Traducción automática neural: cómo funciona y qué se puede esperar de ella.

Mikel L. Forcada[1,2]

[1]Departament de Llenguatges i Sistemes Informàtics,
Universitat d'Alacant, E-03071 Alacant

[2]Prompsit Language Engineering, S.L.,
Edifici Quòrum III, Av. Universitat, s/n, E-03202 Elx

Torre Juana OST / Friday Tech Fever
8 noviembre 2019

Índice

Índice

Traducción automática

Traducción automática

- El procesamiento,
- mediante un ordenador que utiliza software adecuado,
- de un texto escrito en la *lengua origen* (LO)
- que produce otro texto en la *lengua meta* (LM)
- el cual puede a veces ser utilizado como *traducción en bruto*.

Usos de la traducción automática

Dos grandes grupos de aplicaciones:

Asimilación: cuando se usa *tal cual* para comprender texto en otra lengua. Debe ser rápida e inteligible aunque no sea gramatical. Puede que no se guarde.

Diseminación: cuando se usa para producir un texto que se publicará. Debe ser adecuada como está (es raro) o fácil de corregir (post-editar) para hacerla publicable.

Tecnologías de traducción automática

Hay dos grupos principales de tecnologías de traducción automática (TA):

- **La TA basada en reglas**, y
- **La TA basada en corpus**

Traducción automática basada en reglas

Traducción automática basada en reglas (RBMT) (Lucy Software, ProMT, Apertium...):

- **Personas expertas en traducción** escriben diccionarios de traducción y reglas que transforman estructuras de LO a estructuras en la LM
- **Personas expertas en informática** escriben *motores* que usan esos diccionarios y aplican esas reglas al texto de entrada
- **La salida** es estable pero *mecánica* y carece de *naturalidad*
- Tiene problemas con la **ambigüedad** (léxica y sintáctica).
- **Personalización**: edición de los diccionarios y de las reglas

Traducción automática basada en corpus /1

La TA **basada en corpus** aprende a traducir a partir de un corpus de centenares de miles o millones de oraciones traducidas.[1]

- **Personas expertas en traducción** proporcionan las traducciones.
- **Personas expertas en informática** escriben programas que *aprenden* de estas traducciones.
- **Salida:** puede ser *engañosamente natural* (*infiel*).
- **Personalización:** Entrenando con textos relacionados.

[1] Esto puede no estar disponible para algunas lenguas o tipos de texto

Traducción automática basada en corpus /2

Aproximaciones principales a la TA basada en corpus :

- *Traducción automática* ***estadística*** (2000–2015)
  - Usa modelos **probabilísticos** que se estiman contando eventos en los corpus bilingües usados para entrenarlos.
- *Traducción automática* ***neural*** (2015–).
  - Basado en **redes neurales**[2] **artificiales** inspiradas en cómo el cerebro humano aprende y generaliza.

[2] o *neuronales*

Índice

TA neural: la nueva TA basada en corpus

La **traducción automática neural** o basada en *aprendizaje hondo* (*deep learning*), alternativa reciente a la TA estadística:

- Está basada en corpus
  - Se suele decir que necesita más datos y más limpios
- Primeras ideas en los noventa,[3] abandonadas debido a la falta de hardware suficientemente potente
- Retomada alrededor de 2013
- Se ofrece comercialmente en 2016 (Google Translate)
- Competitivo con la TA estadística en muchas aplicaciones

[3] Chalmers, *Conn. Sci. 2*(1990)53; Chrisman, *Conn. Sci.* 3(1991)345; Castaño & Casacuberta, EuroSpeech 1997; Forcada & Ñeco, ICANN 1997

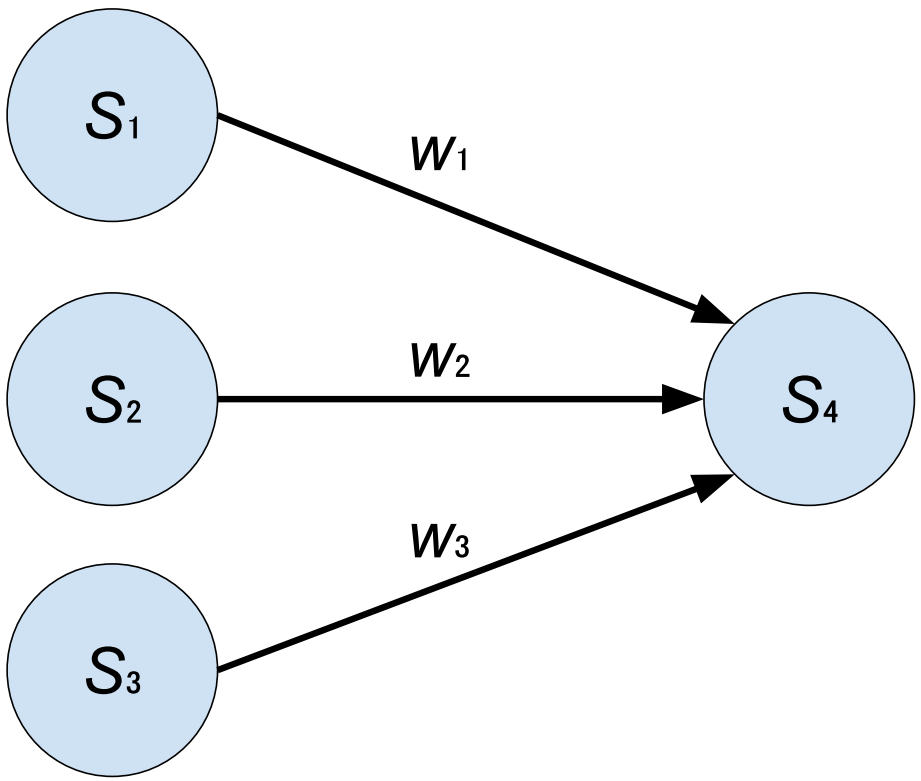
Neuronas artificiales /1

¿Por qué se llama *neural*?

