



Procesamiento del Lenguaje Natural

ISSN: 1135-5948

secretaria.sepln@ujaen.es

Sociedad Española para el  
Procesamiento del Lenguaje Natural  
España

Escolano, Carlos; Costa-jussà, Marta R.

Generación morfológica con algoritmos de aprendizaje profundo integrada en un sistema  
de traducción automática estadística

Procesamiento del Lenguaje Natural, núm. 59, 2017, pp. 107-114

Sociedad Española para el Procesamiento del Lenguaje Natural  
Jaén, España

Disponible en: <http://www.redalyc.org/articulo.oa?id=515754427012>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

# Generación morfológica con algoritmos de aprendizaje profundo integrada en un sistema de traducción automática estadística

## *Integration of morphology generation techniques based on deep learning into a statistical machine translation system*

Carlos Escolano, Marta R. Costa-jussà

TALP Research Center

Universitat Politècnica de Catalunya

Campus Nord, C/Jordi Girona, 08034 Barcelona

carlos.escolano@tsc.upc.edu, marta.ruiz@upc.edu

**Resumen:** La variación morfológica entre un lenguaje fuente y el lenguaje destino genera dificultades a los algoritmos estándares de traducción como el estadístico basado en segmentos. En este trabajo planteamos dividir la tarea de traducción en dos partes: primero, simplificamos el lenguaje destino en términos morfológicos y construimos el sistema de traducción con esta modificación; y después utilizamos un algoritmo de clasificación para generar la morfología final. Este trabajo presenta una arquitectura de aprendizaje profundo que permite añadir de manera efectiva la información morfológica a la traducción simplificada generada por un traductor estadístico basado en segmentos. Demostramos que la arquitectura diseñada presenta resultados superiores a los algoritmos estado-del-arte en términos de precisión y que la calidad de la traducción mejora en términos de METEOR.

**Palabras clave:** Traducción automática, generación morfológica, chino-español

**Abstract:** The morphological variation between a source language and the target language generates difficulties for standard machine translation algorithms such as statistical phrase-based. In this paper, we propose dividing the task of translation in two steps: first, simplify the target language in morphological terms and build the translation system; in a second step, we use a classification algorithm to generate morphology. This paper presents a novel deep learning architecture that allows the effective retrieval of the morphological information to the simplified translation generated by the translation system. We show that the designed architecture improves results compared to state-of-the-art algorithms in terms of accuracy and that the quality of the translation improves in terms of METEOR.

**Keywords:** Machine translation, morphology generation, Chinese-Spanish

## 1 Introducción

La tarea de la traducción automática estadística consiste en obtener la frase destino más probable dada una frase fuente. Para conseguir este objetivo un método popular es el basado en segmentos (Koehn, Och, y Marcu, 2003) que utiliza un corpus paralelo a nivel de oración para aprender modelos estadísticos. Estos modelos estadísticos se combinan en un decodificador que explora un espacio de búsqueda y obtiene la traducción más probable. Estos métodos basados en segmentos han demostrado conseguir resultados estado del arte, pero dada la complejidad de la tarea de traducción todavía quedan mu-

chos retos para resolver. Uno de ellos es el caso de la morfología. Por ejemplo, en pares de lenguas con morfologías muy diferentes (e.g. Chino-Español) donde el lenguaje fuente es poco inflexionado y el lenguaje destino presenta muchas flexiones morfológicas, el sistema basado en segmentos tiene dificultades para generar la forma morfológica adecuada. Principalmente, se debe a que la información morfológica se debe extraer del contexto y el sistema basado en segmentos usa contextos relativamente limitados.

En este artículo presentamos una aproximación para mejorar resultados de traducción usando una arquitectura de traducción

de dos pasos: primero hacemos una traducción a un lenguaje destino simplificado en términos morfológicos y después utilizamos un sistema de clasificación para generar la morfología.

