Redes Neuronales para Lenguaje Natural

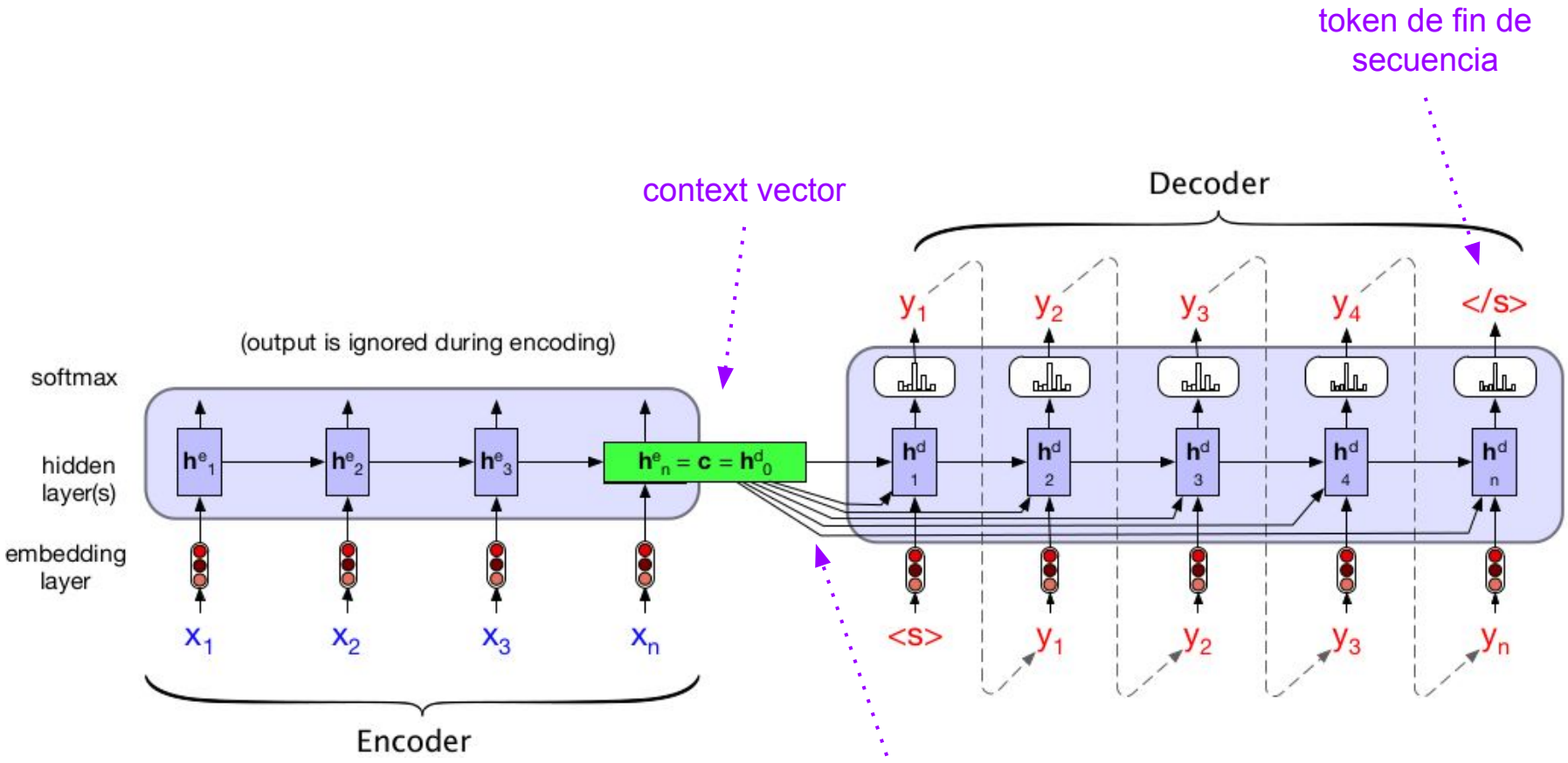
2023

Grupo de Procesamiento de Lenguaje Natural
Instituto de Computación

Bibliografía

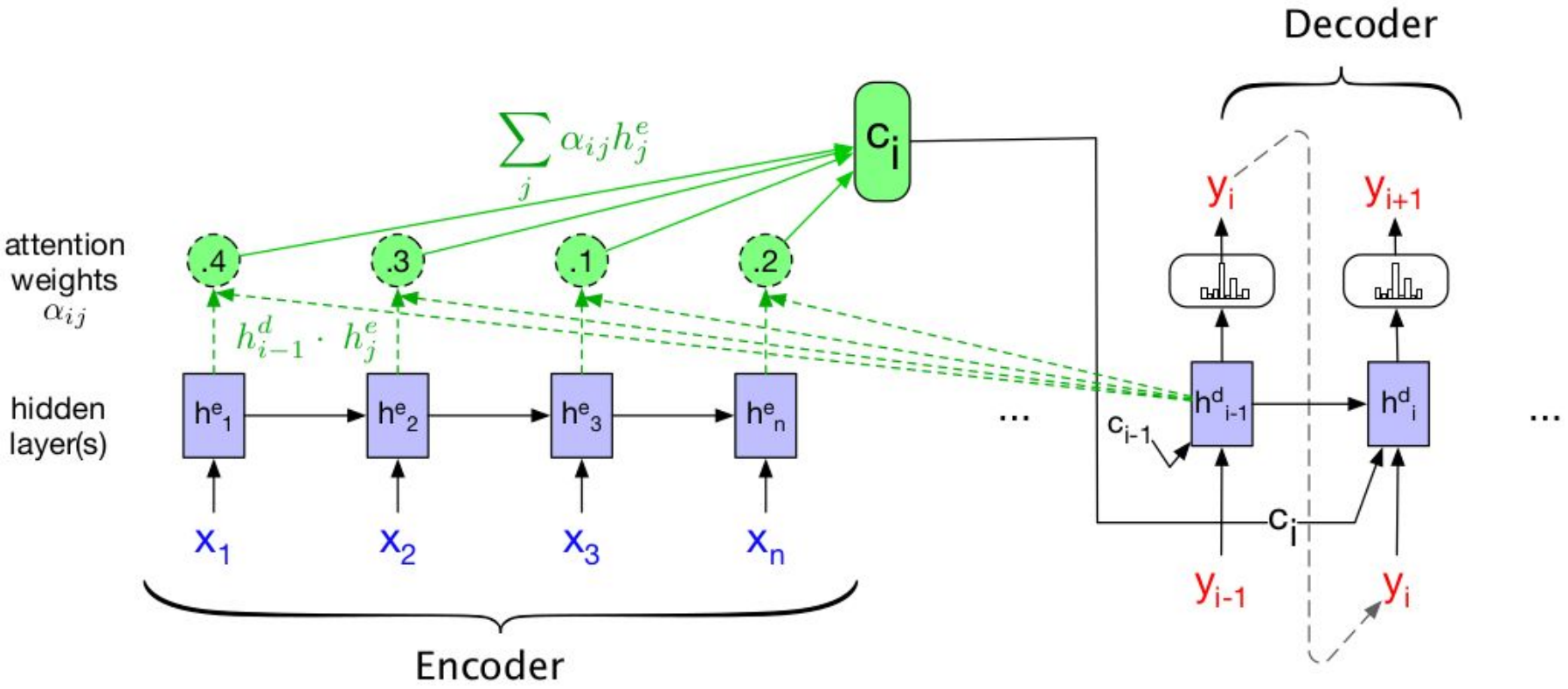
- Jurafsky & Martin, 3rd Ed. (draft) - Capítulos 9, 13 y 2
- Clase de Traducción Automática de IntroPLN
- Papers…

Encoder-Decoder



debido a que la influencia de c sobre la decodificación de la secuencia puede ir disminuyendo, se suele incluir en cada paso

Mecanismo Atencional




Traducción Automática

Traducción Automática

Machine Translation (MT)

Uno de los primeros problemas de PLN

¿Por qué es difícil?