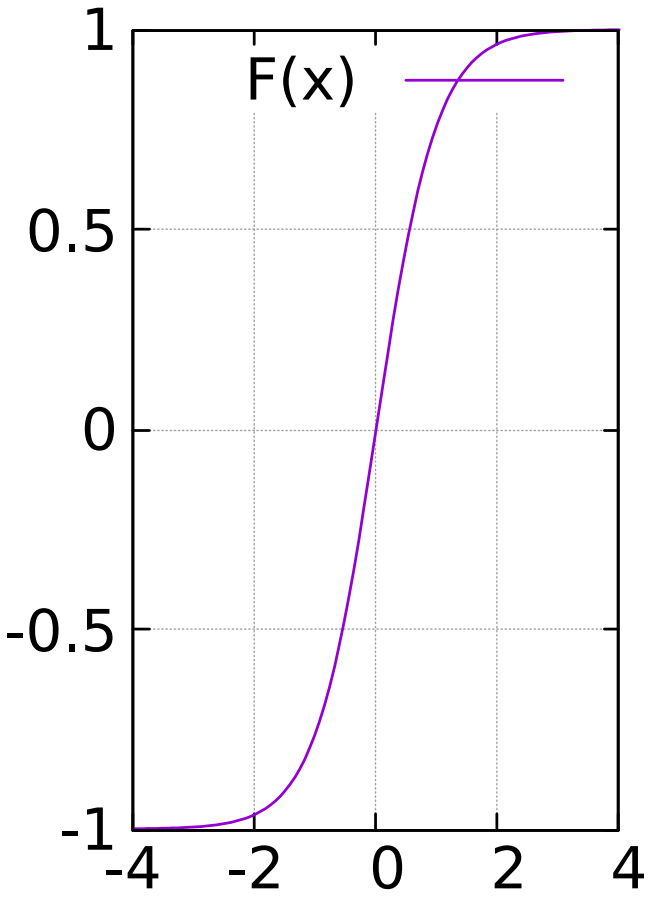
- La realiza software que *simula* grandes redes de *neuronas artificiales* extremadamente simplificadas.
- Su *activación* (excitación) depende de la activación de otras neuronas y del *peso* de sus conexiones.
- El signo y la magnitud de los pesos determina el comportamiento de la red:
  - Neuronas conectadas vía pesos positivos tienden a excitarse o inhibirse simultáneamente.
  - Neuronas conectadas vía pesos negativos tienden a estar en estados opuestos.
  - El efecto aumenta con la magnitud del peso.
- El entrenamiento fija los pesos a los valores necesarios para asegurar un comportamiento concreto.

Neuronas artificiales/2

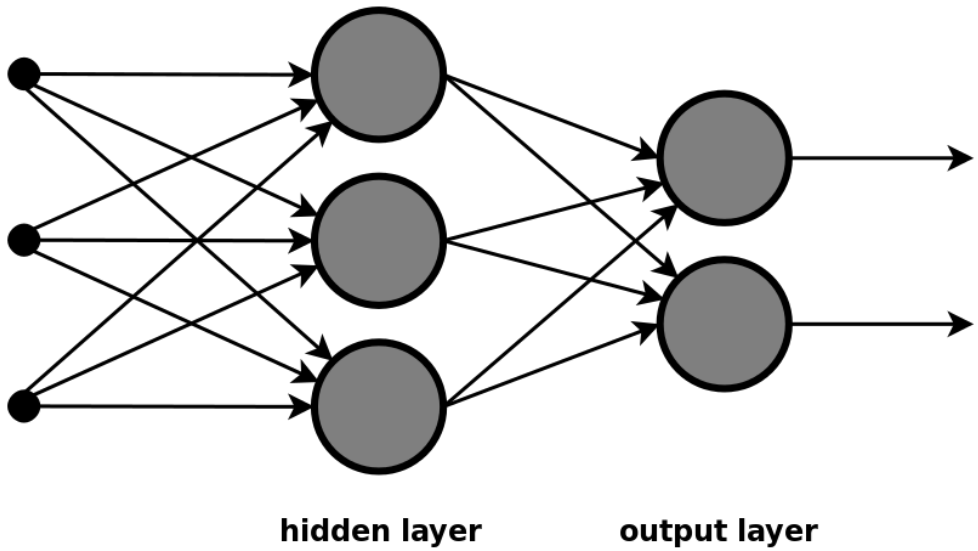
Las neuronas, ¿cómo concilian los estímulos que reciben y reaccionan a ellos?



$$S_4 = F\,(w_1 \times S_1 + w_2 \times S_2 + w_3 \times S_3)$$



Redes neurales



Una red neural con 3 entradas, 3 neuronas en una capa oculta, y dos neuronas de salida.
Se dice *aprendizaje hondo* cuando la información se procesa utilizando *muchas capas ocultas* en cascada.

Representaciones /1

- Los valores de activación de *grupos concretos de neuronas* (normalmente aquellas en una capa) forman *representaciones* (también llamadas *embeddings*) de la información que procesan.
- Por ejemplo, el *vector*

$$(0.35, 0.28, -0.15, 0.76, \ldots, 0.88)$$

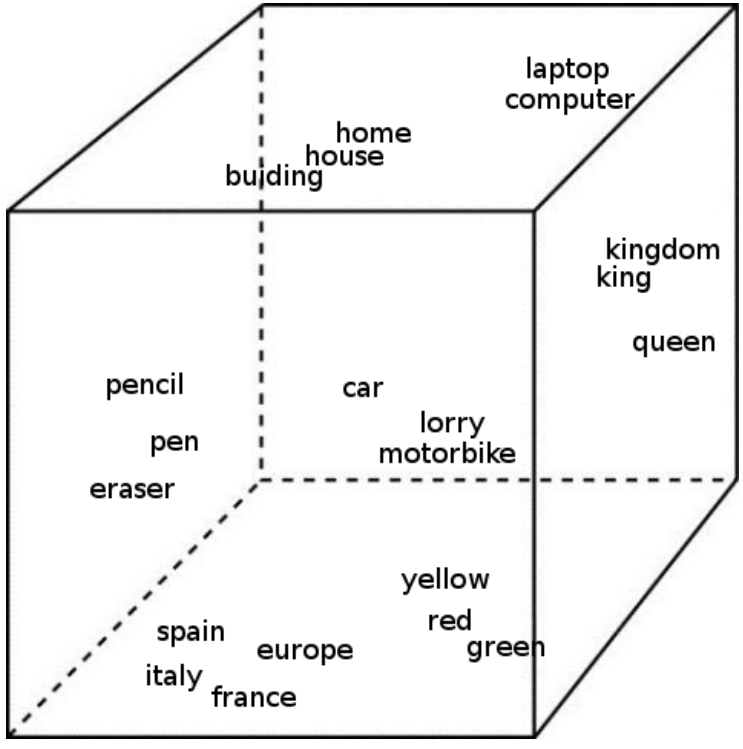
podría ser la representación "$e(\text{estudio})$" de la palabra *estudio*, y el *vector*

$$(0.93, -0.78, 0.22, 0.31, \ldots, -0.71)$$

la representación "$e(\text{gato})$" de la palabra *gato*.

Representaciones /2

Imaginémonos representaciones léxicas con solo tres neuronas:



Las palabras con significados similares se encuentran cerca unas de otras.