Existe una variedad de trabajos relacionados que simplifican la morfología del lenguaje destino en el sistema de traducción y después generan la morfología. A continuación señalamos los que son más similares a nuestra propuesta: (Toutanova, Suzuki, y Ruopp, 2008) utiliza modelos de máxima entropía para predecir la inflexión; (Clifton y Sarkar, 2011) y (Kholy y Habash, 2012) usan técnicas de campos aleatorios condicionales (CRF) para predecir características morfológicas y (Formiga et al., 2013) utiliza máquinas de vectores de soporte (SVMs). Subrayar que la principal aportación de nuestro trabajo, respecto a estos anteriores, es la nueva arquitectura basada en aprendizaje profundo para generar la morfología y su aplicación a un par de lenguas muy distantes, Chino-Español, en términos de morfología.

Queremos mantener un compromiso entre la simplificación en morfología y la complejidad de la generación morfológica. Así pues, basándonos en investigaciones anteriores (Costa-jussà, 2015), donde se muestra que la simplificación en género y número consigue oráculos cercanos a la simplificación en lemas (para el caso Chino-Español), vamos a enfocarnos en esta simplificación en concreto. Para el módulo de clasificación o generación morfológica proponemos técnicas de aprendizaje profundo que nos permiten conseguir resultados de clasificación que mejoran otras técnicas estándar de aprendizaje automático. Precisamente, esta tarea de obtener la información morfológica de una palabra sin que esta tenga información de la misma (no tenga la flexión) resulta un buen reto en si mismo. Por ello, el mejorar el estado del arte en esta tarea ya supone una contribución relevante para la comunidad científica. Además, presentamos resultados en traducción que mejoran el sistema de referencia en términos de la medida estándar de evaluación METEOR (Denkowski y Lavie, 2014).

## 2 Sistema de traducción

En una primera fase, entrenamos un modelo de traducción que nos sirva como base sobre la que aplicar la clasificación y de referencia para medir la mejora de la traducción

tras aplicar nuestro sistema. Para ello utilizaremos *Moses* (K. et al., 2007) como herramienta, que nos permite entrenar y optimizar un modelo de traducción a partir de un corpus de texto paralelo.

Utilizando el analizador de lenguaje *Free-ling* (Padró y Stanilovsky, 2012) realizamos diferentes representaciones del texto destino (ver ejemplo en la Tabla 1). Y para cada representación del texto destino, construimos un sistema de traducción basado en Moses. Las diferentes representaciones son:

- **Texto original:** Es el texto sin modificaciones, en el que cada palabra en chino se corresponde con su correspondiente traducción en castellano. El modelo creado con esta representación nos servirá como referencia para medir los resultados generados.
- **Texto simplificado:** En esta representación eliminamos de las etiquetas toda la información referente a la morfología de la palabra, en términos de género y número. El modelo obtenido nos permitirá acotar la mejora máxima que puede proporcionar nuestro sistema, ya que al eliminar esta información no estamos creando errores a causa de ellas y sería equivalente a predecirla correctamente en todos los casos.
- **Género simplificado:** En esta representación eliminamos la información sobre género del texto en castellano. Utilizaremos este modelo como cota de la mejora posible de nuestro sistema aplicado únicamente al género.
- **Número simplificado:** De forma análoga al género simplificado, este modelo nos permite conocer la mejora máxima que podemos obtener utilizando nuestro sistema únicamente sobre número.

## 3 Selección de características

Hemos de decidir como representaremos el texto anterior y qué información pasaremos al clasificador morfológico con tal que reciba información relevante para la tarea. Hemos experimentado con diversas estrategias entre las que podemos diferenciar aquellas en las que utilizábamos información bilingüe (incorporando información del lenguaje fuente

Modelo	Texto
Original	La casa de la playa
Texto etiquetado	DA0FS0[el] NCFS000[casa] SPS00[de] DA0FS00[el] NCFS000[playa]
Simplificado	DA00[el] NC000[casa] SPS00[de] DA00[el] NC000[playa]
Simplificado género	DA0S0[el] NCS000[casa] SPS00[de] DA0S0[el] NCS000[playa]
Simplificado número	DA0F0[el] NCF000[casa] SPS00[de] DA0FS[el] NCF000[playa]

Tabla 1: Ejemplo de representaciones de texto utilizadas

y destino) y aquellas basadas en sólo uno de ellos.