- Tipologías lingüísticas: SVO vs SOV
- Divergencia léxica: pata vs pierna vs leg
- Diferencias morfológicas: aglutinante vs fusional
- Densidad referencial: sujetos omitidos

Traducción Automática

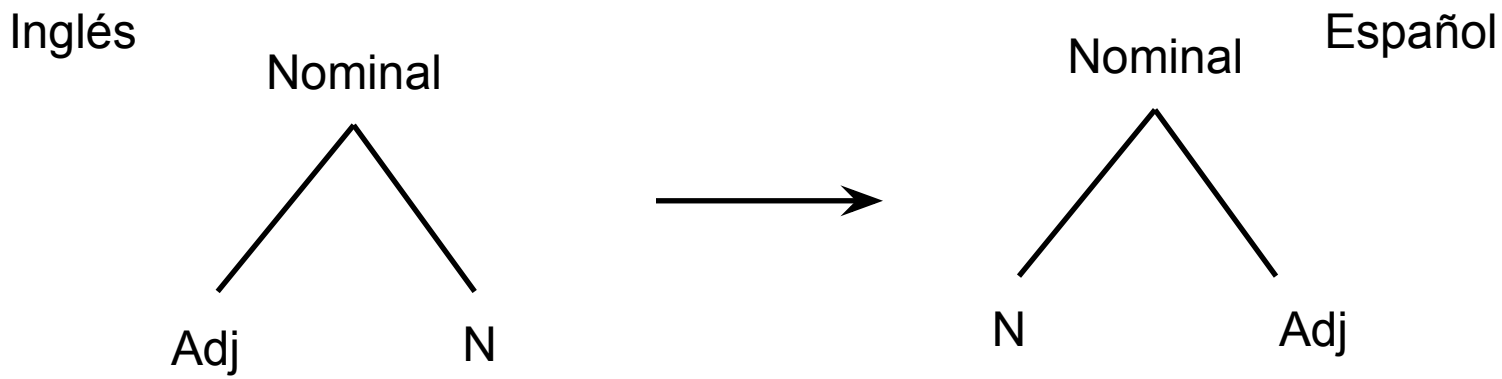
Ha ido evolucionando junto con el PLN (y el aprendizaje automático)

- Métodos basados en reglas (1950s - 1990s)
- Métodos estadísticos (2000s - 2010s)
- Métodos neuronales (2010s hasta hoy)

Métodos Basados en Reglas

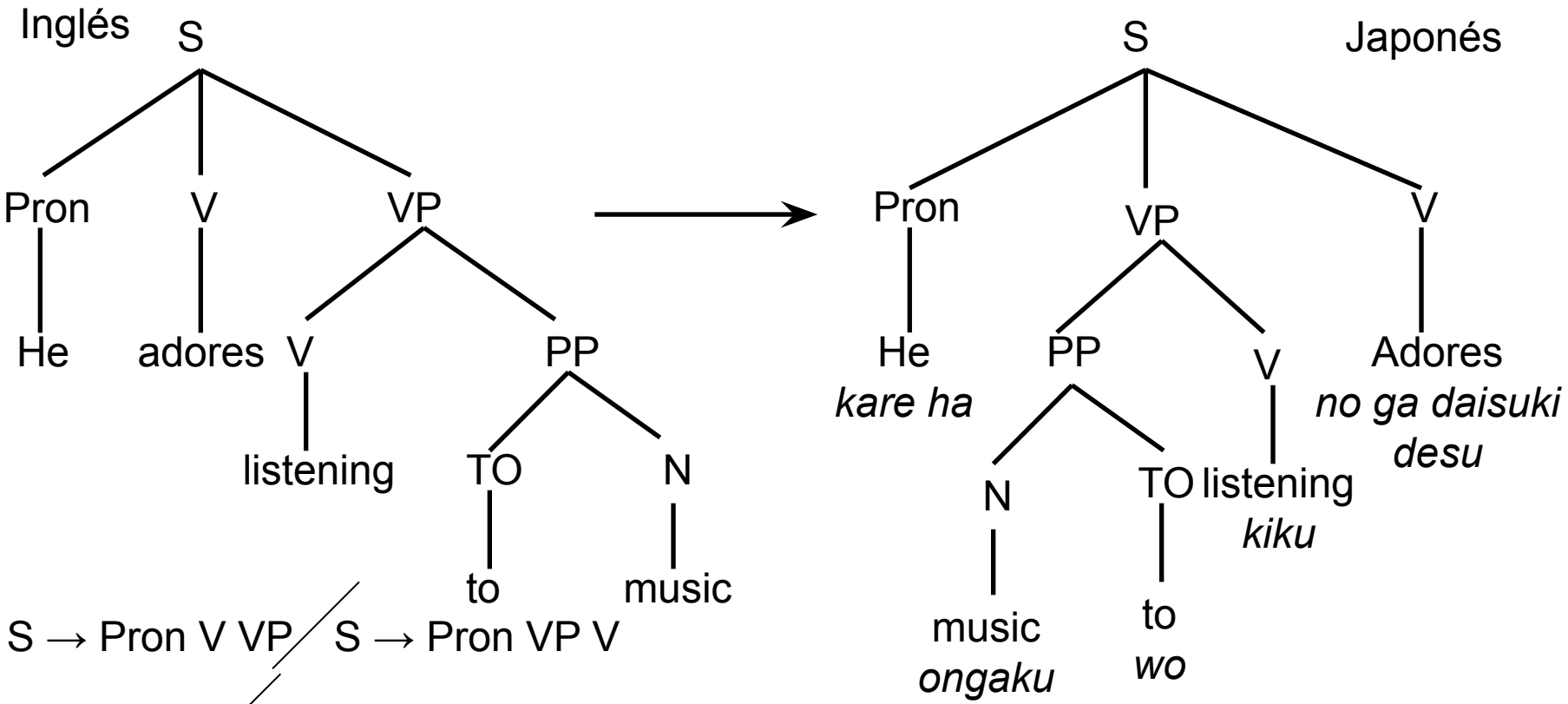
Transferencia Sintáctica

- Parsing del lenguaje origen
- Generación en en lenguaje destino
- Reglas de transferencia entre árboles y subárboles



