 1137-3601. © AEPIA (http://www.aepia.org/revista).

Generación automática de codificaciones de vocabularios para Traducción Automática mediante Redes Neuronales*

Gustavo A. Casañ, Mª Asunción Castaño

Dpto. Ingeniería y Ciencia de los Computadores
Universitat Jaume I
Avda. Sos Baynat s/n
Castellón, 12071
{ncasan,castano}@icc.uji.es

Resumen

Recientemente los autores han desarrollado un método que, utilizando un sencillo modelo conexionista, genera codificaciones de palabras de un vocabulario a partir de ejemplos de frases del lenguaje implicado. Estas codificaciones han demostrado experimentalmente que son útiles para abordar tareas de Traducción Automática en dominios semánticos restringidos. Sin embargo, el método no es completamente automático, dado que depende de la elección del tamaño a priori de las codificaciones. La incorporación de un sencillo mecanismo de poda de las unidades del modelo conexionista ha transformado este método en completamente automático.

Palabras clave: Codificación de Vocabularios, Traducción Automática, Modelos Conexionistas.

1. Introducción

Actualmente, las Redes Neuronales Artificiales, también llamadas Modelos Conexionistas o simplemente Redes Neuronales (RN), constituyen un área de gran interés desde la perspectiva de la ingeniería. El principal atractivo de las RN radica, por un lado, en su capacidad de aprendizaje a partir de ejemplos del mundo real y, por otro lado, en su habilidad para resolver problemas en los que aparecen dependencias temporales, llevándonos esto último a las Redes Neuronales Recurrentes. Estas características convierten a las arquitecturas conexionistas en herramientas interesantes para abordar tareas relacionadas con el procesamiento del lenguaje como son el Reconocimiento de Voz, el tratamiento del Lenguaje Natural, la Comprensión del Lenguaje o la Traducción Automática (TA) entre lenguajes.

En la bibliografía pueden encontrarse algunos mecanismos de traducción llevados a cabo total o parcialmente mediante técnicas conexionistas, como son los presentados en [Koncar,94] y [Waibel,91]. Sin embargo, el sistema conexionista de [Koncar,94] emplea topologías estáticas que no son muy apropiadas para abordar tareas reales de TA. Por otro lado, en [Waibel,91] se plantea un esquema complicado que resuelve la traducción mediante análisis sintácticos y semánticos de las frases a traducir y traducidas y a través de un lenguaje intermedio o interlingua.

En contraposición a estas aproximaciones, en [Castaño,99] se aborda el problema de la TA con

* Financiado parcialmente por la fundación Bancaixa-Castelló, proyecto P1.1B2002-1.

una aproximación conexionista dinámica sencilla, denominada RECONTRA (del inglés "Recurrent Connectionist Translator"), y en la que se obvian procesos de análisis sintácticos y semánticos, permitiendo abordar de manera directa (sin un lenguaje intermedio) la traducción entre el lenguaje fuente y el destino.

En los primeros experimentos llevados a cabo con este modelo, la codificación de los vocabularios implicados en el proceso de traducción se realizó mediante representaciones locales, sencillas y fáciles de interpretar. Sin embargo, las codificaciones locales no son prácticas con vocabularios grandes, al generar RN con un tamaño demasiado grande para poder ser entrenadas en un tiempo razonable. Este problema se abordó de forma parcial en [Castaño,98] y [Casañ,99], donde se demostró cómo la utilización de determinados tipos de codificaciones distribuidas (más compactas que las codificaciones locales) permite abordar problemas con vocabularios más amplios. Ahora bien, en dicha experimentación las codificaciones se generaron de forma manual. Recientemente, en [Casañ,03] se presentó un método, basado también en modelos conexionistas, para obtener automáticamente codificaciones distribuidas con un determinado tamaño preestablecido.

En este artículo se presenta un método para determinar el tamaño de dichas codificaciones, convirtiendo en completamente automático el proceso de generación de codificaciones. Dicho mecanismo se aplicó a tareas de TA ligeramente más complejas que las abordadas hasta el momento.

El artículo se organiza de la siguiente forma: En la sección 2 se explican distintos tipos de codificaciones posibles para representar las palabras de un vocabulario en una RN. En la sección 3 se describen los modelos neuronales utilizados en la experimentación. En la sección 4 se comenta la tarea sobre la que se ha experimentado. En la sección 5 se presentan los resultados de la experimentación y, finalmente, se exponen las conclusiones que se pueden extraer de ellos.

2. Codificaciones de los vocabularios

Las representaciones (codificaciones) de los elementos que procesa una RN (a su entrada y a su salida) tienen una importancia crítica en los modelos conexionistas, puesto que en gran parte el esquema de representación utilizado determina la capacidad de dicho sistema. Representaciones cuidadosamente construidas pueden proporcionar propiedades útiles a estos sistemas, como habilidad para aprender, generalización automática, etc.

2.1. Tipos de codificaciones

A continuación se realiza una rápida exposición de los tipos de codificación, local y distribuida, que puede procesar una RN en la entrada/salida para representar, en nuestro caso, palabras de un vocabulario. Pueden ampliarse más detalles al respecto en [Miikkulainen,91], [Plate,94] y [Castaño,98], entre otros.

2.1.1. Codificación local

En la *codificación local* (o unaria) cada unidad de entrada o de salida representa a una de las palabras del vocabulario del lenguaje fuente o destino, respectivamente. Para ello, una unidad está activa (toma, por ejemplo, valor 1) y las demás inactivas (valor 0). Así, en un vocabulario de 10 palabras, dos posibles palabras de éste, *círculo* y *pirámide*, podrían representarse como sigue:

círculo	1 0 0 0 0 0 0 0 0 0
pirámide	0 1 0 0 0 0 0 0 0 0

Las principales ventajas de las codificaciones locales son su representación explicita de los componentes del vocabulario y la facilidad de creación y comprensión para el experimentador, así como la facilidad de manipularlos. Sus problemas son inherentes al tipo de representación utilizada: se produce una gran ineficiencia cuando se trabaja con vocabularios grandes, puesto que se genera un número de conexiones de la red excesivo.