A continuación, en la subsección 3.1, presentamos las características definitivas que fueron las ventanas de palabras. Y en la siguiente subsección 3.2, comentamos con qué otras características experimentamos (sin obtener mejoras significativas).

### 3.1 Ventanas de palabras

Las ventanas de palabras del lenguaje destino simplificado fueron las características que utilizamos como entrada al clasificador.

Definimos ventana como una lista de palabras de tamaño  $n$  en el cual el elemento central es aquel que queremos clasificar. El resto de la ventana contiene las palabras que preceden y siguen a la palabra en la frase manteniendo su orden.

Añadimos los caracteres especiales “< s >” y “< e >” para marcar el inicio y el final de la frase respectivamente, para asegurar que todas las ventanas tengan el mismo tamaño. En la figura 1 podemos ver un ejemplo de ventanas aplicadas a una oración:

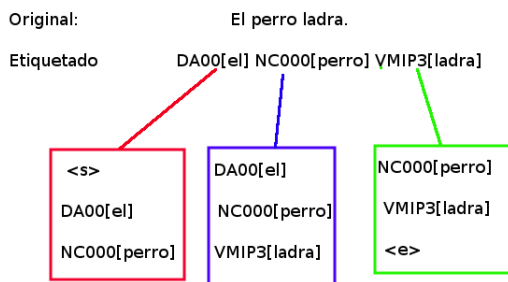


Figura 1: Ejemplo de representación en ventanas

El objetivo de este método es que los clasificadores utilicen únicamente el contexto de la palabra en la oración para predecir los resultados.

Esta representación nos presenta diversas ventajas sobre otras alternativas:

- Al utilizar únicamente características del castellano evitamos errores producidos por el alineamiento automático.
- Nos proporciona un parámetro adicional que podemos ajustar en nuestros modelos. Distintos atributos a clasificar pueden beneficiarse de tener información de un contexto amplio de la palabra mientras que para otros puede generar ruido que empeore los resultados.
- El chino es un idioma en el que la información morfológica no está representada por lo que es complicado ajustar los modelos ya que para dos entradas idénticas podemos tener dos resultados distintos.

Una vez separado el texto en ventanas hemos de representarlas de forma numérica para utilizarlas posteriormente. Para ello ordenamos las palabras de nuestro corpus por número de apariciones y seleccionamos las  $m$  primeras palabras como nuestro vocabulario. Donde  $m$  es un parámetro y tiene un impacto notable sobre la clasificación. Finalmente a cada palabra la representamos con su índice en el vocabulario.

Este método nos permite entrenar el modelo con valores desconocidos, tal y como los encontraría al clasificar un texto diferente, manteniendo aquellas más comunes y con más impacto en los resultados.

### 3.2 Estudio de características

Dedicamos esta subsección a explicar qué otras características o información de entrada al clasificador estudiamos como alternativas a las ventanas de palabras.

Para poder emplear algoritmos de aprendizaje automático sobre un conjunto de datos, éste ha de estar representado numéricamente. En la tarea que nos ocupa, en la que nos basamos en texto, no es trivial encontrar una representación que preserve la información relevante de la entrada para conseguir un buen resultado de clasificación.

Inicialmente nos planteamos utilizar la información procedente de las palabras del tex-

to original en chino como entrada para los algoritmos. La motivación detrás de esta idea era el hecho de que el texto en chino no tiene ningún tipo de procesado, a diferencia del texto en castellano al que se le ha quitado la información morfológica.

Realizamos distintos experimentos utilizando esta información. En primer lugar se probó utilizar un vocabulario bilingüe en el que cada entrada representa una palabra del texto en castellano con su correspondiente traducción al idioma chino.

Un factor a favor de esta estrategia es que permitía obtener información cuando la palabra en chino era diferente, pero el texto el castellano era igual por estar simplificado. Por ejemplo, para la palabra *NC000[doctor]* podemos encontrar en el texto en chino las palabras *Yīshēng* (doctor) y *Nu yīshēng* (doctora), en pinyin simplificado.