$\text{Nom} \rightarrow \text{Adj N} \;/\; \text{Nom} \rightarrow \text{N Adj}$

Métodos Basados en Reglas

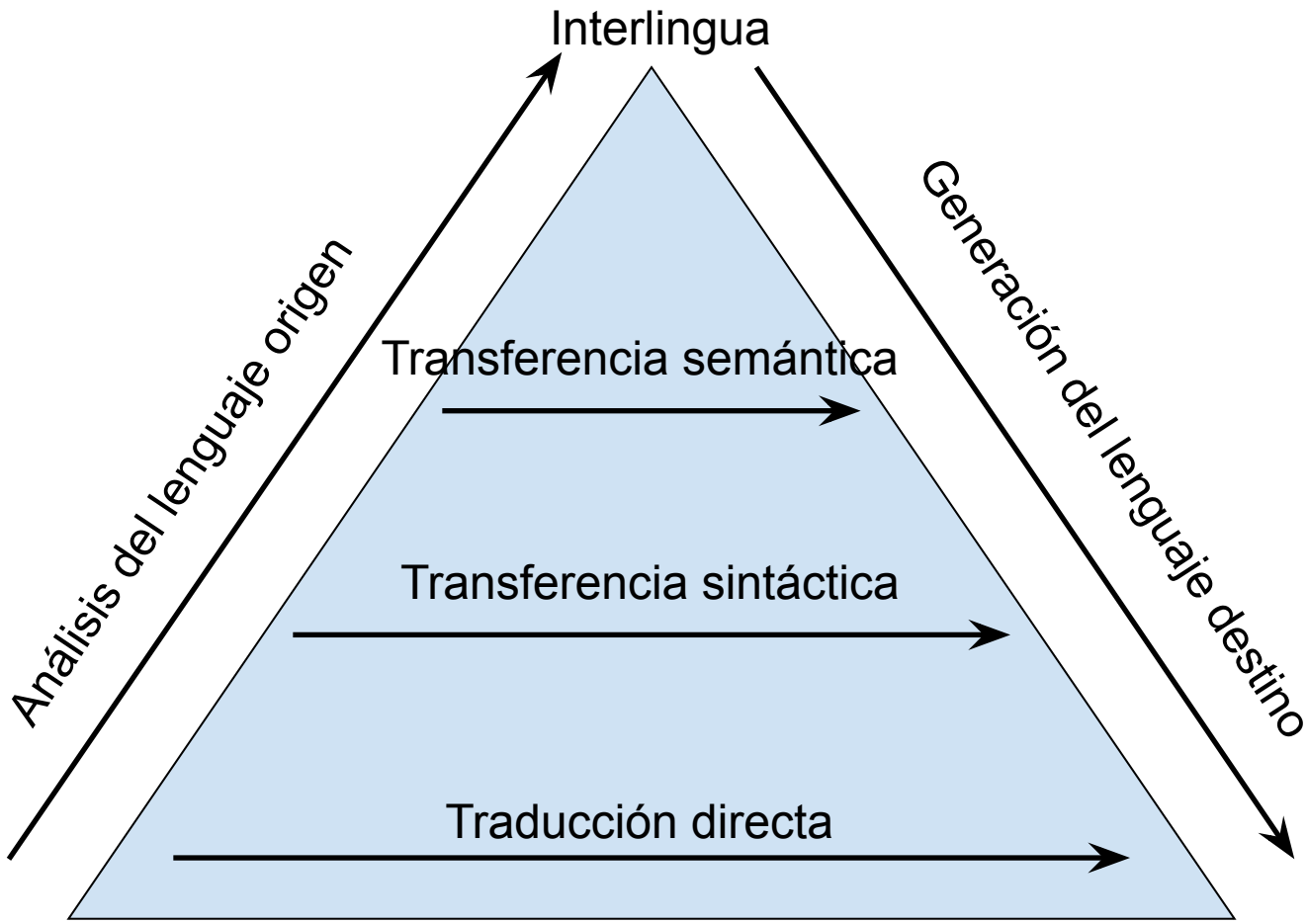


S → Pron V VP / S → Pron VP V

VP → V PP / VP → PP V

PP → TO N / PP → N TO

Métodos Basados en Reglas



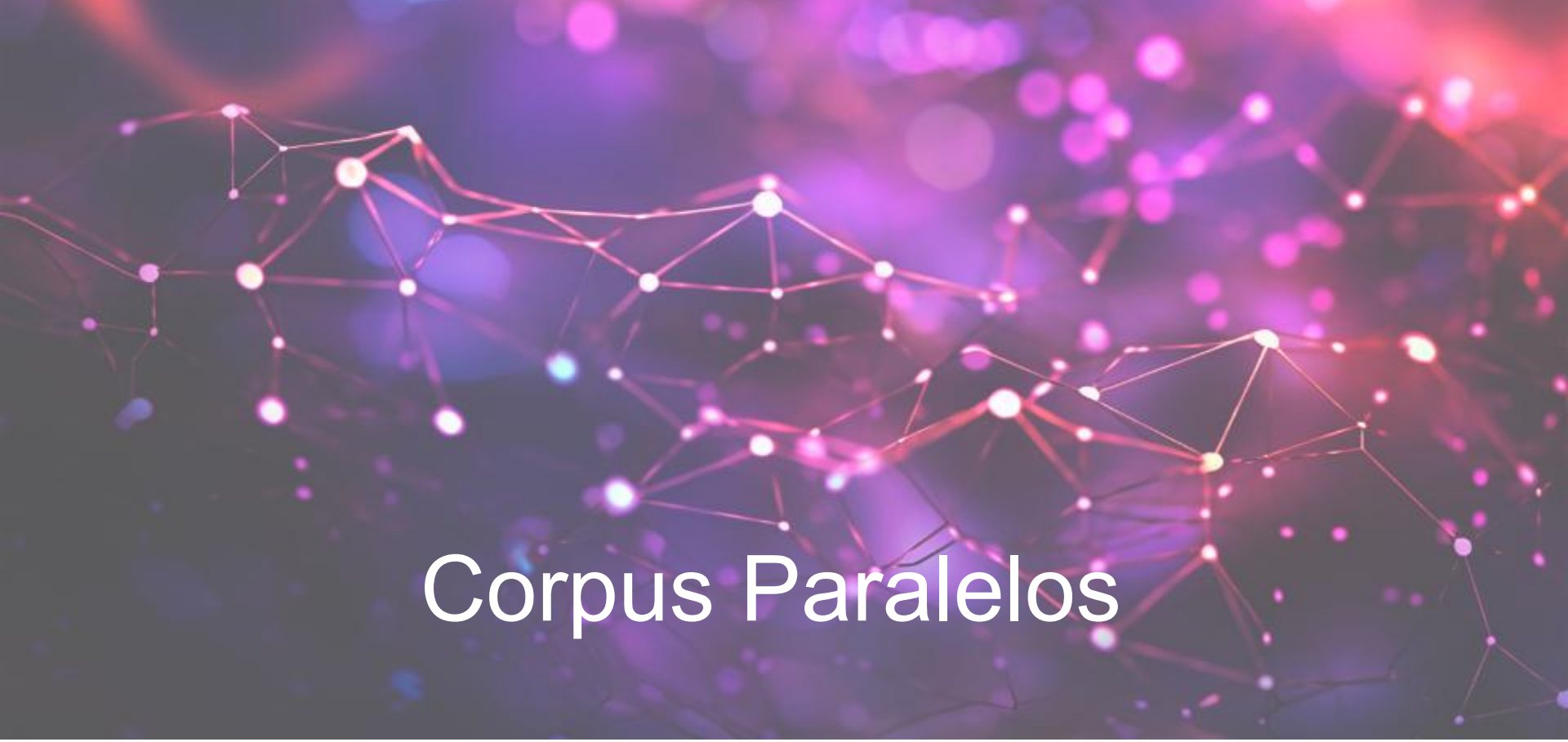
A survey of formal grammars and algorithms for recognition and transformation in machine translation
Vauquois, 1968