Representaciones /3

- Incluso puede realizarse aritmética semántica con representaciones (añadiendo y restando valores de activación neurona a neurona):

$$e(\text{rey}) - e(\text{hombre}) + e(\text{mujer}) \simeq e(\text{reina})$$

TA neural: la arquitectura codificador–descodificador

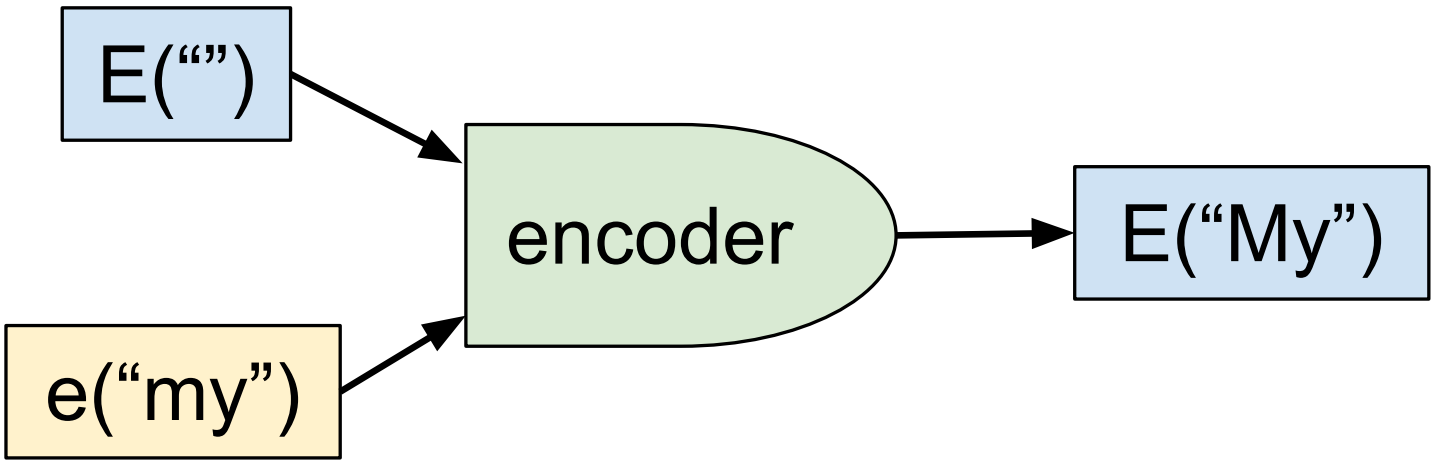
Una arquitectura posible de TA neural es el *codificador–descodificador* (*encoder–decoder*):[4]

- El *codificador* es una red neural que lee, una por una, representaciones de las palabras de la frase original y construye *recursivamente* una representación de la frase; después,
- el *descodificador* es otra red neural que predice, una por una, las palabras de la oración meta:
  - Cada unidad de salida computa la probabilidad de cada posible palabra meta.
  - Se selecciona la palabra más probable.
  - Funciona de modo parecido al teclado de nuestros *smartphones*.

[4] Extensiones a la misma como la *atención* y otras arquitecturas como la *transformer* son ahora también muy comunes.

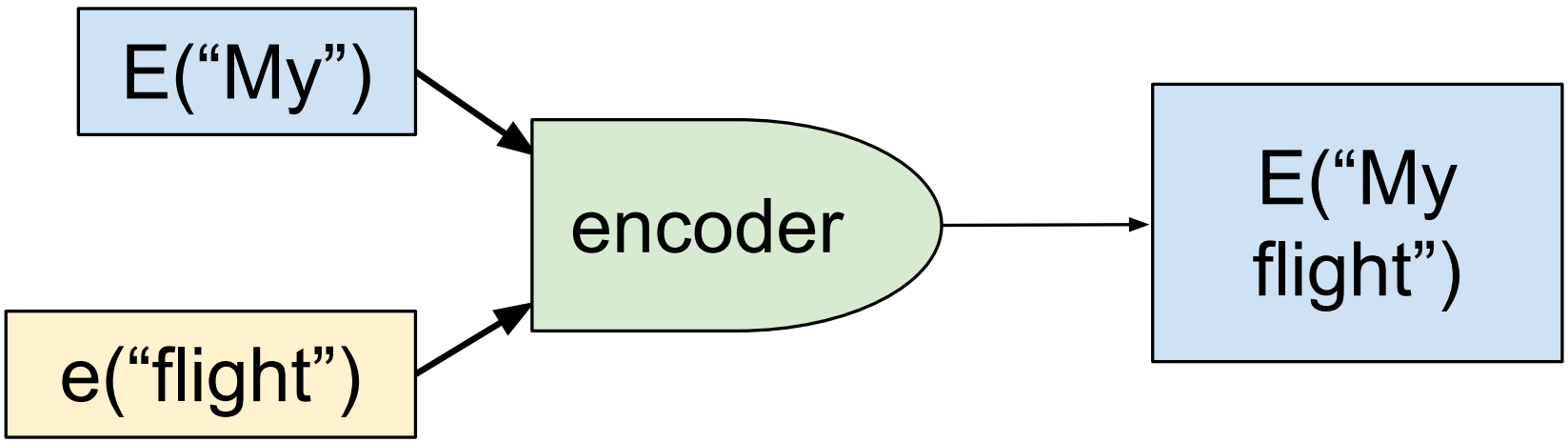
Codificación

Entrada: “**My** flight is delayed .”



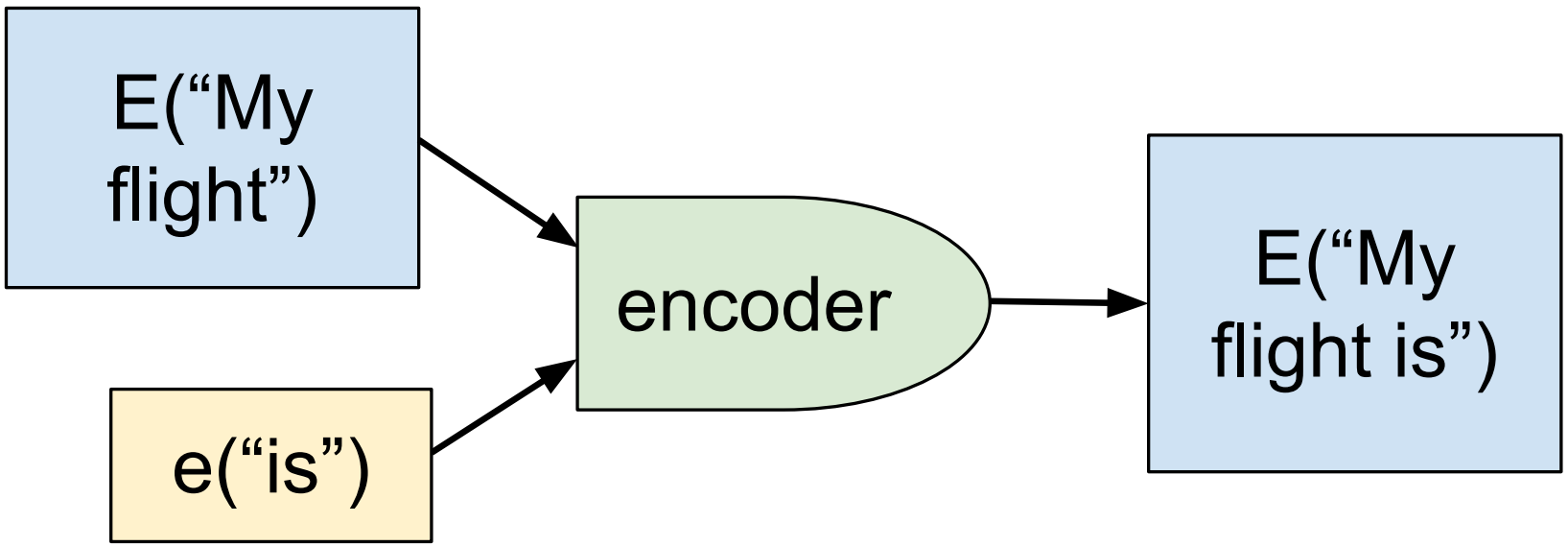
Codificación

Entrada: “**My flight** is delayed .”