2.1.2. Codificación distribuida

La *codificación distribuida* supone que cada palabra se representa mediante un patrón de activación distribuido en toda o parte de la entrada y/o salida de la red. Esto significa que cada unidad puede intervenir en la representación de más de una palabra. Cuando todas las unidades forman parte de la representación de cada elemento, hablamos de una *codificación holográfica*. Siguiendo con el ejemplo anterior, una posible codificación holográfica adoptando cuatro unidades de representación, en las que cada una de las cuatro unidades toma un valor real entre 0 y 1, podría ser:

círculo	0,1	0,1	0,1	0,1
pirámide	0,2	0,2	0,2	0,2

Hay dos tipos principales de codificación distribuida: la *codificación distribuida simbólica*, en la que cada unidad activada corresponderá a una microcaracterística (siempre de mayor nivel de

abstracción que el conjunto de conceptos que queremos codificar) que representa una categoría sintáctica o semántica; y la *codificación distribuida subsimbólica*, en la que cada entrada no es sintáctica o semánticamente interpretable. Continuando con el ejemplo anterior, una codificación distribuida subsimbólica podría ser:

círculo	1 1 1 0 0 0 0
pirámide	0 1 1 0 1 0 0

En una codificación simbólica, cada unidad tendría un significado y tomaría un valor según la palabra descrita posea o no esa característica, en cuyo caso se podrían utilizar simplemente valores binarios (1 ó 0) o, según el grado en que la posea, valores reales entre 0 y 1, por ejemplo. Una posible codificación simbólica binaria para el ejemplo anterior, en la que la primera unidad representa la categoría gramatical "nombre" y la segunda unidad el "género", sería:

círculo	1 1 0 0 0 0 0
pirámide	1 0 0 0 0 0 0

Existe todavía otra forma de ver las codificaciones distribuidas: se denomina *representación "tosca"* (o, en inglés, "coarse representation") a una codificación en la que cada palabra se representa por varias unidades consecutivas o agrupadas, que pueden adoptar valores binarios o reales. En el siguiente ejemplo la agrupación de unidades se realiza según la función sintáctica de cada una de estas palabras en la frase:

círculo	1	1	1	0	0	0	0
pirámide	0	1	1	0	0	0	0
está	0	0	0	0	0	1	1
tocan	0	0	0	0	1	0	1

Las principales ventajas de las representaciones distribuidas son:

- Pueden representar aspectos relevantes de los objetos (sus propiedades o características).
- Permiten que objetos similares puedan tener representaciones similares (aspecto interesante para el traductor que vamos a utilizar).
- Hacen un uso (más) efectivo de los recursos de representación (unidades).
- Pueden estar distribuidas en un espacio continuo de vectores.

Sus desventajas se presentan a la hora de decidir qué tipo de representación será más adecuado y cuando hay que representar asociaciones variables (por ejemplo, una palabra que puede tener varias acepciones). Por otro lado, los traductores que utilizan este tipo de codificación suelen necesitar mayor tiempo de entrenamiento.

3. Modelos conexionistas

3.1. Un perceptrón multicapa como generador de codificaciones distribuidas

Un Perceptrón Multicapa (PM) es un tipo de RN no recurrente que se considera uno de los tipos básicos de modelo conexionista. Su principal característica es su capacidad como clasificador automático [Rojas,96]. En la figura 1 se puede observar la estructura de un PM con tres capas de unidades, la de entrada, la de salida y la capa oculta (o, lo que es lo mismo, con dos capas de conexiones o pesos).

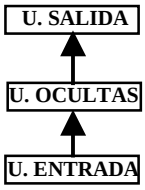

Figura 1. Perceptrón multicapa.

Un PM entrenado como clasificador, dado un patrón a la entrada proporcionará como salida la clase a la que cree que pertenece. En la bibliografía se han utilizado diversos tipos de PM como codificadores y decodificadores de representaciones para palabras y frases. Así, en [Pollack,90] se emplearon con las máquinas RAAM (del inglés "Recurrent Auto-Associative Memories") y en [Miikkulainen,91] con el modelo FGREP (del inglés "Forming Global Respresentations with Extended back-Propagation"). Con el modelo RAAM se generaban codificaciones de las frases a partir de las activaciones que se desarrollaban en la capa oculta para fragmentos de frases progresivamente mayores. Con el modelo FGREP la red recibía como entrada una frase completa y las representaciones de las palabras en las unidades de entrada de la red eran sucesivamente modificadas durante el entrenamiento.

Siguiendo las ideas subyacentes en los trabajos presentados en [Pollack,90] y [Miikkulainen,91], y continuando los experimentos presentados en [Castaño,98], en [Casañ,03] se desarrolló un método para generar representaciones distribuidas (de un tamaño preestablecido) de vocabularios, que aprovechaba la capacidad de aprendizaje de un PM. Para ello, se presentaba al PM como entrada y como salida la misma palabra adoptando una codificación local y se obtenían las activaciones que desarrollaba el PM para representar cada palabra en su capa

oculta. Que el PM obtenga su propia representación interna es sencillo, basta con que el número de unidades ocultas sea distinto al número de entradas y salidas de la red. Como ejemplo, para conseguir codificaciones distribuidas de tamaño 10 se entrena un PM con una capa oculta de este tamaño (con este número de neuronas) siguiendo el mecanismo descrito anteriormente y la representación de una palabra se obtiene introduciéndola como entrada al PM entrenado y extrayendo los valores de las activaciones de las neuronas ocultas que éste proporciona. Cabe hacer notar que, a diferencia de las máquinas RAAM [Pollack,90], no se divide la entrada y la salida en dos partes, una para la palabra actual y otra para toda la estructura que se ha presentado anteriormente a la red (un fragmento de frase etiquetada); tampoco se realimenta a la entrada y a la salida de la red las codificaciones producidas por la capa oculta, como ocurría en el codificador FGREP [Miikkulainen,91].

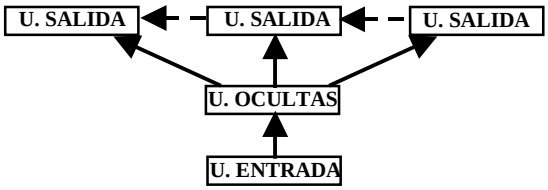

Figura 2. Perceptrón multicapa con retrasos de tiempo en la salida.