Otra estrategia empleada fue en lugar de crear ventanas de tamaño fijo, utilizar una herramienta externa que nos permita separar los sintagmas de las oraciones en chino. Para ello escogimos la herramienta *Stanford parser* (Chen y Manning, 2014), ya que es ampliamente utilizada y nos resultaba muy conveniente por incluir modelos ya entrenados.

Paralelamente experimentamos con estrategias híbridas entre la presentada en la arquitectura definitiva y las basadas únicamente en información en chino. Realizamos experimentos combinando las ventanas creadas con el texto en chino y el texto en castellano.

También se probó a añadir a las ventanas del texto en castellano información de su correspondiente par en el texto en chino. Añadimos la información sobre pronombres presentes en la oración y la longitud de la palabra en chino. Podemos ver en el ejemplo anterior para la palabra doctor que la flexión de género de la palabra se realiza añadiéndole un carácter.

Todas estas implementaciones fueron finalmente descartadas debido a que dependen de la calidad del fichero de alineamiento generado por la traducción realizada con *Moses*. Analizando este fichero vemos que existen diferencias entre las traducciones correctas y los pares de palabras chino-castellano generados. Esta situación causaba que las entradas del vocabulario no fueran correctas y que no representara la frecuencia real de aparición en el texto.

## 4 Clasificación

En esta sección explicamos los diferentes algoritmos de clasificación que testamos en el contexto de nuestra experimentación y detallamos la arquitectura final de clasificación.

### 4.1 Alternativas

Utilizamos distintos algoritmos lineales como referencia de las modificaciones que realizábamos con las características de entrada explicadas en la sección anterior. El clasificador bayesiano ingenuo resultaba muy conveniente como medida de la mejora que suponían los cambios introducidos por necesitar poco tiempo de entrenamiento y no tener parámetros que ajustar.

A la vez utilizamos SVMs con una función de núcleo (o kernel) lineal para tener una medida de los resultados del momento usando un algoritmo más sofisticado.

Con un proceso más definido y próximo al finalmente empleado comenzamos a realizar experimentos con algoritmos no lineales. Estos algoritmos son los más utilizados en tareas de etiquetado gramatical en los últimos años. Y con ellos ya pretendíamos obtener resultados que se aproximaran a los resultados del estado del arte.

Decidimos utilizar las dos familias más utilizadas para esta tarea, SVMs con funciones de núcleo no lineales y redes neuronales.

Finalmente intentamos abordar la tarea desde un enfoque distinto. Habíamos probado métodos lineales pero en los que generábamos un único modelo. Motivados por probar otra familia de algoritmos muy utilizada en tareas de clasificación, los métodos de *ensembraje*. Y entre ellos el más comúnmente utilizado, *Selvas Aleatorias* (o Random Forest).

### 4.2 Arquitectura propuesta

A continuación, describimos la arquitectura basada en aprendizaje profundo que proponemos en este trabajo. La arquitectura desarrollada es una combinación de las arquitecturas propuestas en diversos artículos. La estructura de la red está fundamentada en la propuesta (Collobert et al., 2011) en el cual se muestran dos arquitecturas distintas. En la primera se utilizan ventanas de texto y en la segunda se utilizan sintagmas y se añade el uso de una red convolucional. La red que presentamos a continuación combina ambas aproximaciones.

A su vez, al utilizar palabras simplificadas la red ha de obtener gran parte de la información a partir del orden de la secuencia de palabras que recibe. Teniendo en cuenta este aspecto y basándonos en los resultados obtenidos en generación de texto utilizando redes recurrentes (Sutskever, Martens, y Hinton, 2011; Graves, 2013) decidimos utilizarlas en nuestra arquitectura. Tal y como se muestra en la Figura 2, la arquitectura concatena las siguientes capas: embedding, convolucional, recurrente, sigmoide y softmax.