Métodos Neuronales

Vemos el problema de traducción como un problema de aprendizaje automático

Cosas que necesitamos definir

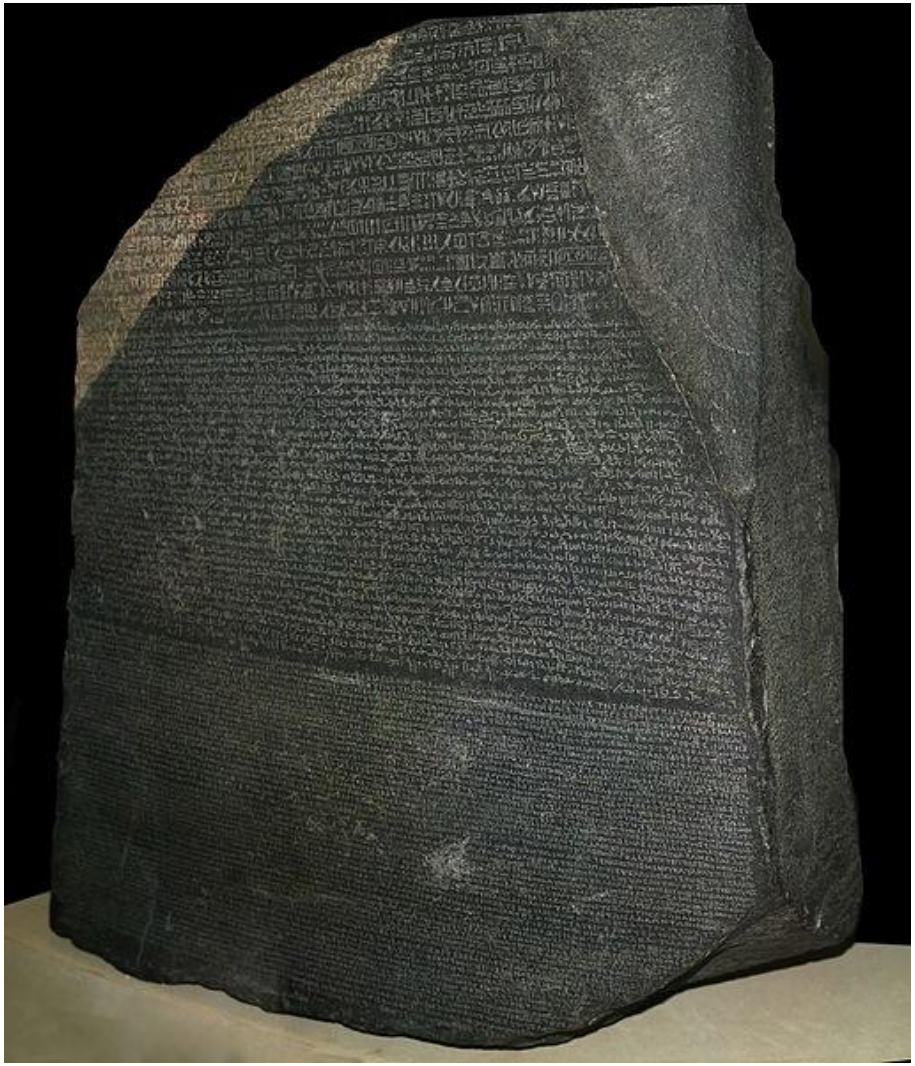
- ¿Cómo son los datos en MT?
  - Corpus paralelos
- ¿Cómo representamos las palabras?
  - Tokenización, embeddings
- ¿Qué modelos se utilizan?
  - Modelos seq2seq: encoder-decoder con RNNs, LSTMs, mecanismo atencional, transformers…
- ¿Qué medimos la performance del sistema?
  - Diferentes métricas: BLEU, chrF…


Corpus Paralelos

Corpus Paralelos

Un ejemplo famoso de corpus paralelo:

La Piedra de Rosetta



Corpus Paralelos

- Conjuntos de pares de textos
  Un texto en el idioma origen y otro en el idioma destino

- Hay corpus paralelos para un puñado de pares de idiomas
  - Árabe-Inglés
  - Chino-Inglés
  - Las lenguas europeas más usadas
    - Europarl: 11 idiomas, c.44M de palabras por idioma
  - United Nations Parallel Corpus
    - 10M palabras - árabe, chino, español, francés, inglés, ruso

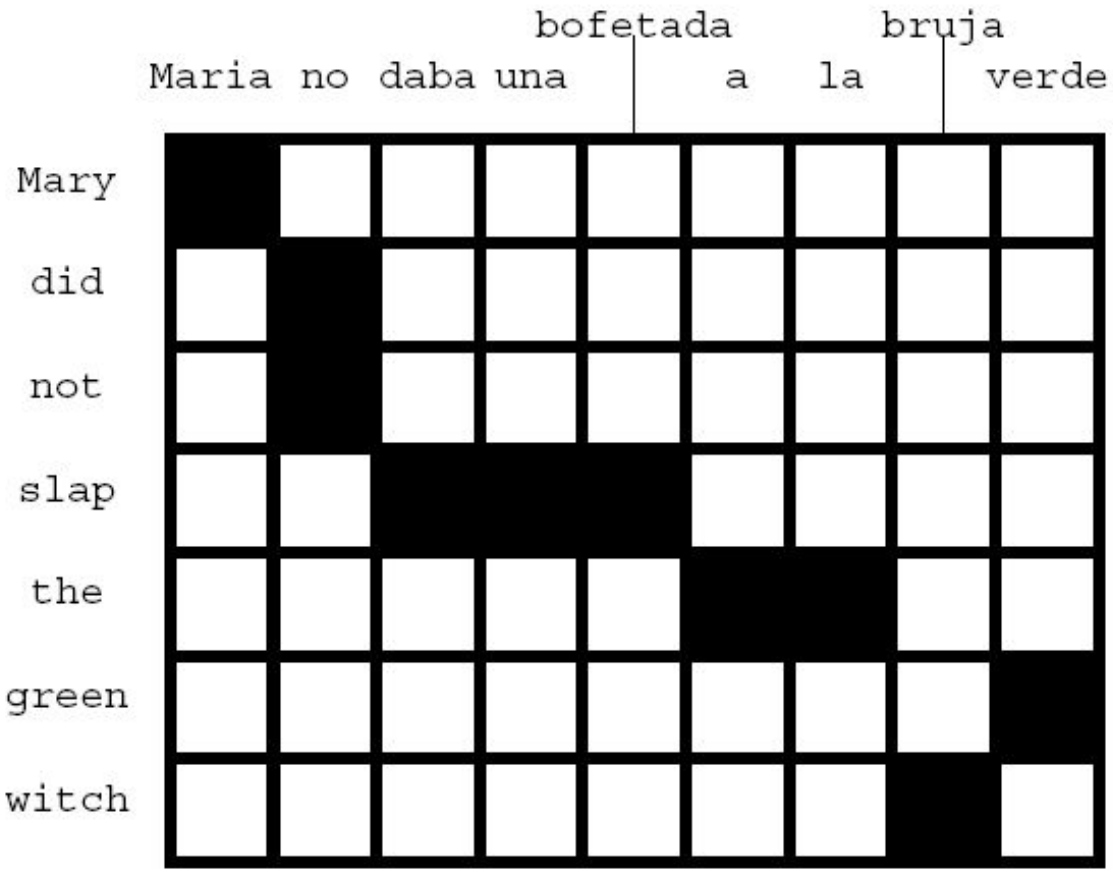
- ...pero para la mayoría de los pares de lenguas, no existen corpus

Corpus Paralelos

Diferentes tipos de alineación:

- Alineados a nivel de documento
- Alineados a nivel de oración
  Programación dinámica (algoritmo de Gale y Church)
  Distancia coseno con embeddings multilingües
- Alineados a nivel de palabra
  Este es el ideal, pero en general no existen

Alineación


bofetada
bruja
Maria no daba una a la verde
Mary
did
not
slap
the
green
witch

Traducción Automática Neuronal

Traducción Automática Neuronal

Modelo codificador-decodificador (encoder-decoder)