En la experimentación presentada en este artículo se trabaja con un PM como el de la figura 2, en el que en la capa de salida se han incorporado retrasos de tiempo. La inclusión de esta ventana de salida se debe al hecho de que en una frase las palabras están relacionadas entre sí, y las activaciones de las neuronas de la capa oculta de dicho modelo conexionista reflejan la codificación de la palabra de entrada y su contexto. Los experimentos de [Casañ,03] demostraban que, proporcionando dicha ventana a la red codificadora, se obtenían mejores resultados de traducción. Este esquema es en cierta forma similar a la salida de las máquinas RAAM, con la diferencia fundamental de que no se lleva al extremo para representar toda la frase en una única codificación. Por otra parte, esta arquitectura está directamente relacionada con la teoría del ámbito de la psicología denominada, en inglés, "syntactic bootstraping", que sostiene que el método de aprendizaje de las funciones de nuevas palabras se ve determinado por las palabras que han aparecido anteriormente en contextos similares. Así, se cree que el niño aprende a relacionar unas palabras con otras según los contextos que comparten. En [Desai,02] se puede encontrar una explicación de esta y otras teorías relacionadas, así como referencias a trabajos sobre el tema.

Por otro lado, los experimentos presentados en [Casañ,03] mostraban que se obtenían mejores resultados de traducción cuando la palabra que deseábamos representar tenía igual o más importancia que su contexto a la salida del PM codificador. Para potenciar el peso de la palabra de entrada frente al contexto se recurre a un procedimiento muy sencillo: se repite en la ventana de salida varias veces la palabra de entrada. Este procedimiento es equivalente a utilizar una función de aprendizaje que dé más importancia al error producido por las activaciones que proporciona la palabra de entrada del PM a la salida, que al generado por las palabras de su contexto.

3.2. Poda del perceptrón multicapa

El mecanismo de generación de codificaciones presentado en la sección anterior no es completamente automático, ya que requiere establecer un tamaño a priori de las codificaciones distribuidas a obtener. En esta sección se presenta un método para determinar de forma automática un posible tamaño de las codificaciones mediante la eliminación progresiva de unidades de la capa oculta del PM, basado en el algoritmo de "Esqueletonización" [Mozer,90]. Dado un PM determinado, este método decide qué unidad eliminar analizando el cambio que se produciría en la función del error si se eliminase cada unidad. Para ello, para cada neurona *i* se introduce una *fuerza de atención* α_i que lleva a una fórmula diferente para la entrada de cada unidad *j*:

$$net_j = \Sigma_i \, \omega_{ij} * \alpha_i * o_i$$

donde net_j es la entrada a la unidad *j* del PM, ω_{ij} es el peso correspondiente a la conexión entre las neuronas *i* y *j* y o_i es la salida de la unidad *i*.

Definiendo la *relevancia de una unidad i*, ρ_i, como el cambio en la función del error que se produce al eliminar dicha unidad se obtiene:

$$\rho_i = E_{(\alpha_i=0)} - E_{(\alpha_i=1)}$$

donde $E_{(\alpha_i=0)}$ es la función del error con la unidad *i* presente y $E_{(\alpha_i=1)}$ si prescindimos de ella. La relevancia de una neurona *i* nos permite calcular, por lo tanto, el efecto que produce en la red la eliminación de dicha neurona.

Como función lineal del error se utiliza:

$$E = \Sigma_j \, |t_j - o_j|$$

donde t_j es el valor esperado para la neurona de salida j y o_j es el valor producido como salida por dicha neurona.

Tras seleccionar la neurona oculta a eliminar, el entrenamiento del PM podado continúa durante un cierto número de iteraciones, tras las cuales se repite de nuevo la poda. El proceso se detiene cuando el Error Cuadrático Medio (ECM) residual (sobre el conjunto de entrenamiento) producido por la red podada seleccionada vuelve a aumentar respecto a las redes podadas seleccionadas en anteriores iteraciones.

3.3. El traductor conexionista RECONTRA

Las codificaciones distribuidas generadas mediante los mecanismos planteados en las dos secciones anteriores se evaluaron sobre una tarea de TA utilizando un traductor conexionista denominado RECONTRA (del inglés "REcurrent CONectionist TRAnslator") [Castaño,99]. Dicho modelo se basa en una Red Neuronal Recurrente Simple presentada por Elman [Elman,90], que incorpora una recurrencia simple en un PM mediante la reintroducción en la capa oculta de las activaciones de dicha capa oculta en el instante de tiempo anterior. Así se consigue que la capa oculta tenga una memoria explícita de su salida en instantes precedentes.

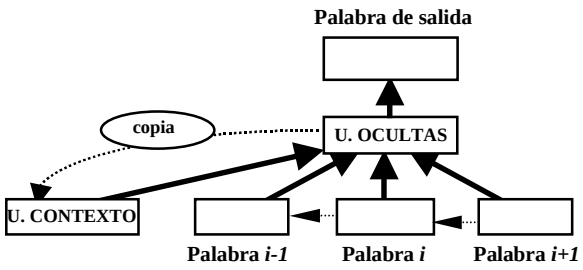

Figura 3. Traductor conexionista RECONTRA.

El traductor conexionista RECONTRA es una red de Elman a la que se incorpora una ventana con retrasos y adelantamientos de tiempo en la entrada de la red. La topología resultante puede verse en la figura 3. Así pues, en RECONTRA se utilizan dos métodos para representar el tiempo, redes dinámicas (recurrentes) y ventanas deslizantes en la entrada. Con las recurrencias simples de la red de Elman se consigue que la capa oculta tenga una memoria explícita de lo acontecido en instantes precedentes. Y la incorporación de una ventana a la entrada de la red permite incrementar la información contextual de la señal de entrada, pudiendo tener en cuenta tanto las palabras precedentes como las posteriores a la palabra de entrada en la frase del lenguaje fuente a traducir.

Al traductor se le presentan secuencialmente las palabras de la frase de entrada y se entrena para que proporcione, también secuencialmente y de manera continua, las palabras que constituyen la frase traducida. Los retrasos de tiempo introducidos en la capa de entrada permitirán ver a la red simultáneamente $m+1+n$ palabras consecutivas de la frase de entrada, las palabras *i-m ... i-1 i i+1 ... i+n*; en el sigiente paso se presentarán las palabras *i-m+1 ... i-1 i i+1... i+n+1* y así sucesivamente; esto es, el traductor irá viendo la frase de entrada a través de una ventana de $m+1+n$ palabras que se desplaza palaba a palabra, y proporcionará una a una las sucesivas palabras de la correspondiente frase traducida.