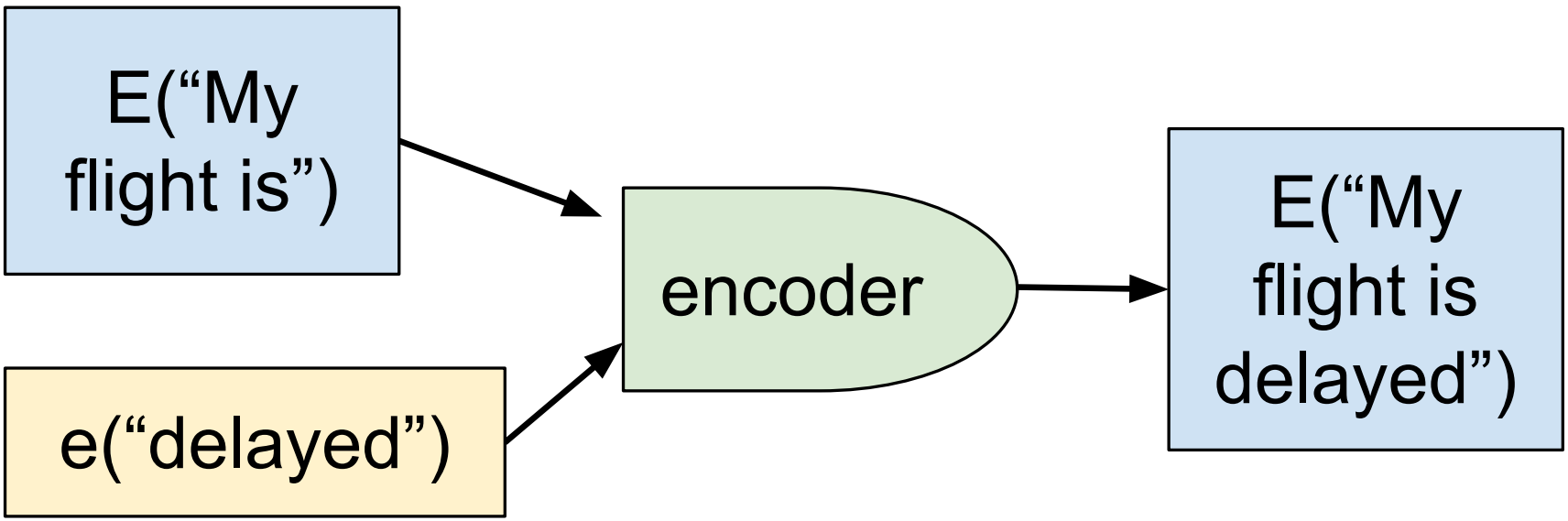
Codificación

Entrada: “**My flight is** delayed .”



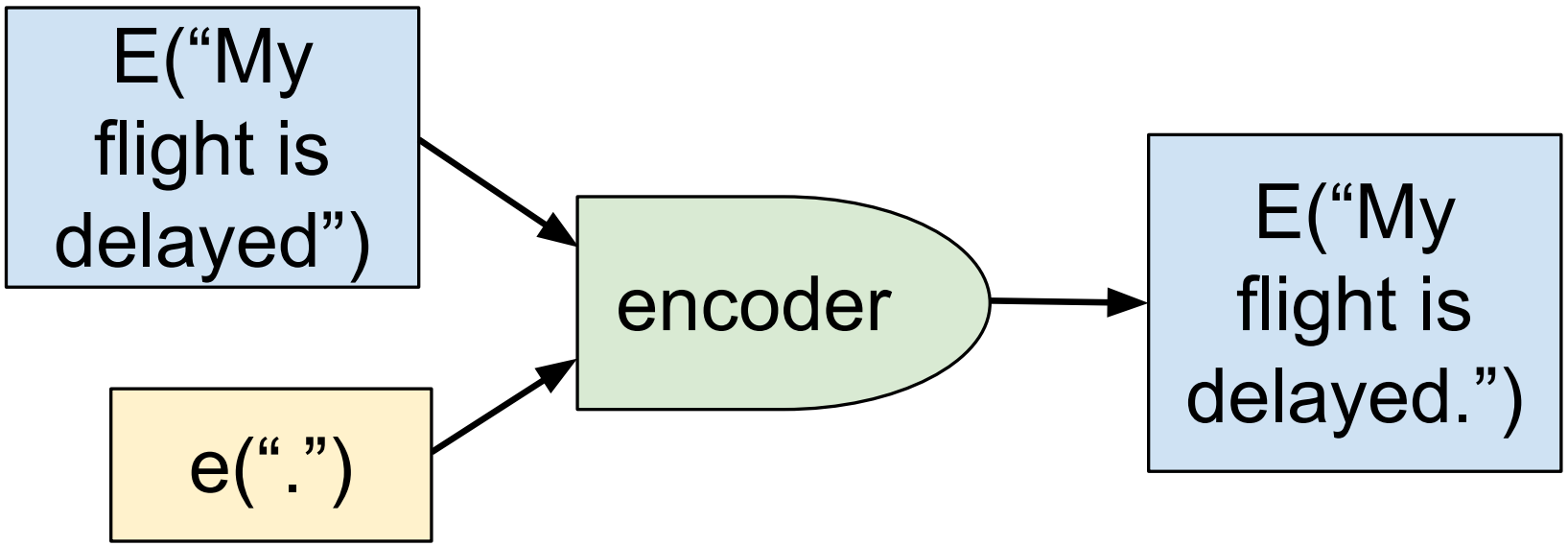
Codificación

Entrada: “**My flight is delayed .**”