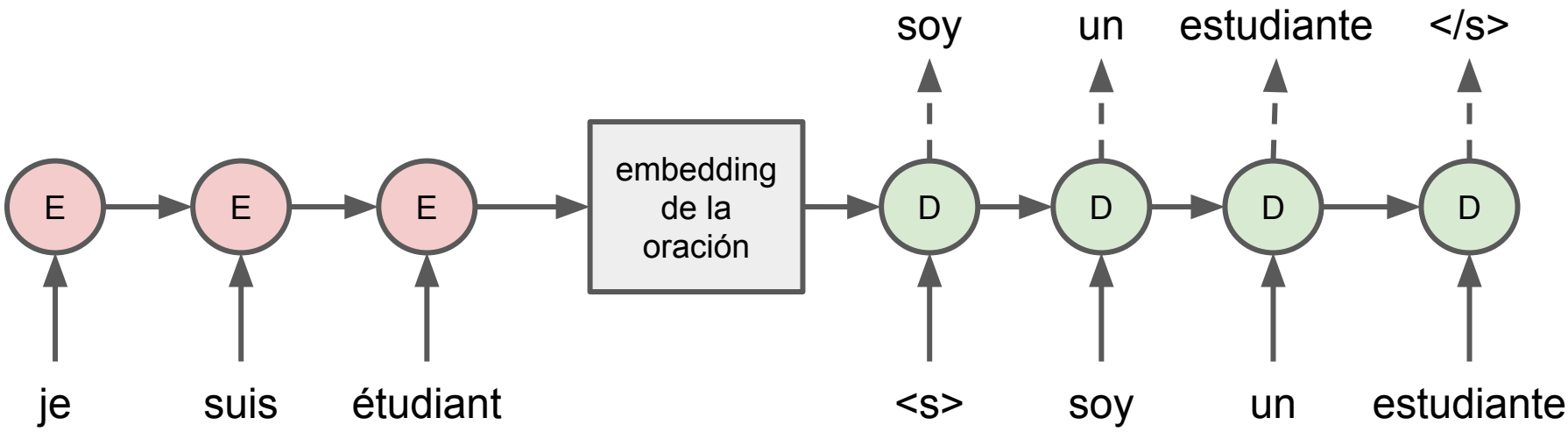
Una red codifica la oración en el idioma origen

Una red genera la decodificación en el idioma destino

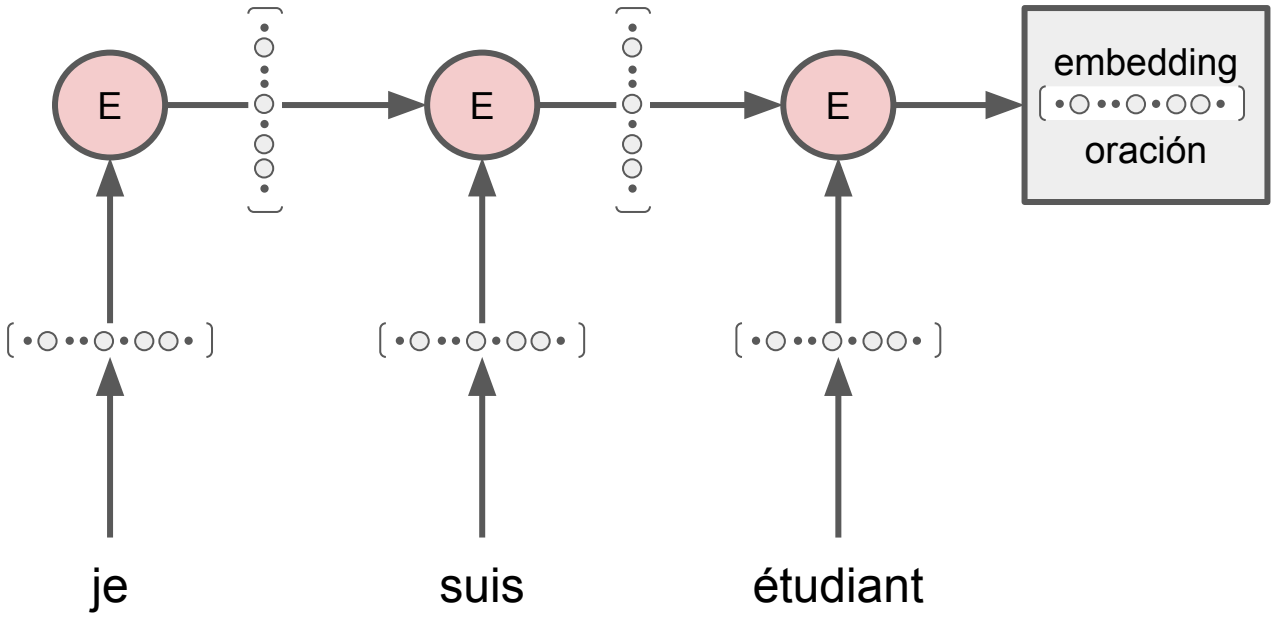
Implementaciones habituales:

- RNNs (LSTMs) con mecanismo atencional
- Transformers

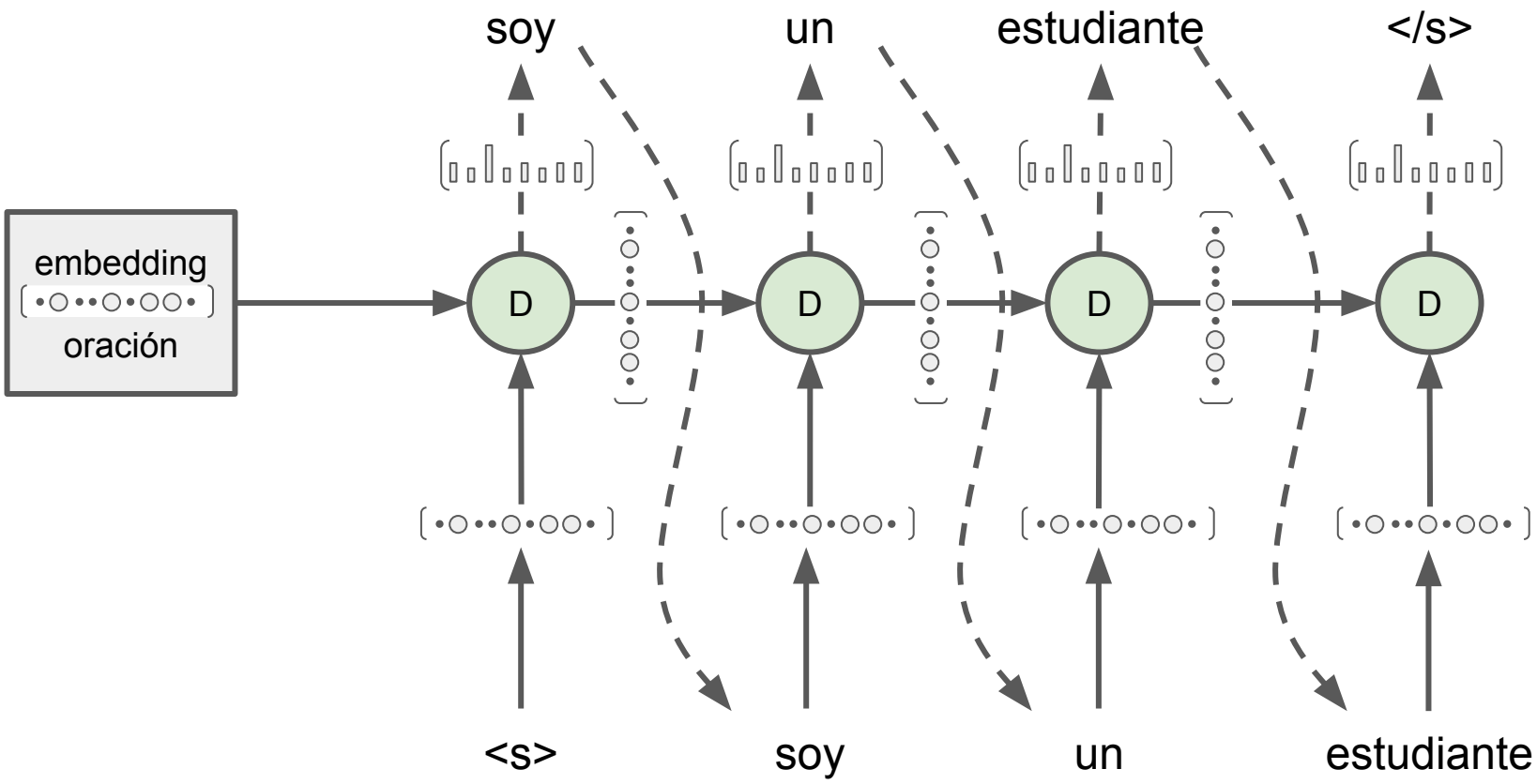
Modelo Encoder-Decoder



Encoder



Decoder

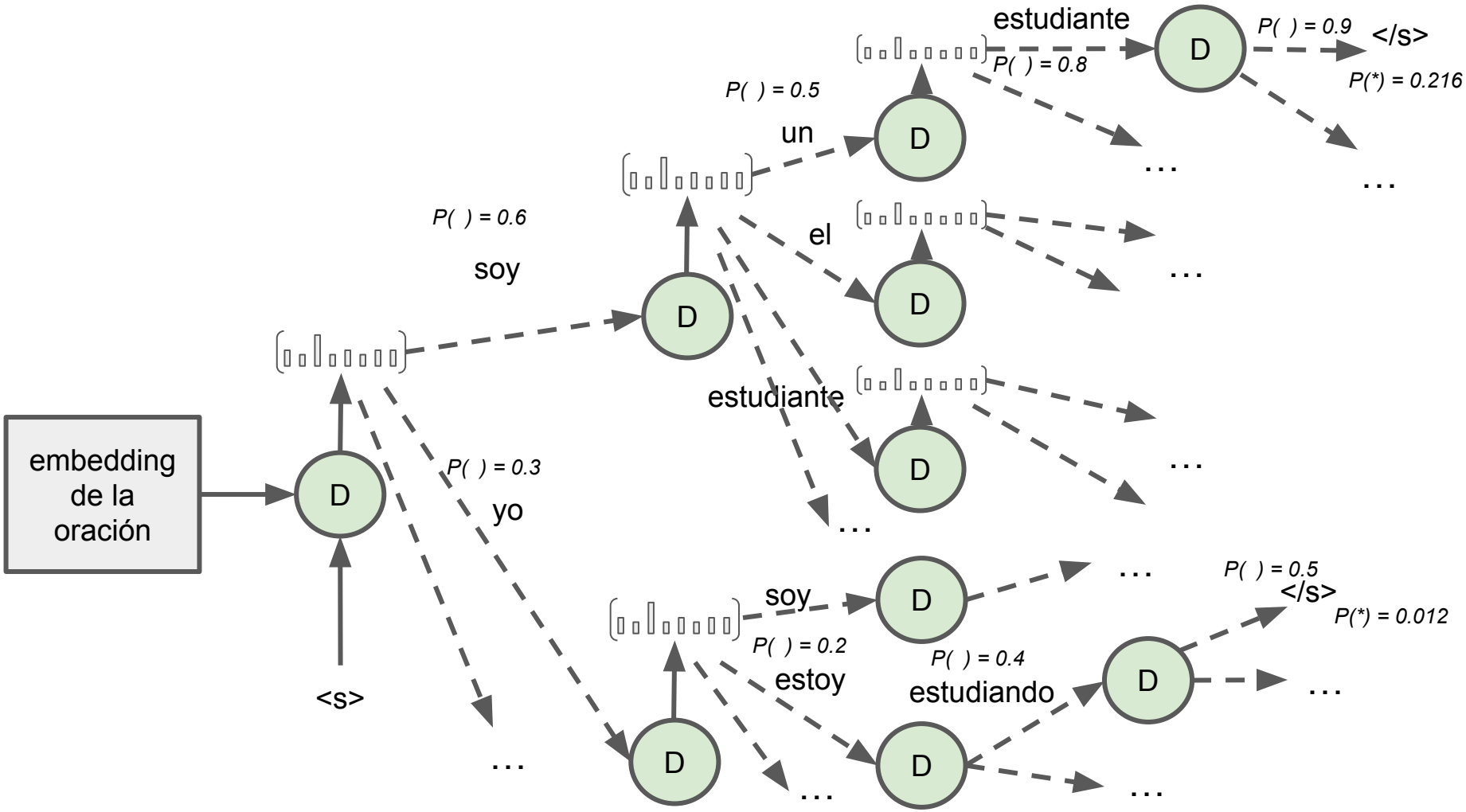

soy
un
estudiante
</s>
embedding
oración
D
D
D
D
<s>
soy
un
estudiante

Entrenamiento

End-to-end: usamos pares de oraciones alineados como ejemplos de entrada y salida esperadas

Cada entrada del decoder usa la palabra correcta esperada, y no la salida del paso anterior *(teacher forcing)*

Beam Search


embedding de la oración
D
<s>
P( ) = 0.6
soy
P( ) = 0.3
yo
un
P( ) = 0.5
el
estudiante
P( ) = 0.8
P( ) = 0.9
</s>
P(*) = 0.216
soy
P( ) = 0.2
estoy
P( ) = 0.4
estudiando
P( ) = 0.5
</s>
P(*) = 0.012

Beam Search

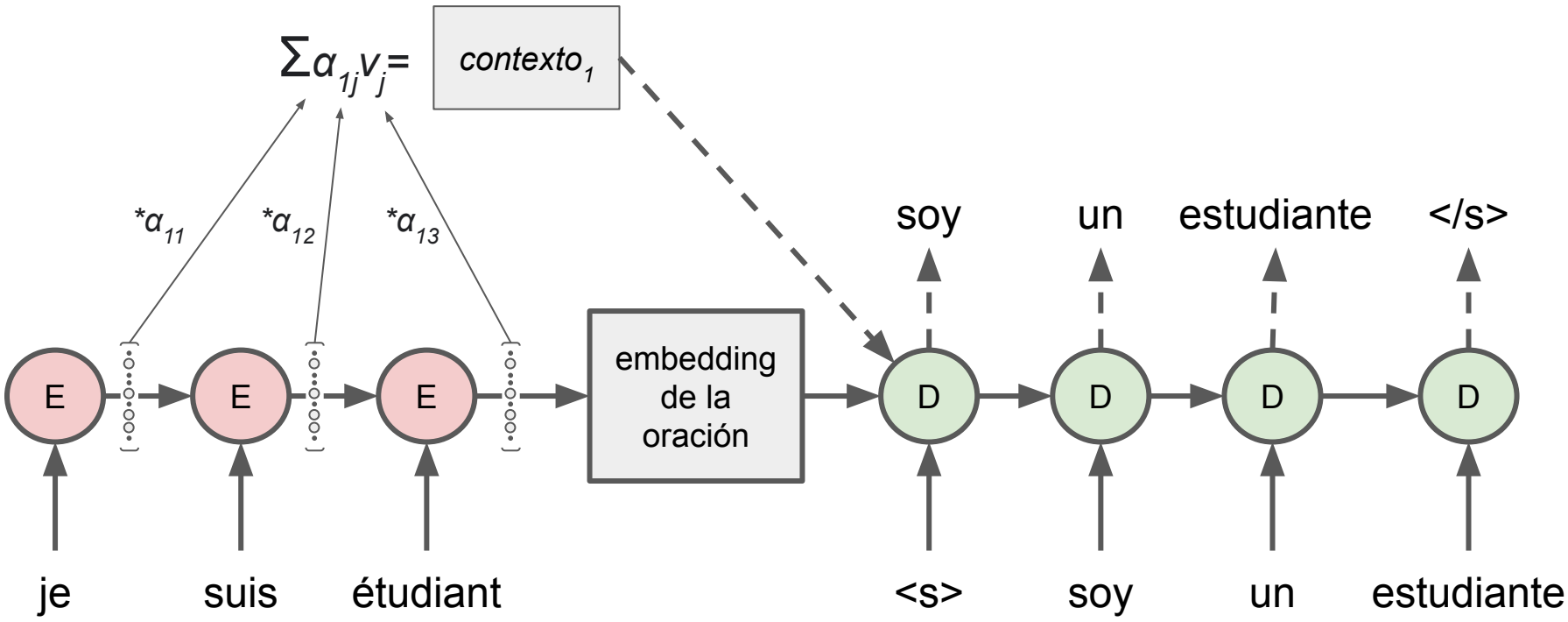
- Seleccionamos un tamaño del beam *n*
- En el primer paso del decoder, elegimos las *n* palabras más probables
- Expandimos a partir de esas *n*, y nos quedamos con los *n* pares más probables
- Iteramos el proceso hasta llegar a generar las </s>, siempre nos quedamos con las *n* más probables