La figura 4 muestra una posible presentación a la red de la frase en español

¿ les importaría bajar el equipaje a recepción ?

y las correspondientes palabras en inglés que debería mostrar la red en cada momento a la salida (activaciones de salida deseadas), asumiendo que la traducción al inglés de dicha frase es

would you mind sending the luggage down to reception ?
FinFrase

Paso	Ventana de palabras de entrada			Salida
1	……..	¿	les	would
2	¿	les	importaría	you
3	les	importaría	bajar	mind
4	importaría	bajar	el	sending
5	bajar	el	equipaje	the
6	el	equipaje	a	luggage
7	equipaje	a	recepción	down
8	a	recepción	?	to
9	recepción	?	……..	reception
10	?		……..	?
11	……..	……..	……..	FinFrase

Figura 4. Sucesivas entradas y salidas deseadas de un traductor RECONTRA con una ventana de 3 (1+1+1) palabras ante la siguiente frase fuente a traducir: *¿ les importaría bajar el equipaje a recepción ?*

Nótese que el tamaño de la ventana debe ser tal que la red haya visto a la entrada información suficiente como para poder traducir en su momento las palabras de salida; esto es, la red no podría traducir algo que aún no ha visto. Sin embargo, no parece preciso que la palabra o palabras del lenguaje fuente relacionadas con la palabra traducida en la salida estén justamente en la ventana de entrada, ya que, en principio, la red dispone de memoria y es capaz de recordar eventos pasados.

Con el fin de identificar el final de la frase traducida por la red, se añade al vocabulario del lenguaje destino una palabra específica, *FinFrase*, dedicada a tal efecto. De esta manera, el proceso de presentación de la frase de entrada finalizará cuando la red proporcione dicha palabra como salida o, en

su defecto, tras la presentación a la red de un número razonable (predeterminado) de ventanas de entrada completamente vacías (con palabras en blanco).

Todos los modelos conexionistas utilizados en la experimentación presentada en este artículo han sido implementados utilizando el simulador neuronal SNNS [Zell,95].

4. La tarea del Turista

Como ejemplo de aplicación del modelo codificador anterior se empleó la tarea de TA texto-a-texto del Turista, diseñada recientemente en el marco del proyecto EuTrans-I [Amengual,01]. Esta tarea aborda situaciones típicas en las que puede encontrarse un turista que visita un país extranjero y cuya lengua desconoce. El escenario contemplado para las posibles traducciones es la comunicación persona-a-persona en la recepción de un hotel.

Aunque en el proyecto Eutrans-I se contemplaron tres pares de lenguajes en las traducciones: español-inglés, español-alemán y español-italiano, nosotros consideraremos únicamente el caso español-inglés.

Los pares de frases para entrenar y probar los traductores se generaron a partir de esquemas de traducción dirigidos por la sintaxis.

4.1. Una subtarea del Turista

En los experimentos abordados en este trabajo con el traductor RECONTRA nos centramos en una subtarea dentro de la tarea del Turista. En ella se incluyeron frases en las que el turista notifica su partida, solicita la factura, se queja o pregunta sobre la factura o pide que, ante su marcha, sea trasladado su equipaje.

Con el fin de incrementar progresivamente el tamaño de los vocabularios y la complejidad de la subtarea, se agruparon algunas palabras y algunas secuencias de palabras en *categorías*. Concretamente, se consideraron dos categorías que representaban a horas y fechas genéricas etiquetadas, respectivamente, por $HORA y $FECHA. En un primer experimento se consideraron pares de frases categorizados (que contemplaban las anteriores categorías) y, posteriormente, los correspondientes pares de frases no categorizados. En lo que sigue nos referiremos a ellos como la *tarea del Turista categorizada* y *no categorizada*, respectivamente.

El vocabulario castellano de la tarea no categorizada tenía 178 palabras diferentes que, tras la categorización, se redujeron a 132 palabras. El vocabulario inglés tenía 140 y 82 palabras en las correspondientes tarea no categorizada y categorizada. La figura 5 muestra algunos ejemplos de las traducciones español-inglés tanto no categorizadas como categorizadas.

Pares de frases no categorizadas	
Español:	He de marcharme el día veintisiete de febrero a las siete y media de la tarde .
Inglés:	I should leave on Febrary the twenty-seventh at half past seven in the afternoon .
Español:	¿Podemos abonar en efectivo ?
Inglés:	Can we pay in cash ?
Pares de frases categorizados	
Español:	Nos marchamos hoy mismo a $HORA por la noche .
Inglés:	We are leaving today at $HORA in the evening .
Español:	Me voy a ir el día $FECHA a $HORA de la mañana .
Inglés:	I am leaving on $FECHA at $HOUR in the morning .

Figura 5. Ejemplos de pares de frases de la tarea del Turista no categorizada y de la tarea categorizada.

4.2. Corpora de aprendizaje y prueba

Los corpora adoptados en las tareas de TA abordadas en el artículo estaban formados por pares de frases texto-a-texto. Cada par consistía en una frase en español y la correspondiente traducción al inglés. Los traductores de la tarea no categorizada se entrenaron con 5.000 pares de frases y los resultantes modelos entrenados se evaluaron sobre 1.000 pares diferentes. Estos corpora de entrenamiento y prueba se categorizaron posteriormente y fueron empleados para la tarea categorizada.

De los 5.000 pares de entrenamiento no categorizados, 3.425 pares eran diferentes; tras categorizar estos 5.000 pares, el número de pares diferentes se redujo a 2.687. En el corpus de test, 991 de los 1.000 pares no categorizados eran diferentes y, tras la categorización, este número bajó a 771.

No había solapamiento entre los corpora de aprendizaje y prueba no categorizados; sin embargo, el 54% de los pares del conjunto de test categorizado formaban parte de la muestra de entrenamiento categorizada.

La longitud de las frases en castellano no categorizadas oscilaba entre 3 y 20 palabras, y la longitud de las frases inglesas no categorizadas, entre 3 y 17. El número de palabras de las frases categorizadas variaba entre 3 y 13 para el castellano y entre 3 y 12 para el inglés.