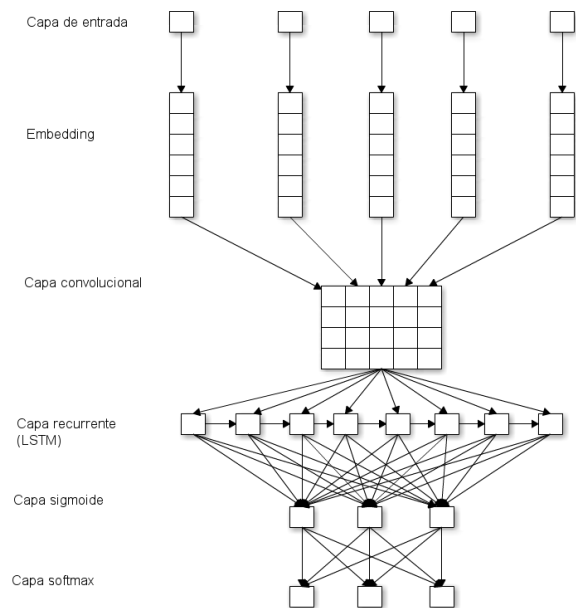


Figura 2: Arquitectura de la red

En primer lugar tenemos la capa de *embedding*, la cual nos permite a partir de la representación discreta de nuestras palabras crear una representación de las mismas en un espacio continuo de mayor dimensionalidad.

Pese a que los datos tras esta capa nos facilitan más información que las ventanas originales podemos mejorar su representación. Para ello utilizamos la capa convolucional. Su cometido es a partir de la salida de la capa anterior reducir el tamaño de los datos mostrando las similitudes entre las palabras de la ventana.

Tras estos primeros pasos ya disponemos de los datos preparados para realizar la clasificación. Nuestra clasificación se basa en el contexto de la palabra dentro de la oración, por lo que decidimos utilizar una red recurrente que nos permita tratar las ventanas como secuencias en el tiempo.

Es decir, en qué orden aparecen las pala-

bras en nuestra ventana marca el efecto que tienen sobre las otras. Podemos ver un ejemplo de ello en la frase *La gata del vecino*, suponiendo que todas las palabras pertenezcan a la misma ventana de texto, tratarlo como una secuencia nos permite representar que el impacto de *gata* sobre *la* sea mayor que el de *vecino*, minimizando el impacto del ruido en los datos.

Pero tratar los datos como secuencia no nos asegura un buen resultado por si mismo, ya que encontramos situaciones en las no deseamos que palabras encontradas anteriormente afecten a las siguientes. Un ejemplo de ello son los signos de puntuación, los cuales marcan un cambio en aquello de lo que estamos hablando como puede ser en el caso de un punto, o elementos sin relación morfológica entre ellos como podemos encontrar entre comas en una enumeración.

Por ello utilizamos en esta tarea una capa *long short-term memory (LSTM)* que nos proporciona además de poder utilizar información de la secuencia de palabras, la capacidad de entrenar cuando se deben olvidar la información de las anteriores.

Tras ella tenemos una capa con activación Sigmoide con tantas unidades como clases pretendemos clasificar y acabar de ajustar los resultados proporcionados por la capa recurrente.

Y finalmente, al ser el objetivo de nuestra red proporcionar la probabilidad de nuestras palabras de pertenecer a cada una de las clases, una capa con Softmax como activación que nos normaliza la salida de forma que la suma de todas las salidas sea 1.

## 5 Postprocesado

Con las probabilidades obtenidas por los modelos podemos generar el texto incluyendo la información morfológica.

Para ello por cada palabra del texto generamos sus etiquetas ordenadas de mayor a menor probabilidad según los resultados obtenidos. El motivo de generar todas sus etiquetas es reducir el impacto en el texto generado de las palabras que los modelos no han clasificado correctamente y que la opción más probable no existe en el idioma.

Un ejemplo de ello son palabras invariantes que el modelo clasifica como masculinas o femeninas y que de solo generar esa opción sólo obtendríamos el lema como resultado.

Para realizar este proceso utilizamos los

diccionarios proporcionados por *Freeling*, así como sus reglas de afijos para obtener a partir de la etiqueta generada y su lema la forma final que necesitamos. Con este proceso hemos añadido el conocimiento morfológico a nuestro modelo pero en orden de obtener mejores resultados diversas reglas han de ser añadidas:

- Las conjunciones *y* y *o* antes de vocal. Debemos controlar una vez tenemos nuestro texto generado que si encontramos una conjunción y antes de *i* o *hi* debemos sustituirla por *e*. De la misma forma la conjunción *o* antes de *o* u *ho* debe ser substituida por *u*.
- Los verbos que tengan un pronombre como sufijo. Cuando la forma conjugada de un verbo es llana y le añadimos un sufijo, pasa a ser esdrújula y por lo tanto debemos acentuarla siempre, pese a que originalmente no.

