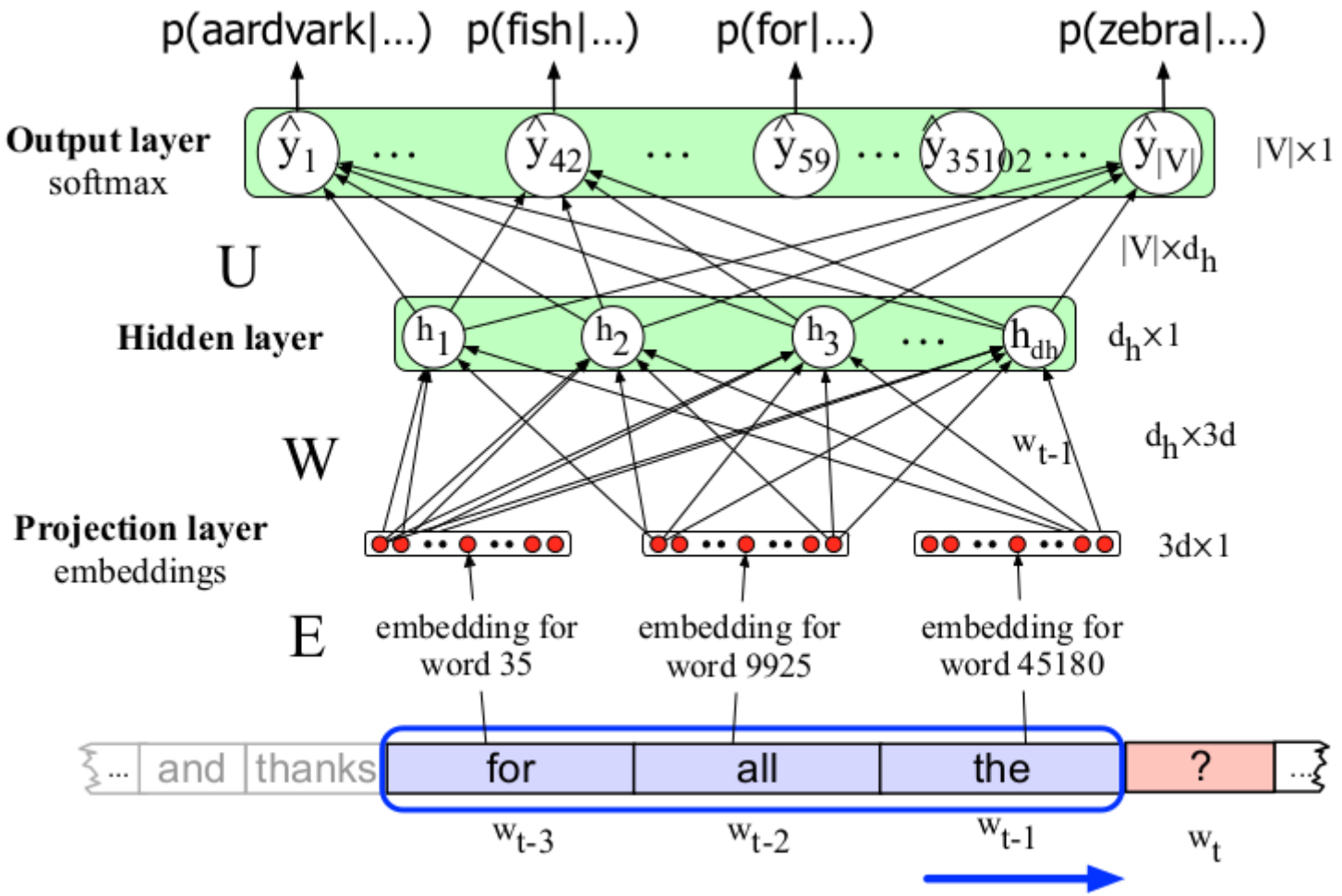
IIC 3800 Tópicos en CC
NLP

https://github.com/marcelomendoza/IIC3800

- MODELOS DE LENGUAJE -

Background: Modelos de lenguaje neuronales

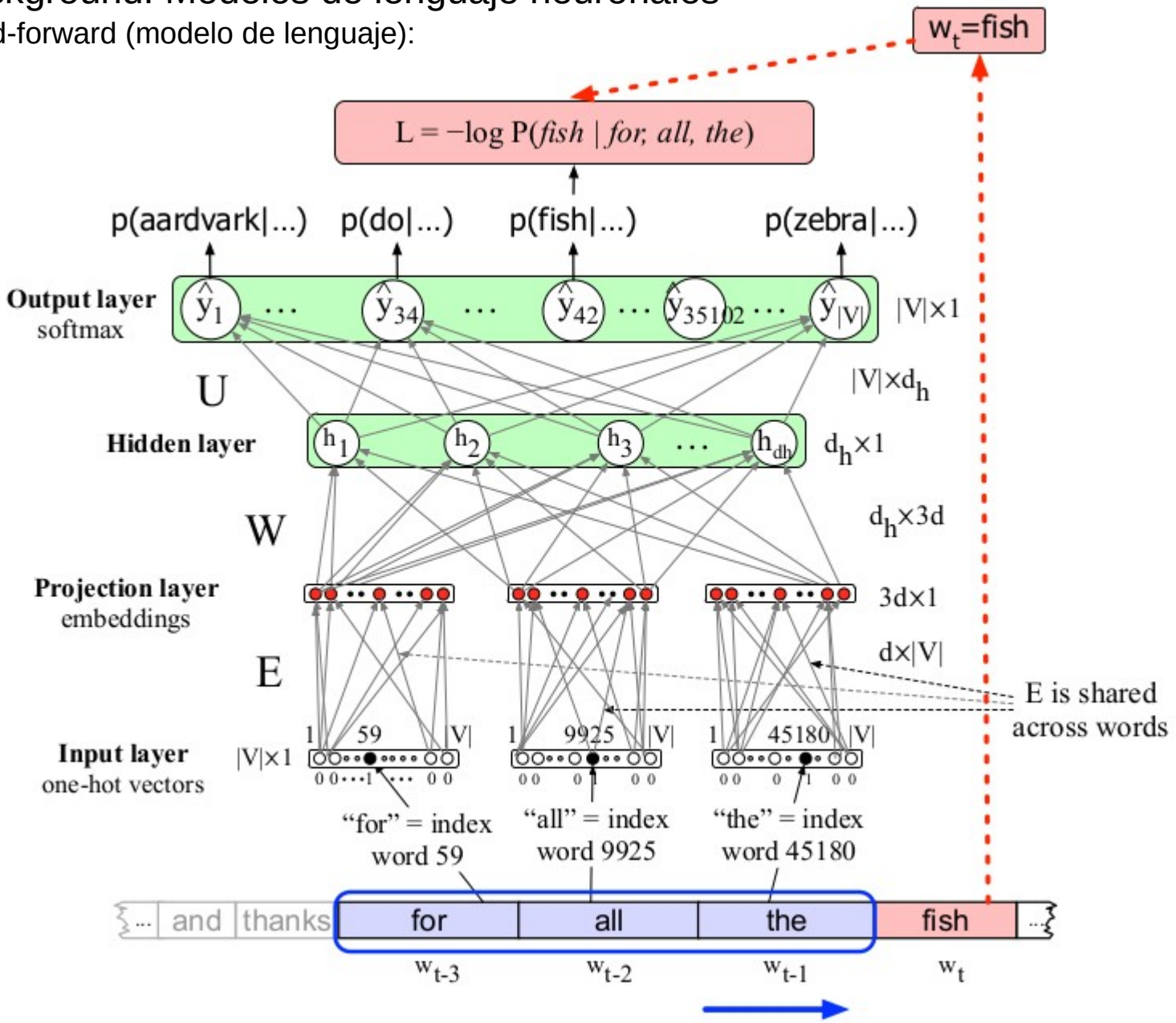
Feed-forward (modelo de lenguaje):



Background: Modelos de lenguaje neuronales