Los conjuntos de entrenamiento para los PM codificadores se construyeron a partir de los correspondientes corpora de aprendizaje de los traductores. Para ello, se extraían de estos conjuntos las diferentes palabras del vocabulario de un lenguaje dado junto con los posibles contextos en los que éstas aparecían. No se consideraron corpora de test para los PM ya que estos se evaluaban indirectamente a partir de las traducciones de prueba proporcionadas por RECONTRA.

5. Experimentación

5.1. Entrenamiento de los modelos

5.1.1. Entrenamiento del codificador

El modelo de PM codificador (descrito en las secciones 3.1 y 3.2) que se adoptó en todos los experimentos con la (sub)tarea del Turista tenía las siguientes características:

- Topología con dos capas de conexiones.
- Una palabra de entrada y varias de salida para recoger el contexto de la palabra de entrada (trabajando con un total de 4, 6 u 8 palabras de salida), todas ellas presentadas a la red con una codificación local.
- 25 unidades ocultas.

Para cada experimento se procedió, en primer lugar, a inicializar las conexiones de la red codificadora con valores aleatorios generados dentro del rango [-0.01, 0.01]. A continuación, se realizó la estimación de los parámetros de entrenamiento de la red (factor de aprendizaje y momentum). Posteriormente, se entrenó ésta durante 1.000 iteraciones con el conjunto de entrenamiento utilizando el algoritmo de entrenamiento de Retropropagación hacia Atrás del Error ("Backward Error Propagation") con momentum [Rumelhart,86]. Como función de activación no lineal se adoptó la función sigmoide asimétrica definida en el intervalo [0,1].

Una vez entrenada la red codificadora, se le presentaron todas las palabras del vocabulario, extrayendo las activaciones de la capa oculta para cada una de ellas, y se procedió a abordar la tarea de TA con estas codificaciones.

5.1.2. Entrenamiento del traductor

El modelo del traductor RECONTRA utilizado en la experimentación tenía las siguientes características:

- Ventana de entrada con 6 (3+1+2) palabras de entrada. Para la tarea no categorizada también se utilizaron en algunos casos 8 (3+1+4) palabras.
- 140 unidades ocultas. Para la tarea no categorizada también se hicieron experimentos puntuales con otros tamaños (160, 180, 220 y 250 neuronas).

El ajuste de las conexiones de los traductores RECONTRA se realizó siguiendo una adaptación del algoritmo de Retropropagación hacia Atrás del Error [Rumelhart,86] que truncaba el gradiente del error; esto es, que no tenía en cuenta las recurrencias de la red, ya que consideraba dichas recurrencias como entradas adicionales a la arquitectura. Consecuentemente, en la estimación de pesos se truncaba el gradiente del error y no se calculaba de manera exacta. Pese a ello, este método de aprendizaje funciona bien en la práctica, tal y como se demuestra empíricamente en [Castaño,97], donde se compara con métodos (computacionalmente más costosos) que calculan el gradiente de forma exacta. Los pesos se actualizaron entrada a entrada ("en línea") tras procesar cada par de entrenamiento "ventana de entrada – palabra de salida traducida" hasta que se alcanzaba el final de la frase traducida.

Al comienzo de cada frase se procedía a la inicialización de las unidades de contexto con el fin de que la red no dispusiese inicialmente de información contextual alguna. Teniendo en cuenta que en la experimentación se adoptó la función sigmoide definida entre 0 y 1 como función de activación no lineal, el valor inicial de cada neurona de contexto fue de 0,5. Por otro lado, antes de comenzar el entrenamiento en sí de la red, las conexiones de ésta se inicializaron a valores aleatorios dentro del rango [-0.01, 0.01] y, a partir de ellas, se realizó la estimación del factor de aprendizaje y del momentum.

Los traductores se entrenaron durante 500 iteraciones con el conjunto de aprendizaje. En cada una de las iteraciones las frases se presentaban a la red de manera aleatoria. Posteriormente se "congelaron" las conexiones de las redes traductoras entrenadas y se evaluó su funcionamiento sobre el corpus de pares de frases de prueba, obteniendo las tasas de traducciones correctas.

5.2. Medidas de éxito de las traducciones

Una frase se consideró bien traducida si las palabras de salida proporcionadas por la red (entrenada) coincidían exactamente con las palabras de la frase destino esperada. Además de proporcionar el

porcentaje de *frases* de prueba *bien traducidas* (FBT), en el proceso de evaluación también resulta útil analizar el porcentaje de *palabras bien traducidas* (PBT). Esta última medida se obtuvo comparando cada frase de prueba traducida por la red con su traducción esperada y usando un mecanismo convencional de distancia de edición, basado en técnicas de programación dinámica [Marzal,93], que calculaba el número de errores de inserción, sustitución y borrado requeridos para convertir una frase en la otra. Los porcentajes de aciertos a nivel de palabra presentados aquí corresponden al cociente entre el número total de no errores respecto al número total de operaciones de edición (no errores + errores).

5.3. Resultados con codificaciones manuales

Con el fin de comparar los resultados presentados en las siguientes secciones, conseguidos con el traductor RECONTRA utilizando codificaciones distribuidas obtenidas automáticamente, en la tabla 1 se muestran las mejores tasas de traducción sobre el corpus de prueba alcanzadas con dicho traductor para la tarea categorizada y no categorizada utilizando codificaciones distribuidas binarias creadas manualmente. En dichas codificaciones manuales se asignaban representaciones similares a palabras próximas semánticamente. En la tabla también puede verse el tamaño adoptado para las codificaciones del vocabulario castellano y para las codificaciones del inglés (denotadas, respectivamente, por |Castellano| e |Inglés|).

Tarea	|Castellano|/|Inglés|	FBT	PBT
Categorizada	25/25	98,40	99,72
No-categorizada	61/52	90,40	98,60

Tabla 1. Porcentajes de FBT y PBT utilizando codificaciones manuales.

5.4. La subtarea del Turista categorizada

En primer lugar se realizaron una serie de experimentos sobre la tarea categorizada con PM en los que la capa oculta (o lo que es lo mismo, las codificaciones de los vocabularios) tenía un tamaño fijo preestablecido. Posteriormente se empleó el algoritmo de poda para determinar automáticamente un posible tamaño de estas codificaciones.