## 6 Experimentación

En esta sección comentamos los datos y los parámetros que usamos en el contexto experimental.

### 6.1 Datos

El corpus utilizado consiste en fragmentos extraídos de discursos de la ONU (Rafalovitch y Dale, 2009). Para cada uno de los fragmentos disponemos de sus correspondientes traducciones en chino y castellano. Las estadísticas están presentadas en la Tabla 2.

	Líneas	Palabras
Entrenamiento	58.688	2297656
Desarrollo	990	43489
Validación	1.010	44306

Tabla 2: Tamaño de los conjuntos del corpus

### 6.2 Optimización

Los algoritmos de aprendizaje automático generalmente presentan distintos parámetros que es necesario ajustar a nuestros datos para conseguir un resultado óptimo. La Tabla 3 muestra los valores más relevantes de la arquitectura de clasificación. Estos parámetros han demostrado ser los óptimos después de una experimentación que se puede ver en nuestro estudio previo de la arquitectura (Escolano y Costa-jussà, 2017).

	Número	Género
Tamaño de ventana	9	7
Tamaño de vocabulario	9000	7000
Tamaño de los <i>filtros</i>	5	5
Tamaño del <i>embedding</i>	128	128
Unidades de la capa recurrente	70	70

Tabla 3: Valores escogidos para los parámetros de la red

## 7 Evaluación

En esta sección presentamos los resultados obtenidos en clasificación y también las mejoras obtenidas en el sistema final de traducción automática.

### 7.1 Resultados de Clasificación

La Tabla 4 muestra los resultados de clasificación para los distintos algoritmos testeados. En el caso de SVMs y *Selvas Aleatorias* se utilizó *10K* de crossvalidación. Respecto a la representación de los datos todos los sistemas fueron entrenados con el mismo tratamiento de los datos, ventanas de 7 palabras y vocabulario de 7000 palabras.

Algoritmo	% Género	% Número
Bayesiano ingenuo	53,5	61,3
SVM lineal	71,7	68,1
SVM cuadrático	81,3	77,8
SVM sigmoid	87,4	83,1
Selvas Aleatorias	91,8	81,6
convNet+LSTM	<b>98,4</b>	<b>93,7</b>
convNet-GRU	95,1	91,4

Tabla 4: Resultados obtenidos por los diferentes algoritmos de clasificación. Los mejores resultados aparecen en negrita

Los datos de la tabla fueron obtenidos utilizando el preproceso explicado en apartados anteriores salvo en el caso de *Selvas Aleatorias* con *one hot encoding* donde en lugar de realizar el *embedding* de los datos se entrenó el sistema con la representación *one hot encoding*.

Respecto a los resultados obtenidos por un clasificador bayesiano ingenuo hemos de aclarar que son muy bajos debido a que al analizar su predicción, su salida para toda entrada era la clase mayoritaria en el conjunto de entrenamiento (*invariable* para género y *singular* para número).

Vemos como los resultados obtenidos por los clasificadores basados en redes neuronales superan a los obtenidos por el resto por lo que son los escogidos para la arquitectura final del sistema. Observamos también que dentro de ellos, los resultados de la red *LSTM* son lige-

ramente mejores que los obtenidos utilizando GRU.

## 7.2 Resultados de Traducción

Usando *Moses* concatenado con nuestro clasificador y el postprocesado, podemos generar el texto traducido completo y compararlo con el sistema de referencia.

Para comparar nuestros resultados utilizaremos METEOR. La Tabla 5 muestra los oráculos y los resultados de traducción. Vemos que tanto la simplificación en género como en número por separado aportan mejoras en traducción. La combinación de ambas simplificaciones y su recuperación consiguen mejorar el METEOR en casi 0,2 % absoluto.