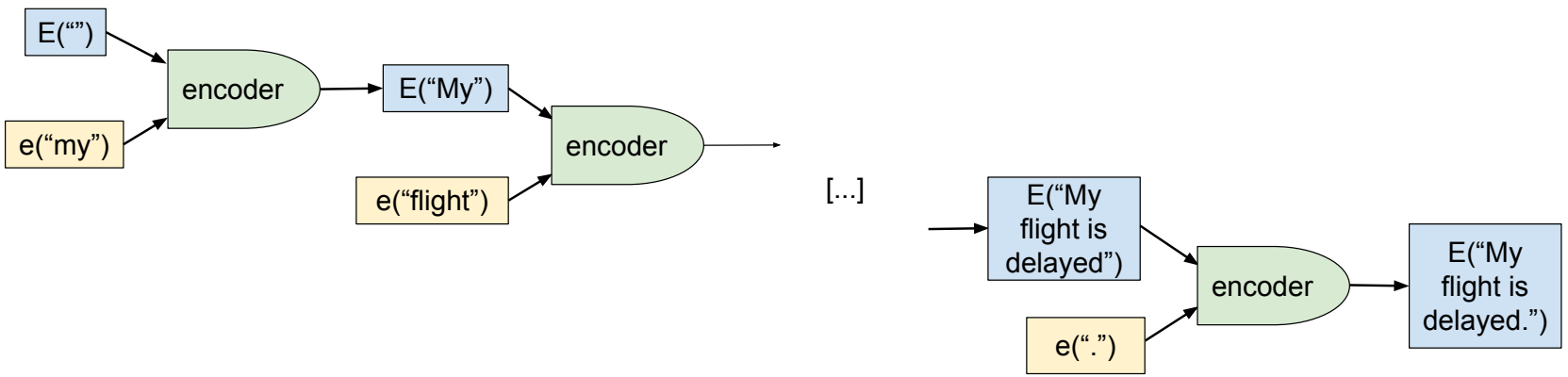
Codificación

Entrada: “**My flight is delayed .**”



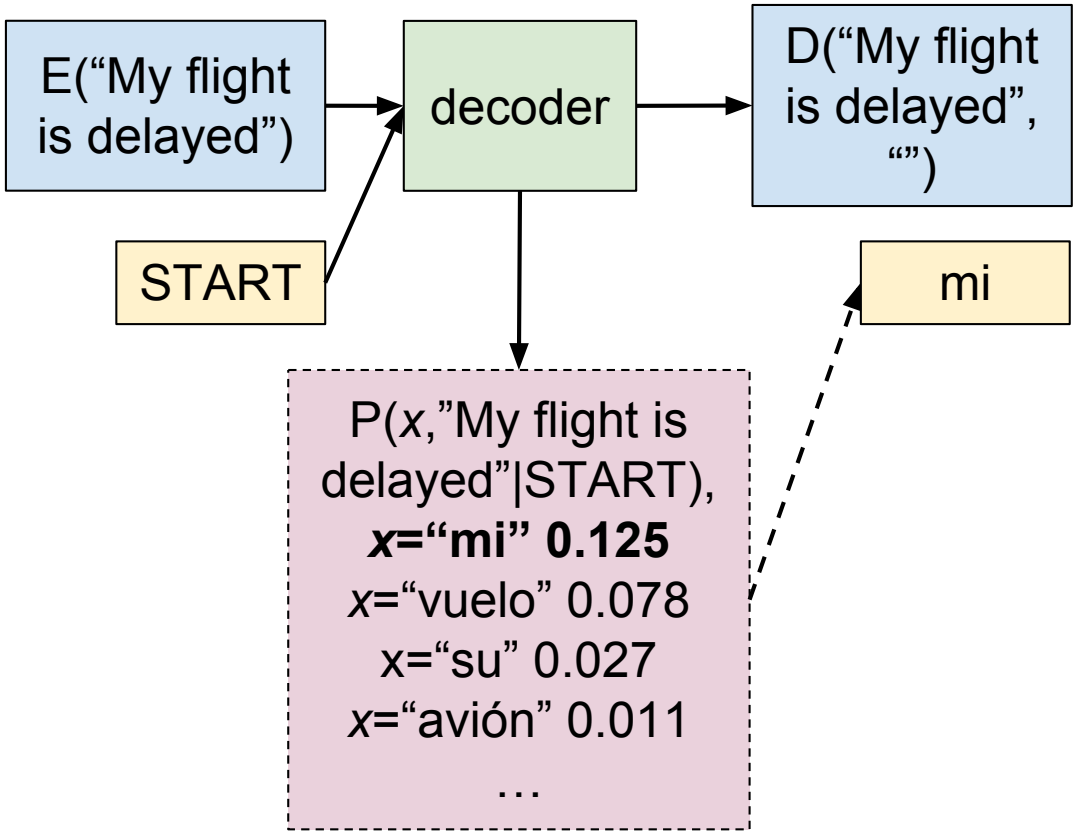
Codificación

Codificación de la oración “Mi vuelo está retrasado .” a partir de las representaciones de sus palabras.



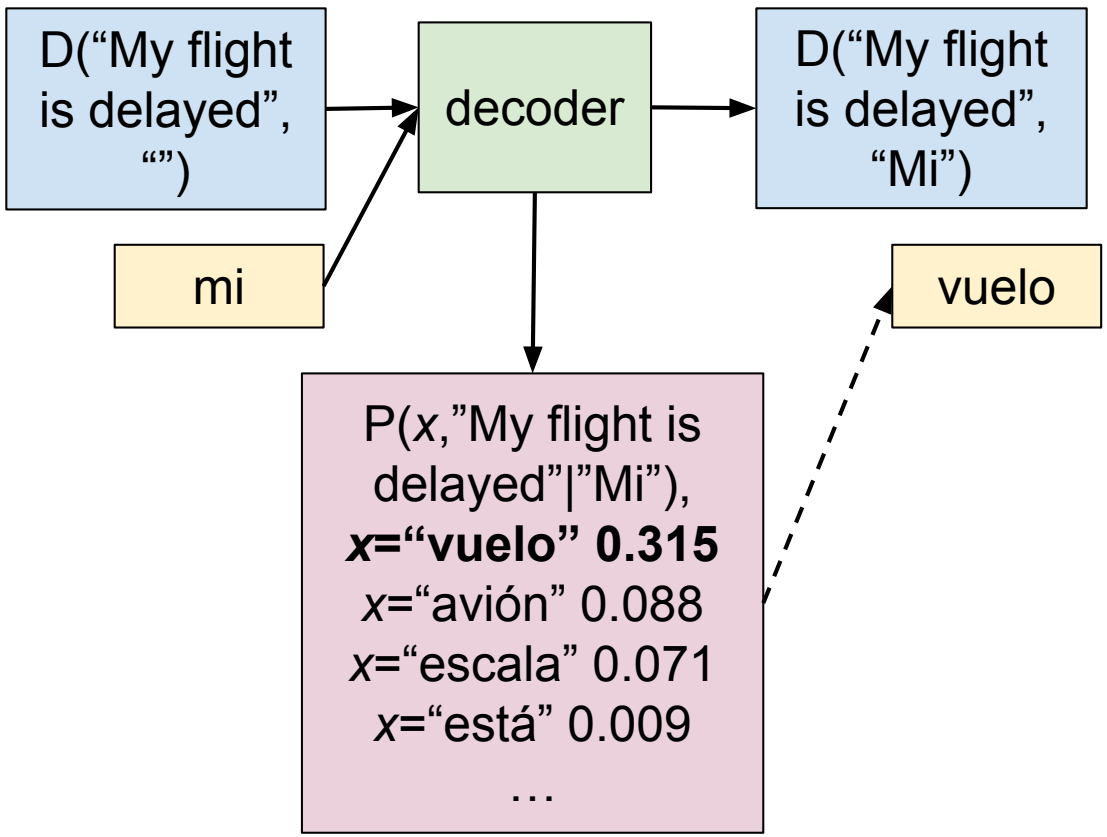
Descodificación

Descodificando "“My flight is delayed .” → “**Mi** vuelo está retrasado .”