En todos los experimentos, el número de unidades de entrada y de salida del PM para representar cada palabra fue de 132 para el castellano y de 82 para el inglés (recordemos que se utilizaba una codificación local a la entrada y a la salida).

5.4.1. Resultados con codificaciones de tamaño fijo

Con el fin de obtener codificaciones distribuidas automáticas (de tamaño 25) para los vocabularios de la tarea categorizada, se entrenaron PM (con 25 neuronas ocultas y) con ventanas de salida de distinto tamaño (4, 6 y 8 palabras) y distinto número de repeticiones (2, 4, y 6) de la palabra de entrada en la salida durante 1.000 presentaciones del conjunto de entrenamiento. A continuación, se extrajeron de ellos las representaciones reales conseguidas para cada palabra de los vocabularios de la tarea de TA.

Para no aumentar excesivamente el número de experimentos de traducción a realizar, se decidió utilizar las mismas características para el PM codificador del lenguaje fuente que para el PM codificador del lenguaje destino. Es decir, que las codificaciones reales para el vocabulario castellano extraídas de un PM con ventana de salida de tamaño 4 y 2 repeticiones de la palabra de entrada se emparejaron con las codificaciones para el vocabulario inglés extraídas de un PM con ventana de salida de tamaño 4 y 2 repeticiones de la palabra de entrada.

Una vez entrenados los traductores con dichas codificaciones, se extrajeron las tasas de traducción sobre el conjunto de prueba. Los resultados, mostrados en la tabla 2, fueron muy buenos, y apenas ligeramente peores que los obtenidos con codificaciones distribuidas manuales del mismo tamaño (tabla 1).

|Ventana de salida PM|	Ventana de salida PM	FBT	PBT
4	x-1 **x x** x+1	98,00	99,64
6	x-1 **x x x x** x+1	97,70	99,59
8	x-2 x-1 **x x x x** x+1 x+2	97,60	99,49
8	x-1 **x x x x x x** x+1	96,70	99,51

Tabla 2. Porcentajes de FBT y PBT de la tarea categorizada utilizando codificaciones automáticas de tamaño fijo (25) extraídas de PM con diversos tamaños de ventana de salida.

5.4.2. Resultados con codificaciones podadas

En la experimentación llevada a cabo en el anterior punto, un paso del proceso de traducción continuaba siendo manual. A pesar de que las codificaciones se obtenían automáticamente de PM entrenados al efecto, el tamaño de estas codificaciones aún tenía que ser determinado por el experto humano. Para automatizar este proceso, se puede utilizar un algoritmo de poda que elimine progresivamente

neuronas de la capa oculta del PM y, por tanto, reduzca el tamaño de las codificaciones.

A dos de los PM codificadores utilizados en los experimentos del punto anterior se les aplicó sobre las neuronas de la capa oculta el algoritmo de poda "Esqueletonización", descrito en la sección 3.2. Concretamente, se seleccionaron los PM (para castellano e inglés) con ventana de salida de tamaño 4 y 2 repeticiones de la palabra de entrada a la salida. Estos PM tenían inicialmente 25 unidades en la capa oculta, que se podaron sucesivamente cada 10 presentaciones completas (iteraciones) de la muestra de aprendizaje. En cada uno de los PM se seleccionó la red cuyo ECM era mínimo y se extrajeron de ella las codificaciones correspondientes. Para el PM codificador del castellano se consiguió un tamaño de 9 unidades ocultas y para el del inglés, de 10; es decir, el tamaño de las codificaciones era de 9 y 10 unidades para el vocabulario castellano e inglés, respectivamente.

Algunos experimentos (no presentados en este artículo) demostraron que las codificaciones obtenidas tras podar PM entrenados durante 1.000 presentaciones del conjunto de aprendizaje no proporcionaban buenas tasas de traducción. Así pues, las redes codificadoras podadas arriba indicadas se entrenaron hasta 3.000 iteraciones. Las tasas de traducción obtenidas con las codificaciones extraídas tras entrenar los PM durante 1.000 y 3.000 iteraciones del conjunto de entrenamiento aparecen en la tabla 3. Como puede observarse, los porcentajes de traducción obtenidos son similares en ambos casos, si bien son ligeramente mejores para 1.000 iteraciones del PM. Por otro lado, los buenos resultados que se observan en la tabla muestran claramente que este método es válido para reducir el tamaño de las codificaciones y mantener los porcentajes de frases y palabras bien traducidas.

|Iteraciones PM|	Codificación Traductor |Español|/|Inglés|	FBT	PBT
1.000	9/10	96,50	99,00
3.000	9/10	95,20	98,88

Tabla 3. Porcentajes de FBT y PBT de la tarea categorizada con codificaciones extraídas tras podar PM con formato x-1 x x x+1 para la ventana de salida y entrenarlos durante diferentes presentaciones del conjunto de aprendizaje.

5.5. Resultados con la tarea no categorizada

Atendiendo a los resultados prometedores alcanzados con la tarea del Turista categorizada, el siguiente paso fue repetir la experimentación con la tarea sin categorías, un poco más compleja que la anterior y con mayor vocabulario. Al igual que con aquélla, en primer lugar se entrenaron PM con codificaciones de tamaño fijo y, posteriormente, PM podados que proporcionaron codificaciones más compactas con un posible tamaño calculado automáticamente.

En todos los experimentos, el número de unidades de entrada y de salida del PM para representar cada palabra fue de 178 para el castellano y de 140 para el inglés.

5.5.1. Resultados con codificaciones de tamaño fijo

Con el fin de obtener codificaciones distribuidas automáticas (de tamaño 25) para los vocabularios de la tarea no categorizada, inicialmente se entrenaron PM (con 25 neuronas ocultas y) con ventanas de salida de distinto tamaño (4, 6 y 8 palabras) y distinto número de repeticiones de la palabra de entrada en la salida (2, 4 y 6 repeticiones) durante 1.000 iteracines del conjunto de aprendizaje. Luego se extrajeron las representaciones reales obtenidas para cada palabra y se emplearon como codificaciones de los vocabularios de la tarea de traducción.