Modelo	Oráculo	Meteor
Sistema de referencia	-	55,29
Simplificado Número	55,60	55,35
Simplificado Género	55,45	55,39
Simplificado	56,81	<b>55,48</b>

Tabla 5: Resultados en traducción en términos de METEOR. El mejor resultado aparece en negrita

## 8 Conclusiones

La principal contribución de este artículo ha sido el desarrollo de un arquitectura específica basada en aprendizaje profundo para la generación de conocimiento morfológico que presenta una mejora cuando se integra en un sistema de traducción basado en segmentos. Los resultados en términos de METEOR mejoran el sistema de referencia.

Asimismo, otra contribución relevante de este artículo es el desarrollo de un clasificador que mejora los resultados del estado del arte en tareas de etiquetado morfológico.

La dificultad de esta tarea de etiquetación morfológica respecto a tareas similares, como el etiquetado gramatical, radica en que las palabras que usamos no tienen ninguna información sobre la flexión que queremos generar. Así pues, en nuestro caso, tenemos que aprender a partir del contexto.

Como trabajo futuro, queremos precisamente aplicar nuestra arquitectura a la tarea de etiquetado gramatical. De esta manera, podremos simplificar las palabras al nivel de lemas y después generar la traducción final con nuestro etiquetador gramatical.

## Agradecimientos

Este trabajo ha sido financiado mediante el programa *Ramón y Cajal* y el contrato TEC2015-69266-P (MINECO/FEDER, UE) por el Ministerio de Economía y Competitividad y el Fondo Europeo de Desarrollo Regional.

## Bibliografía

- Chen, D. y C. Manning. 2014. A fast and accurate dependency parser using neural networks. En *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, páginas 740–750, Doha, Qatar, October. Association for Computational Linguistics.
- Clifton, A. y A. Sarkar. 2011. Combining morpheme-based machine translation with post-processing morpheme prediction. En *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT '11*, páginas 32–42, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Collobert, R., J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, y P. Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, Noviembre.
- Costa-jussà, M. R. 2015. Ongoing study for enhancing chinese-spanish translation with morphology strategies. En *Proceedings of the Fourth Workshop on Hybrid Approaches to Translation (HyTra)*, páginas 56–60, Beijing, July. Association for Computational Linguistics.
- Denkowski, M. y A. Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. En *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.
- Escolano, C. y M. R. Costa-jussà. 2017. Spanish Morphology Generation with Deep Learning. *International Journal of Computational Linguistics and Applications (IJCLA)*. In press.
- Formiga, L., M. R. Costa-jussà, J. B. Mariño, J. A. R. Fonollosa, A. Barrón-Cedeño,



- y L. Màrquez. 2013. The TALP-UPC phrase-based translation systems for WMT13: System combination with morphology generation, domain adaptation and corpus filtering. En *Proceedings of the Eighth Workshop on Statistical Machine Translation*, páginas 134–140, Sofia, Bulgaria, August.
- Graves, A. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.
- K., Philipp, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, y E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. En *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, páginas 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.
- Kholy, A. E. y N. Habash. 2012. Rich morphology generation using statistical machine translation. En *Proceedings of the Seventh International Natural Language Generation Conference, INLG '12*, páginas 90–94, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Koehn, P., F. J. Och, y D. Marcu. 2003. Statistical phrase-based translation. En *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, NAACL '03*, páginas 48–54, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Padró, L. y E. Stanilovsky. 2012. Free-ling 3.0: Towards wider multilinguality. En *Proceedings of the Language Resources and Evaluation Conference (LREC 2012)*, Istanbul, Turkey. ELRA.
- Rafalovitch, A. y R. Dale. 2009. United nations general assembly resolutions: A six-language parallel corpus. En *Proceedings of the MT Summit*, volumen 12, páginas 292–299.
- Sutskever, I., J. Martens, y G. E. Hinton. 2011. Generating text with recurrent neural networks. En *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, páginas 1017–1024.
- Toutanova, K., H. Suzuki, y A. Ruopp. 2008. Applying morphology generation models to machine translation. En *Proceedings of the conference of the Association for Computational Linguistics and Human Language Technology (ACL-HLT)*, páginas 514–522, Columbus, Ohio.