Beam Search

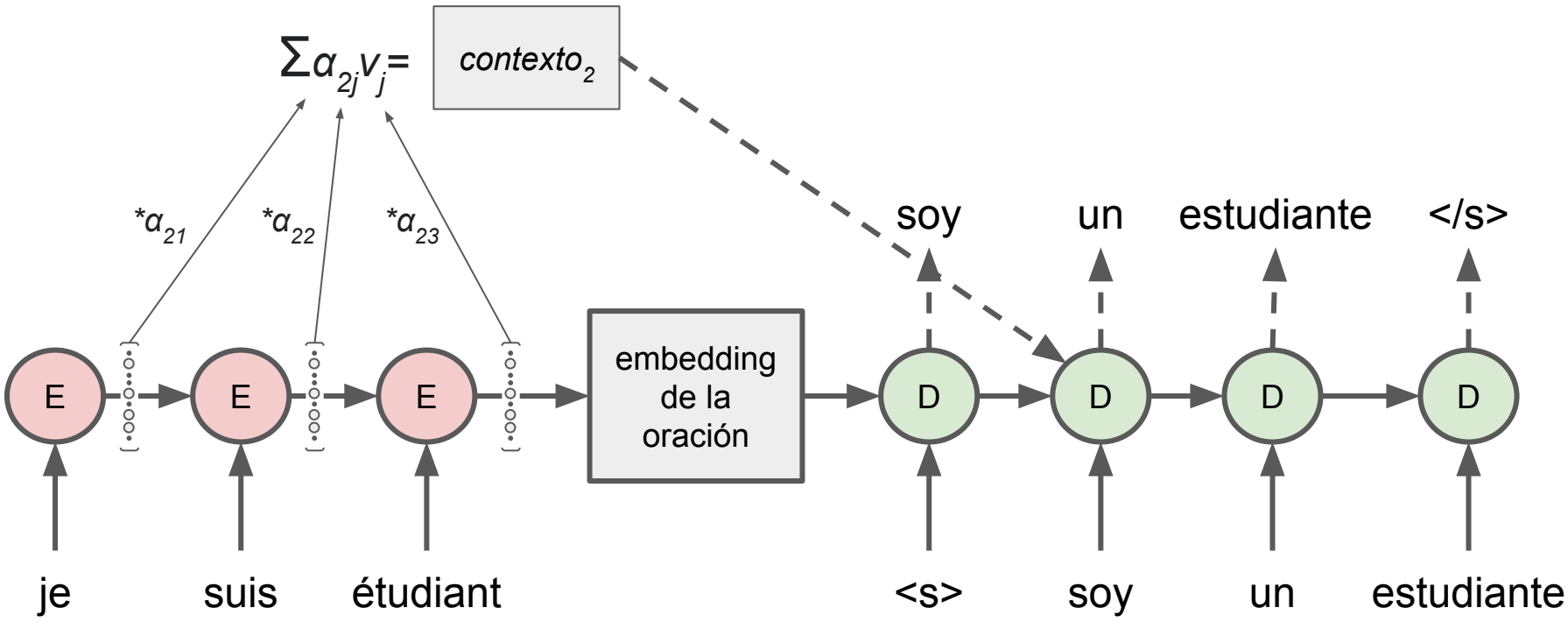
No es un proceso perfecto!

- No asegura que la decodificación más probable esté en el beam
- Pero sí asegura que la nueva solución encontrada va a ser al menos tan buena como la greedy
- Aún puede ser lento, se le pueden hacer mejoras de performance como podas tempranas

Modelo Atencional



Modelo Atencional



Modelo Atencional

En cada paso del decoder creo un vector de contexto

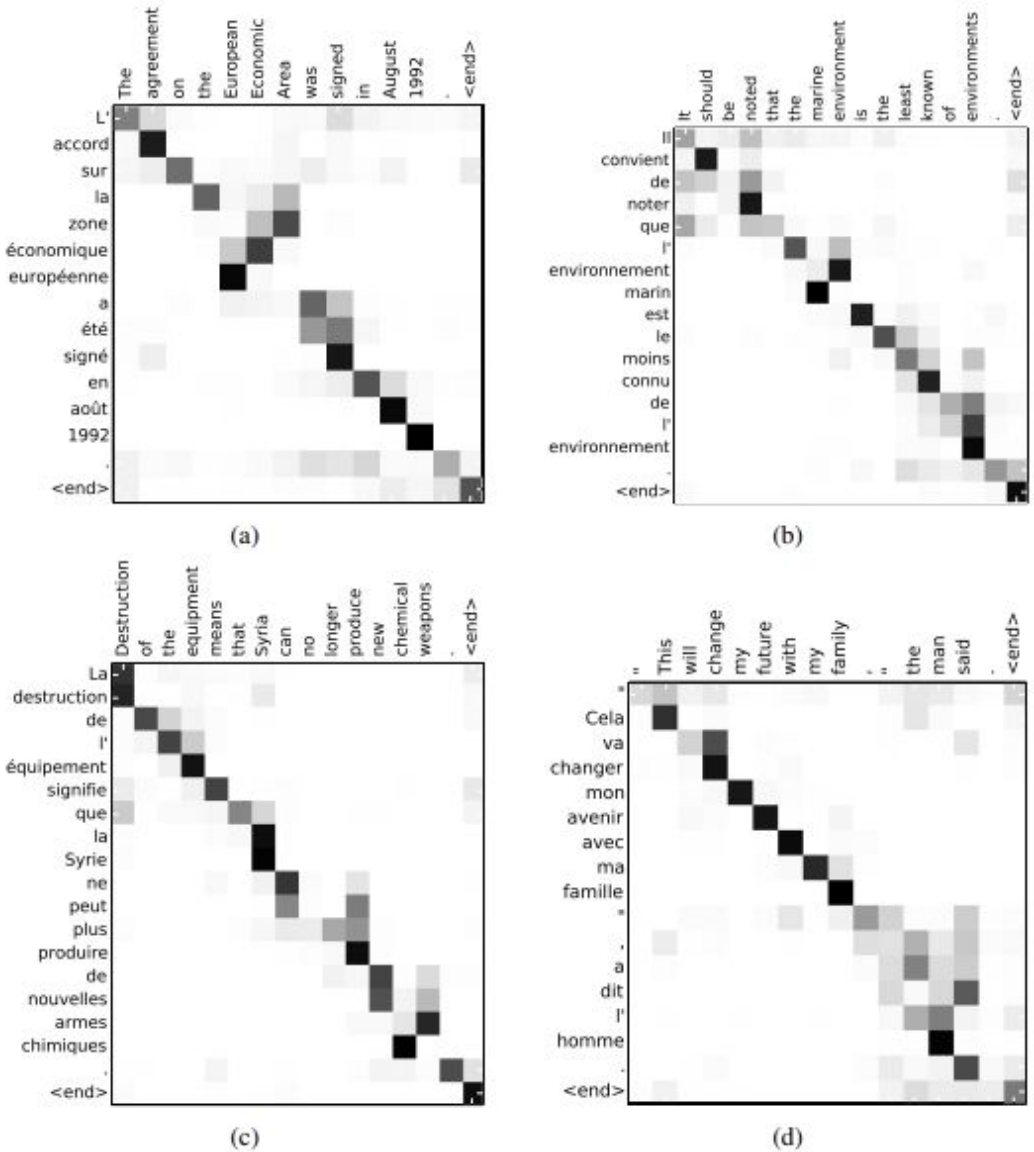
Promedio ponderado de los embeddings del encoder