Una vez entrenados los traductores con dichas codificaciones, se calcularon las tasas de traducción sobre el conjunto de prueba. Al analizar los resultados obtenidos (mostrados en la tabla 4) se observa que son mucho peores que los alcanzados con la tarea categorizada. Esto se debe a que RECONTRA confundía sistemáticamente ciertas palabras que tenían las mismas funciones semánticas (principalmente los días del mes).

|Ventana de salida PM|	Ventana de salida PM	FBT	PBT
4	x-1 **x x** x+1	18,70	88,22
6	x-1 **x x x x** x+1	62,40	94,16
8	x-2 x-1 **x x x x** x+1 x+2	45,40	91,25
8	x-1 **x x x x x x** x+1	50,40	92,68

Tabla 4. Porcentajes de FBT y PBT para la tarea no categorizada utilizando codificaciones automáticas de tamaño fijo (25) extraídas de PM con diversos tamaños de ventana de salida.

Con el fin de mejorar las codificaciones, se realizaron nuevos experimentos a partir de las codificaciones extraídas de PM con ventana de salida de tamaño 6 y 4 repeticiones de la palabra de entrada, alargando tanto el entrenamiento del PM codificador como el del traductor, aumentando el tamaño de la capa oculta y/o de la ventana de entrada de este último. La tabla 5 recoge los valores específicos de estos datos y las tasas de traducción alcanzadas sobre el conjunto de prueba.

PM	RECONTRA			FBT	PBT
|Iteraciones|	|Neuronas ocultas|	|Ventana entrada|	|Iteraciones|		
1.000	140	6	500	62,40	94,16
		6	1.000	62,60	94,10
	160	6	500	57,20	93,04
	180	8	500	73,10	95,74
3.000	140	6	500	22,80	88,67

Tabla 5. Porcentajes de FBT y PBT de la tarea no categorizada con traductores con distintas características, usando codificaciones extraídas de PM con ventana de salida con formato x-1 x x x x x+1, tras distintos números de iteraciones.

Al analizar los resultados allí mostrados se observa que, alargando el entrenamiento del PM hasta 3.000 presentaciones del conjunto de entrenamiento, empeoraron los porcentajes de traducción, debido probablemente a un fenómeno de sobreentrenamiento. Por otro lado, alargar el entrenamiento de RECONTRA hasta 1.000 presentaciones no produjo diferencias significativas.

Respecto al incremento del número de neuronas en la capa oculta de RECONTRA, en la tabla 5 se ve que se produjo una ligera disminución en los resultados al aumentar el tamaño de la capa oculta de 140 a 160 neuronas, lo cual parece indicar que no existe un problema de capacidad de la red.

Finalmente, ampliando la ventana de entrada de RECONTRA, los resultados obtenidos parecen confirmar que el tamaño de la ventana de entrada podría ser parte fundamental del problema.

5.5.2. Resultados con codificaciones podadas

A los PM de experimentos anteriores con distintos tamaños de ventana de salida y número de repeticiones de la palabra de entrada se les aplicó el algoritmo de poda. En cada caso se seleccionó el PM cuyo ECM residual era mínimo y se extrajeron de él las codificaciones correspondientes tras entrenarlo durante 3.000 presentaciones del conjunto de entrenamiento.

Estas codificaciones se utilizaron para abordar la tarea no categorizada mediante traductores RECONTRA con ventana de entrada de tamaño 6 y 140 unidades en la capa oculta. En la tabla 6 se pueden observar las tasas de traducción alcanzadas sobre el corpus de prueba, así como los tamaños de las codificaciones obtenidas mediante la poda. Los resultados muestran que, en general, se ha producido un descenso en los porcentajes de FBT y PBT (comparar con la tabla 4).

Ventana salida PM	|Castellano|/ |Inglés|	FBT	PBT
x-1 **x x** x+1	11/9	25,80	88,79
x-1 **x x x x** x+1	11/11	32,00	90,22
x-2 x-1 **x x x x** x+1 x+2	14/13	40,30	90,23
x-1 **x x x x x x** x+1	11/9	27,20	88,05

Tabla 6. Porcentajes de FBT y PBT para la tarea no categorizada obtenidos con un traductor con ventana de entrada de tamaño 6 y 140 unidades en la capa oculta usando codificaciones extraídas de PM podados.

Dado que los experimentos sin podar los PM codificadores habían sugerido que un mayor tamaño de la ventana de entrada de RECONTRA mejoraba las tasas de traducción, se aplicaron las dos mejores codificaciones de la tabla 6 sobre traductores RECONTRA con diferentes tamaños de la ventana de entrada y también con diferentes números de unidades ocultas. Los resultados sobre el corpus de prueba pueden verse en la tabla 7 y reflejan que, efectivamente, el incremento de estos dos parámetros de RECONTRA mejoraba el proceso de traducción.

PM	RECONTRA			FBT	PBT
Ventana salida	|Neuronas ocultas|	|Ventana entrada|	|Castellano| /|Inglés|		
x-1 **x x x x** x+1	140	6	11/11	32,00	90,22
	140	10		29,20	89,27
	180	8		42,30	91,89
x-2 x-1 **x x x x** x+1 x+2	140	6	14/13	40,30	90,23
	180	8		51,10	92,17
	220	8		54,60	93,59
	250	8		52,90	93,730

Tabla 7. Porcentajes de FBT y PBT de la tarea no categorizada obtenidos con un traductor con diversas características, utilizando codificaciones obtenidas tras podar PM con diferentes contextos de salida.

Por otro lado, una forma de mejorar las codificaciones obtenidas con los PM codificadores podría ser volver a estimar el factor de aprendizaje y el momentum del PM podado, ya que la topología de éste varía tras la poda y la estimación de dichos parámetros se realiza antes del proceso de poda. Se hizo esta operación con el PM cuyo contexto de salida era *x-2 x-1 x x x x x+1 x+2* (esto es, con ventana de salida de tamaño 8 y 4 repeticiones de la palabra de entrada en la salida). Tras el entrenamiento de éste (durante 3.000 iteraciones con el conjunto de aprendizaje) se extrajeron las codificaciones de la capa oculta y con ellas se entrenó un traductor con 180 neuronas ocultas y una ventana de entrada de tamaño 8 (3+1+4) palabras. Los resultados de traducción alcanzados sobre el corpus de prueba pueden verse en la tabla 8. Y, como en ella se aprecia, la estimación de parámetros sobre el PM podado (y no únicamente sobre el PM

de partida) mejoraron sensiblemente las traducciones.