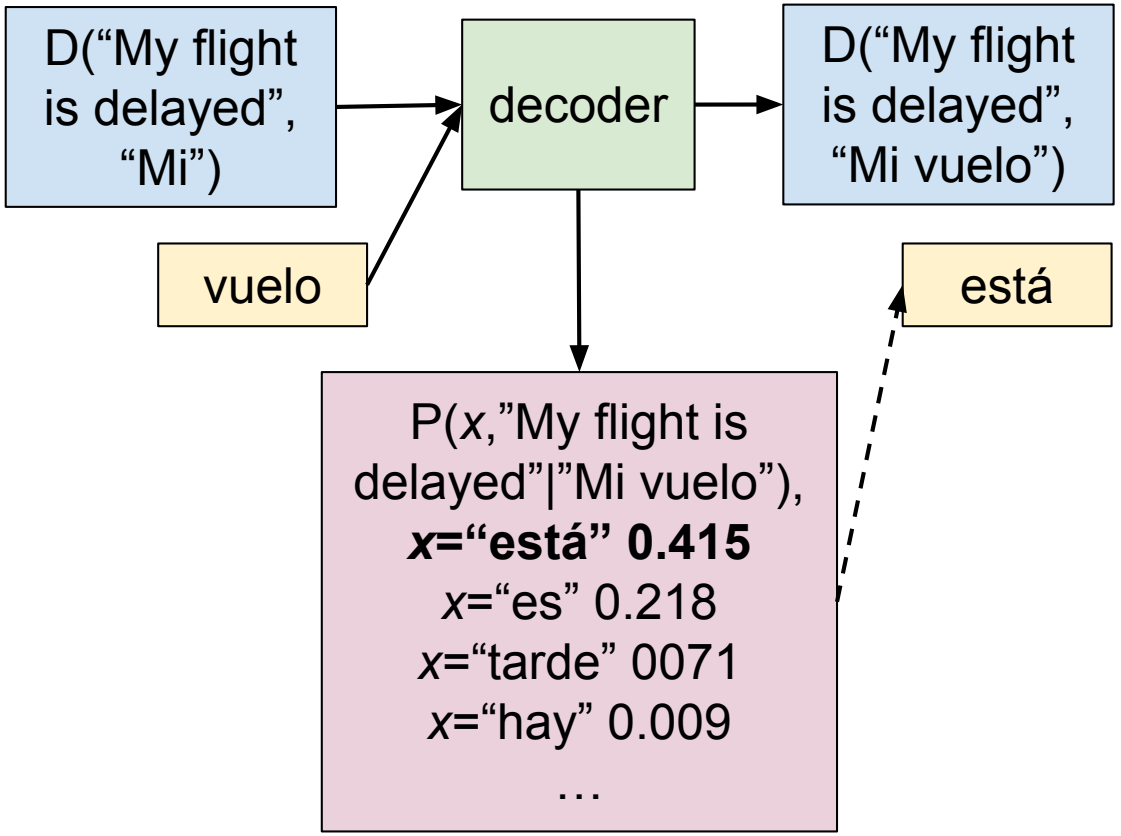
Descodificación

Descodificando “My flight is delayed .” → “**Mi vuelo** está retrasado .”



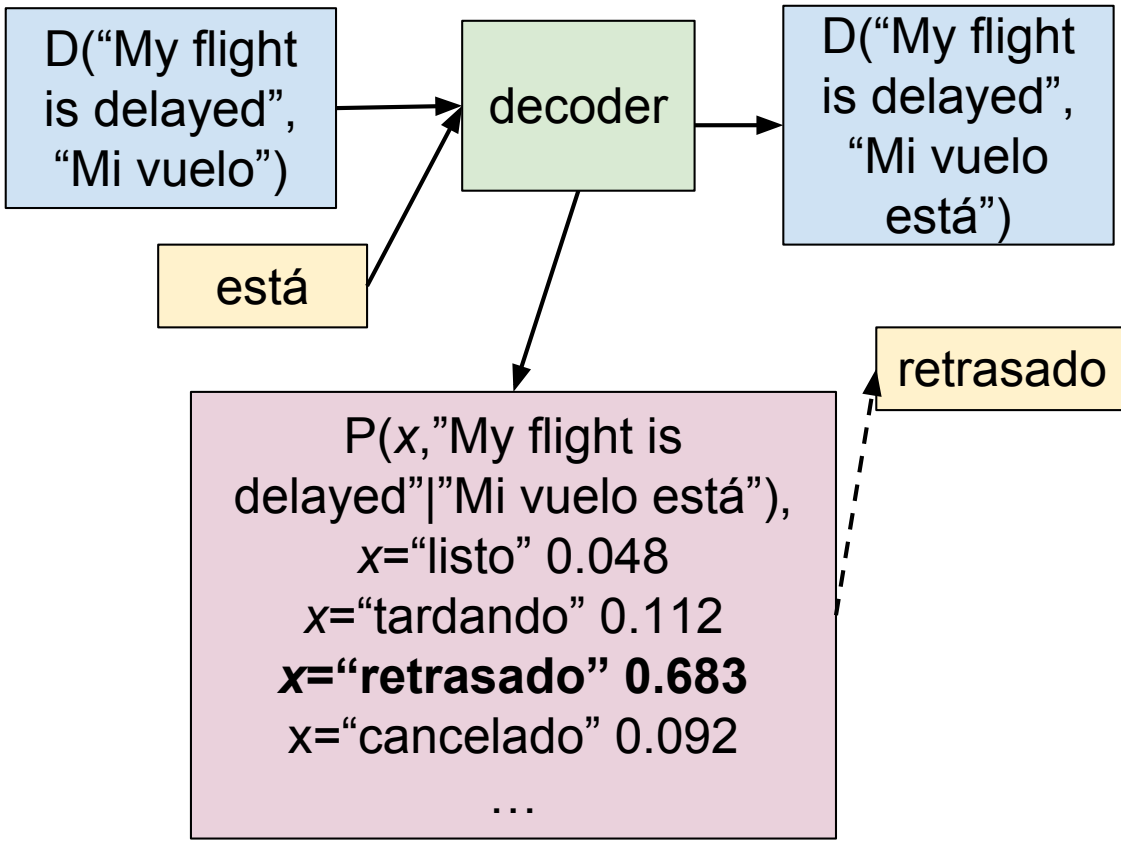
Descodificación

Descodificando “My flight is delayed .” → “**Mi vuelo está** retrasado .”