Ponderaciones α_{ij} : matriz de largo origen * largo destino

Vimos diferentes maneras de aprender esas ponderaciones

- Atención aditiva
- Atención multiplicativa

Modelo Atencional




Tokenización

Representación de las palabras

¿Cómo tratamos las palabras de los idiomas?

- Se define un vocabulario fijo para cada idioma
- O se puede usar un vocabulario compartido entre idiomas
- Esto permite usar embeddings compartidos para representación de tokens

¿Pero cómo son esos tokens?

Tokenización

Es un tópico general de PLN. Formas clásicas de tokenización incluyen:

- Por espacios
- Expresiones regulares
- Morfemas

En MT y métodos modernos se suelen usar métodos estadísticos que obtienen información sub-palabra:

- BPE, ULM, wordpiece

Algoritmo BPE

Codificación de a pares de bytes (Byte-Pair Encoding)

El corpus a procesar es una colección de tokens con separación simple por espacios

El vocabulario inicial está compuesta de todas las letras (bytes) encontradas en el corpus, más un símbolo especial de fin de palabra “_”

En cada paso, uniremos un par de tokens del vocabulario hasta alcanzar la cantidad *k* tokens en total (parámetro del sistema)

Algoritmo BPE

Corpus

```
5  l o w _
2  l o w e s t _
6  n e w e r _
3  w i d e r _
2  n e w _
```

9 veces

Vocabulario

```
_, d, e, i, l, n, o, r,
s, t, w
```

Algoritmo BPE

Corpus

```
5  l o w _
2  l o w e s t _
6  n e w er _
3  w i d er _
2  n e w _
```

9 veces

Vocabulario

```
_, d, e, i, l, n, o, r,
s, t, w, er
```

Algoritmo BPE

Corpus

```
5  l o w _
2  l o w e s t _
6  n e w er_
3  w i d er_
2  n e w _
```

8 veces

Vocabulario

```
_, d, e, i, l, n, o, r,
s, t, w, er, er_
```

Algoritmo BPE

Corpus

```
5  l o w _
2  l o w e s t _
6  ne w er_
3  w i d er_
2  ne w _
```

8 veces

Vocabulario

```
_, d, e, i, l, n, o, r,
s, t, w, er, er_, ne
```

Algoritmo BPE

Corpus

7 veces

```
5  l o w _
2  l o w e s t _
6  new er_
3  w i d er_
2  new _
```

Vocabulario

```
_, d, e, i, l, n, o, r,
s, t, w, er, er_, ne,
new
```

Algoritmo BPE

Corpus

```
5  lo w _

2  lo w e s t _

6  new er_

3  w i d er_

2  new _
```

Vocabulario

```
_, d, e, i, l, n, o, r,
s, t, w, er, er_, ne,
new, lo
```

Algoritmo BPE

Sigue uniendo de a dos palabras:

`(lo, w) → _, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo, low`

`(new, er_) → _, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo, low, newer_`

`(low, _) → _, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo, low, newer_, low_`

Hasta que hayamos hecho *k* uniones

Notar que palabras enteras quedan en el vocabulario (las frecuentes)

Algoritmo BPE

Al momento de codificar un texto nuevo, recorremos la lista haciendo las uniones en el orden que las fuimos encontrando

`_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo, low, newer_, low_`

`l o w e r _ → l o w er _ → l o w er_ → lo w er_ → low er_`

“lower” no estaba en el vocabulario, y la podemos representar con dos tokens: (low, er_)


Evaluación

Evaluación

La forma ideal para evaluarlo es con anotadores humanos

Dada la oración en el idioma origen, evaluar la traducción candidata

- Adecuación (1 al 5): qué tanto se preserva la semántica
- Fluidez (1 al 5): qué tan bien suena en el idioma destino

Problemas:

- Muy caro!
- No reutilizable

Evaluación

Aunque no son perfectas, las métricas automáticas son lo más habitual

- WER
- BLEU
- METEOR
- chrF

Todas se basan en tener una o más traducciones de referencia

Se olvidan de la oración origen: solo se compara las traducciones candidatas con las de referencia

Evaluación

Métrica BLEU

$$BLEU = BP \exp\left(\sum_{1}^{N} w_n \log p_n \right)$$

- Compara un conjunto de traducciones candidatas con un conjunto de traducciones de referencia
- Cuenta n-gramas (de tokens) presentes en los candidatos que también estén en las referencias (n = 1,2,3,4)
- Incluye una penalización por brevedad (BP) para que las traducciones demasiado cortas tengan menos puntos

BLEU: a Method for Automatic Evaluation of Machine Translation
Papineni et al., 2002

Métrica BLEU

Definición de BP:

$$BP = \begin{cases} 1 & (C' \geq R') \\ \exp\left(1 - \frac{R'}{C'}\right) & (C' < R') \end{cases}$$

Donde R’ es el largo total (en palabras) de todas las referencias (documento referencia) y C’ es la el largo total de las traducciones candidatas (documento candidato)

Métrica BLEU

- BLEU generalmente se correlaciona con la evaluación subjetiva humana
- Captura nociones de adecuación y fluidez
  - Unigramas vs n-gramas mayores
- Difícil de interpretar
  - Valor entre 0 y 1
  - Estado del arte inglés-francés 0.48
  - Una buena traducción que no esté en el conjunto de referencia será penalizada

Métrica chrF

A diferencia de BLEU, que cuenta n-gramas de tokens, chrF utiliza n-gramas de caracteres

$$\mathrm{chrF}\beta = (1+\beta)\frac{\mathrm{chrP}\cdot\mathrm{chrR}}{\beta^2\cdot\mathrm{chrP}+\mathrm{chrR}}$$

- chrP: cantidad de *n-gramas* de caracteres de la hipótesis que están en la referencia
- chrP: cantidad de *n-gramas* de caracteres de la referencia que están en la hipótesis
- En general n≤4 o 6, y β=3

chrF: character n-gram F-score for automatic MT evaluation
Popović, 2015

Métrica chrF

Su valor está entre 0 y 1 como BLEU, y también penaliza traducciones que no estén en las referencias

Pero es menos estricta que BLEU, porque si el sistema le erra al traducir una palabra, podría traducir bien algunas sílabas

Bueno para hacer comparaciones con idiomas morfológicamente ricos, ya que hay muchas flexiones de una misma palabra


Traducción Automática