Estimación tras poda	FBT	PBT
No	51,10	92,17
Sí	67,40	95,34

Tabla 8. Porcentajes de FBT y PBT para la tarea no categorizada obtenidos con codificaciones extraídas de PM podados con ventana de salida 8 y 4 repeticiones de la palabra de entrada a la salida, reestimando parámetros del PM podado.

6. Conclusiones y trabajo futuro

Este artículo propone un método para determinar automáticamente el tamaño de las codificaciones distribuidas de los léxicos implicados en una tarea de traducción texto-a-texto, abordada con el traductor conexionista RECONTRA. El método extrae las codificaciones de la capa oculta de un PM con una ventana de retrasos a la salida. Aplicando un sencillo mecanismo de poda para determinar el tamaño de la capa oculta del PM se ha logrado reducir de modo sustancial el tamaño de la codificación.

Aunque los resultados de traducción obtenidos con el mecanismo de poda no son en todos los casos tan buenos como sería deseable, el método de generación de codificaciones es susceptible de ser mejorado. También cabría aplicar nuevos y mejores algoritmos de poda al PM para obtener codificaciones de vocabularios más discriminativas y con mejores tamaños de éstas. Cabe también abordar tareas de TA con mayores léxicos y con dominios semánticos más amplios que estén más cercanas a procesos reales de TA. Será necesario también abordar traducciones entre pares de lenguajes distintos a los aquí empleados.

Referencias

[Amengual,01] J.C. Amengual, M.A. Castaño, A. Castellanos, D. Llorens, A. Marzal, F. Prat, J.M. Vilar J.M. Benedí, F. Casacuberta, M. Pastor, E. Vidal. *The EUTRANS-I Spoken Language Translation System.* Machine Translation no. 15, pp.75—103. Kluwer Academic Publishers. 2001.

[Casañ,99] G. A. Casañ, M. A. Castaño, *Distributed Representation of Vocabularies in the RECONTRA Neural Translator*. Procs. of the 6th European Conference on Speech Communication and Technology, vol. 6, pp. 2423—2426, Budapest, Septiembre 1999.

[Casañ,03] G. A. Casañ, M. A. Castaño. A*utomatic Word Codification for the RECONTRA Connectionist Translator.* In "Lecture Notes in Computer Science: Pattern Recognition and Image Analysis", vol. 2652, pp. 168—175, F.J. Perarles, A.J.C. Campilho, N. Pérez de la Blanca, A. Sanfeliú (Eds.), Springer-Verlag, 2003.

[Castaño,97] M. A. Castaño, F. Casacuberta. *Training Simple Recurrent Networks through Gradient Descent Algorithms.* In "Lecture Notes in Computer Science: Biological and Atificial Computation: From Neuroscience to Technology", vol. 1240, pp. 493—500, J. Mira, R. Moreno-Díaz, J. Cabestany (Eds.), Springer-Verlag, 1998.

[Castaño,98] M. A. Castaño. *Redes Neuronales Recurrentes para Inferencia Gramatical y Traducción Automática.* PhD. Dissertation, Dpto. Lenguajes y Sistemas Informáticos, Universidad Politécnica de Valencia, Valencia, Febrero 1998.

[Castaño,99] M. A. Castaño, F. Casacuberta. *Text-to-Text Machine Translation Using the RECONTRA Connectionist Model*. In "Lecture Notes in Computer Science: Applications of Bio-Inspired Artificial Neural Networks", vol. 1607, pp. 683—692, J. Mira, J.V. Sánchez-Andrés (Eds.), Springer-Verlag, 1999.

[Desai,02] R. Desai. *Bootstrapping in miniature language adquisition*. Cognitive Systems Research, vol. 3, no. 1, pp. 15—23, Marzo 2002.

[Elman,90] J.L. Elman. *Finding Structure in Time.* Cognitive Science, vol. 2, no. 4, pp. 279—311, 1990.

[Koncar,94] N. Koncar, G. Guthire. *A Natural Language Translation Neural Network.* Procs. of the Int. Conf. On New Methods in Language Processing, pp. 71—77, Manchester, UK. 1994.

[Marzal,93] A. Marzal, E. Vidal. *Computation of Normalized Edit Distance and Applications*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 9. 1993.

[Miikkulainen,91] R. P. Miikkulainen, M. G. Dyer. *Natural Language Processing with Modular Neural Networks and Distributed Lexicon*. Cognitive Science, vol. 15, pp. 393—399, 1991.

[Mozer,90] M.C. Mozer, P. Smolensky. *Skeletonization: a Technique for Trimming the Fat*

from a Network via Relevance Assessment. Advances in Neural Information Processing 1, D.S. Touretzky, Ed. Morgan Kaufmann, pp. 177—185, 1990.

[Pollack,90] J. B.Pollack. *Recursive Distributed Representations*. Artificial Intelligence, vol. 46, pp. 77—105, 1990.

[Rojas,96] R. Rojas. *Neural Networks. A Systematic Introduction.* Springer-Verlag, Berlin Heidelberg, 1996.

[Rumelhart,86] D.E. Rumelhart, G. Hinton, R. Williams. *Learning Sequential Structure in Simple Recurrent Networks.* In "Parallel Distributed Processing: Experiments in the Microstructure of Cognition", vol. 1. Rumelhart D.E., McClelland J.L. and the PDP Research Group (Eds.), MIT Press. Cambridge, 1986.

[Waibel,91] A. Waibel, A.J. Jain, A.E. McNair, H. Saito, A.G. Hauptmann, J. Tebelskis. *JANUS: A Speech-to-Speech Translation System using Connectionist and Symbolic Processing Strategies*. Procs. of the International Conferene on Acoustics, Speech and Signal Processing (ICASSP-91), vol. 2, pp. 793—796, 1991.

[Zell,95] A. Zell et al. *SNNS: Stuttgart Neural Network Simulator. User manual, Version 4.1.* Technical Report no. 6195, Institute for Parallel and Distributed High Performance Systems, University of Stuttgart, 1995.