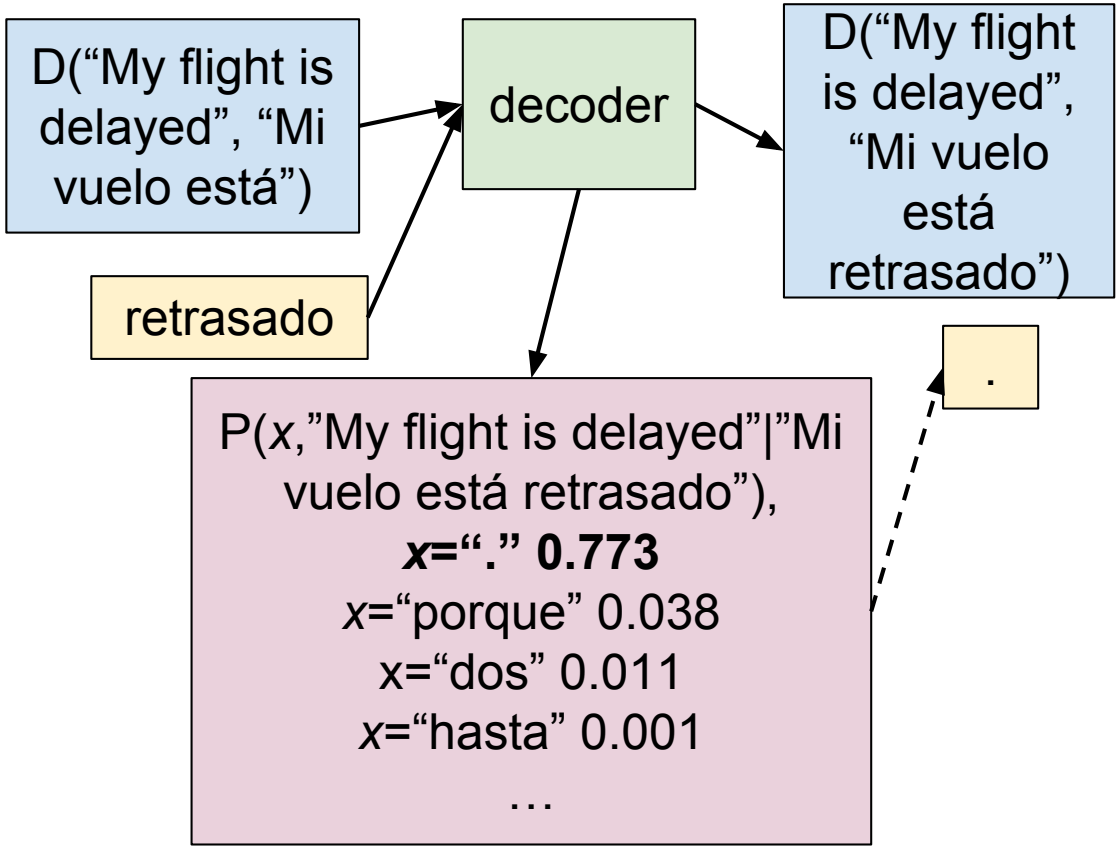
Descodificación

Descodificando “My flight is delayed .” → “**Mi vuelo está retrasado** .”



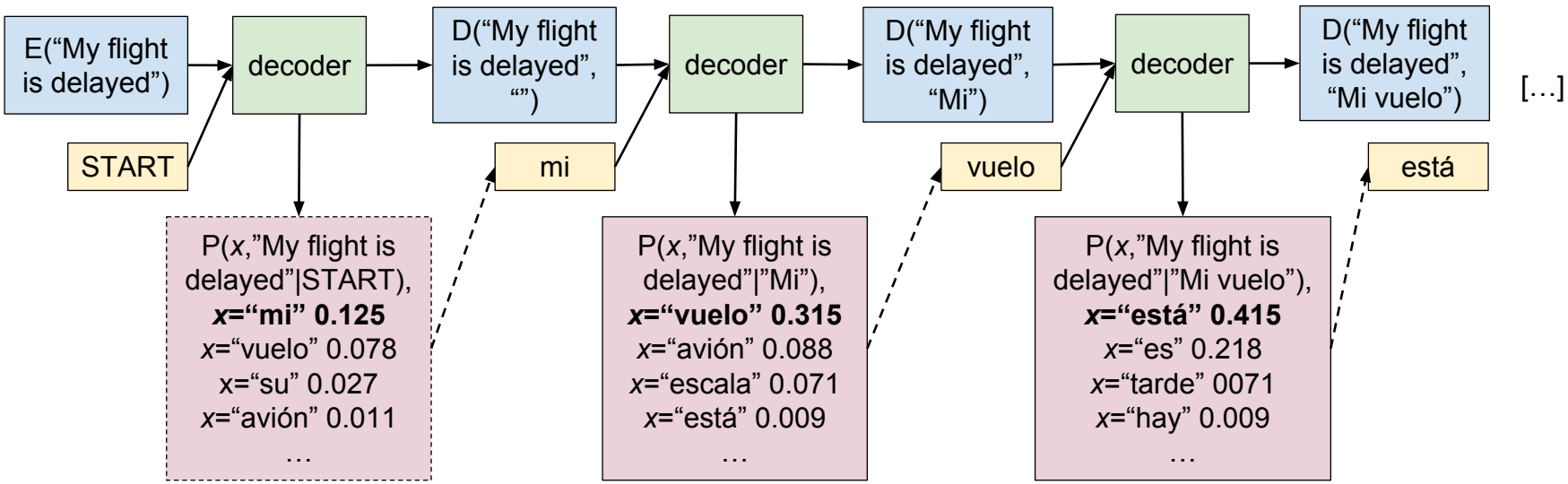
Descodificación

Descodificando “My flight is delayed .” → “**Mi vuelo está retrasado .**”



Descodificación

Descodificación de la traducción de “My flight is delayed .”: “Mi vuelo está retrasado .”.



Índice

Corpus de entrenamiento

Para entrenar un sistema de TA neural necesitamos tres corpus paralelos *disjuntos*:

- Un *gran* **conjunto de entrenamiento**, idealmente de millones de pares de oraciones y representativo de la tarea (puede no estar disponible).
  - Ejemplos que se enseñan a la red neural para entrenarla.
- Un *pequeño* **conjunto de desarrollo** de 1,000–3,000 pares de frases, representativas de la tarea.
  - Los ejemplos de este conjunto reservado sirven para determinar cuándo parar el entrenamiento de modo que se evita el *sobreajuste* al conjunto de entrenamiento y no se daña la generalización.
- Un *pequeño* **conjunto de test** de 1,000–3,000 pares de frases, representativos de la tarea.
  - Los ejemplos de este conjunto reservado dan una idea del rendimiento del sistema.

Preparación del corpus

Para reducir el tamaño del vocabulario efectivo (aprendizaje más fácil), las frases como

You know Chinese, Dr. Walker, don't you?

se *tokenizan* (puntuación y todo) en

You know Chinese , Dr. Walker, don 't you ?

y después se *normalizan* las mayúsculas (*truecasing*)

you know Chinese , Dr. Walker, don 't you ?

o incluso se dividen a en *unidades sub-palabra*:

you know Chin@@ ese , Dr. Walk@@ er, don 't you ?

El proceso de entrenamiento

- Los ejemplos de entrenamiento se presentan repetidamente al sistema.
- Más o menos cada 100 ejemplos, se incrementan o reducen ligeramente los pesos para acercar las salidas de la red a sus valores deseados...
- ...de modo que la probabilidad predicha para las palabras meta indicadas en el conjunto de entrenamiento.
- Los ejemplos se enseñan decenas o centenares de veces → las redes neurales son muy *burras*!
- El entrenamiento para cuando el rendimiento (probabilidad, semejanza) de los ejemplos en el conjunto de desarrollo se degrada.

Cálculos caros

Las TA neural tiene necesita hardware más potente, especializado y caro:

- Las CPU normales de nuestros portátiles o sobremesas son demasiado lentas.
- El entrenamiento neural implica muchas operaciones con vectores, matrices o tensores...
- ...que las GPU (unidades de procesamiento gráfico) realizan de manera más efectiva.[5]
- Usando GPU se puede acelerar el entrenamiento 100 veces o más.

[5] Una GPU típica cuesta unos 2,000 dólares

Índice

Nueva tecnología, nuevo comportamiento/1

La TA neural . . .

- . . . requiere hardware especializado y potente ←**Problema #1**
- . . . necesita muchos datos bilingües ←**Problema #2**
  - que no están normalmente al alcance del traductor medio o de pequeñas compañías de traducción
  - se puede recurrrir a terceros para que nos entrenen y ejecuten el sistema de TA neural:
  - De hecho, esto es un modelo de negocio en la industria de traducción
  - Pueden *condimentar* su enorme corpus de *caldo de fondo* con nuestras pocas traducciones anteriores para construirnos un sistema.

Nueva tecnología, nuevo comportamiento /2

La TA neural...

- ...trabaja con representaciones de la frase completa: es difícil saber de qué palabra original viene cada palabra meta.
  - Falta de transparencia ←**Problema #3**
  - Imaginen la *post-edición* (corrección por parte de un traductor profesional) de una oración larga.
    - (comprobando todas y cada una de las palabras)
- ...produce traducciones engañosamente naturales con errores: ←**Problema #4**
  - No son raras las omisiones, inserciones y "alucinaciones" perfectamente gramaticales.

Nueva tecnología, nuevo comportamiento /3

La TA neural...

- ...produce errores de raíz semántica: si no se ha visto una palabra durante el entrenamiento, se substituye... ←**Problema #5**
  - ...por una palabra similar: *palacio* → *castillo*
  - ...por una paráfrasis: *Michael Jordan* → *el escolta de los Chicago Bulls*;
  - a veces con resultados peligrosos: *Túnez* → *Noruega*.
- ...puede inventarse palabras: *ingenieraje*, *reclutación*, etc. cuando trabajan con unidades sub-palabra (se hace a menudo). ←**Problema #6**

En diseminación, los post-editores tienen que prestar mucha atención al corregir (*carga cognitiva*).

En asimilación, la gente común que usa la traducción tal como está puede recibir información incorrecta.

Comentarios finales

La traducción automática (TA) neural es un nuevo tipo de TA basada en corpus.

- Está desplazando a la TA estadística.
- Se basa en un modelo muy simplificado del sistema nervioso.
- Requiere hardware potente y especializado. ←**Problema #1**
- Requiere grandes corpus. ←**Problema #2**
- Puede producir texto natural con errores que son difíciles de detectar y corregir ←**Problemas #3–#6**

Por tanto, requiere un esfuerzo especial por parte de los post-editores (correctores) y puede despistar a los lectores.

Estas transparencias son libres

Transparencias adicionales

An extension: attention

Encoder–decoders are sometimes augmented with *attention*. The *decoder* learns to “pay (more or less) attention”...

- not only to the last encoded representation E(’My flight is delayed .’)...
- but also to all of the intermediate representations created during encoding,
  - E(’My’), E(’My flight’), E(’My flight is’), E(’My flight is delayed’).
- ...using special *attention* connections.

Probabilities

- Training: adjusting all of the weights in the neural net.
- NMT decoder output: word probabilities in context → sentence probabilities:

$$
\begin{aligned}
P(\text{I love you .}|\text{Je t'aime}) \quad = \quad & p(\text{.} \quad |\text{I love you}, \quad \text{Je t'aime}) \times \\
\times \quad & p(\text{you} \quad |\text{I love}, \quad \text{Je t'aime}) \times \\
\times \quad & p(\text{love} \quad |\text{I}, \quad \text{Je t'aime}) \times \\
\times \quad & p(\text{I} \quad |\text{START}, \quad \text{Je t'aime}).
\end{aligned}
$$

- Objective: *maximize* the likelihood $P(\text{I love you .}|\text{Je t'aime})$ of the reference translation *I love you*.

Gradients and learning

- The training algorithm computes a *gradient* of the probabilities of reference sentences in the training set with respect to each weight w connecting neurons:

$$\text{gradient}(w) = \frac{P(\text{with } w + \Delta w) - P(\text{with } w)}{\Delta w}$$

  - That is, how much the probability varies for a small change Δw in each weight w.
- Then, after showing a number of reference sentences, weights are updated *proportionally* to their effect on their probability → *gradient ascent*

$$\text{new } w = w + (\text{learning rate}) \times \text{gradient}(w)$$

- This is done repeatedly.

Epochs and minibatches

- Examples (sentence pairs) are grouped in *minibatches* of e.g. 128 examples.
- Weights are updated after each *minibatch*.
- An *epoch* completes each time the whole set of examples, e.g. 1,000,000 examples, have been processed.
- It is not uncommon for a training job to require tens or hundreds of epochs.

Stopping

When does one stop?

- Training “too deep” may lead to “memorization” of the training set.
- But we want the network to generalize.
- This is what the *development set* is used for.
  - Every certain number of weight updates, the system automatically evaluates the performance on the sentences of the development set.
  - It compares MT output to the reference outputs and computes a measure such as BLEU.

How expensive?

Neural MT training is *computationally very expensive*.
Example:

- A *very small* encoder–decoder (2 layers of 128 units each)...
- ...with a *small* training set of 260,000 sentences...
- ...using a *small vocabulary* of 10,000 byte-pair encoding operations ...
- ...between French and Spanish (easy language pair)...
- ...takes about 1 week on a 4-core, 3.2 GHz desktop ...
- ...to reach a BLEU score of around 25% (barely starts to be posteditable).

The BLEU score

What is BLEU?

- The most famous automatic evaluation measure is called BLEU, but there are many others.
- BLEU counts which fraction of the 1-word, 2-word,. . . *n*-word sequences in the output match the reference translation.
- These fractions are grouped in a single quantity.
- The result is a number between 0 and 1, or between 0% and 100%.
- Correlation with measurements of translation usefulness is still an open question.
- A lot of MT research is still BLEU-driven and makes little contact with real applications of MT.