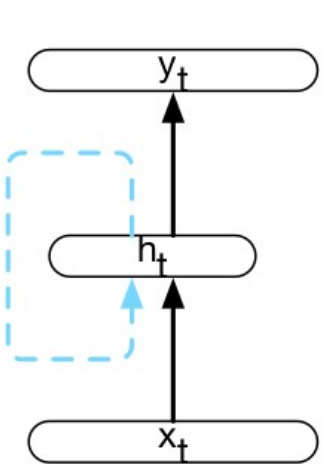
Feed-forward (modelo de lenguaje):

$$
\begin{aligned}
e &= (Ex_1, Ex_2, ..., Ex) \\
h &= \sigma(We + b) \\
z &= Uh \\
\hat{y} &= \text{softmax}(z)
\end{aligned}
$$

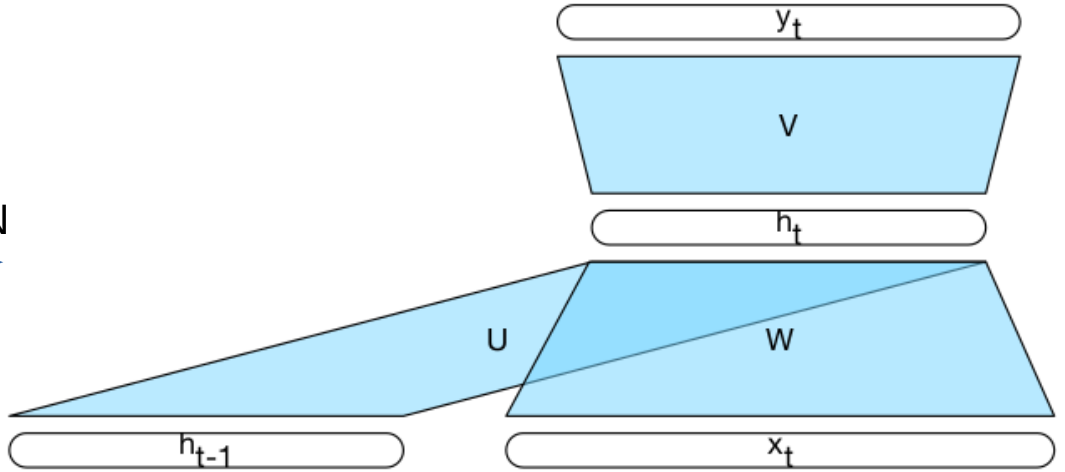


Background: Modelos de lenguaje neuronales

Red recurrente:



como FFNN



$$h_t = g(Uh_{t-1} + Wx_t)$$
$$y_t = f(Vh_t)$$

Típicamente, en NLP: $y_t = \text{softmax}(Vh_t)$

Background: Modelos de lenguaje neuronales

Red recurrente:

output distribution

$$\hat{\boldsymbol{y}}^{(t)} = \text{softmax}\left(\boldsymbol{U}\boldsymbol{h}^{(t)} + \boldsymbol{b}_2\right) \in \mathbb{R}^{|V|}$$

hidden states

$$\boldsymbol{h}^{(t)} = \sigma\left(\boldsymbol{W}_h\boldsymbol{h}^{(t-1)} + \boldsymbol{W}_e\boldsymbol{e}^{(t)} + \boldsymbol{b}_1\right)$$

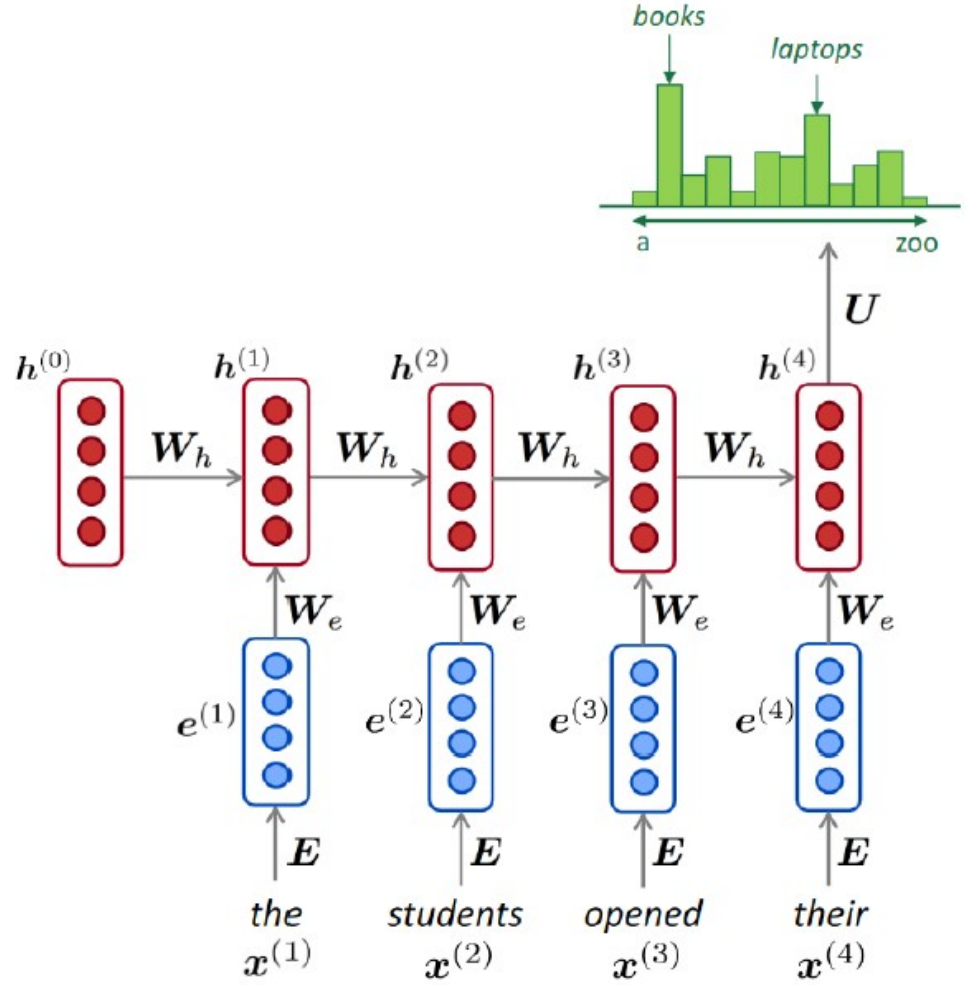
$\boldsymbol{h}^{(0)}$ is the initial hidden state

word embeddings

$$\boldsymbol{e}^{(t)} = \boldsymbol{E}\boldsymbol{x}^{(t)}$$

words / one-hot vectors

$$\boldsymbol{x}^{(t)} \in \mathbb{R}^{|V|}$$



Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C. A Neural Probabilistic Language Model. Journal of Machine Learning Research 3:1137-1155, 2003.

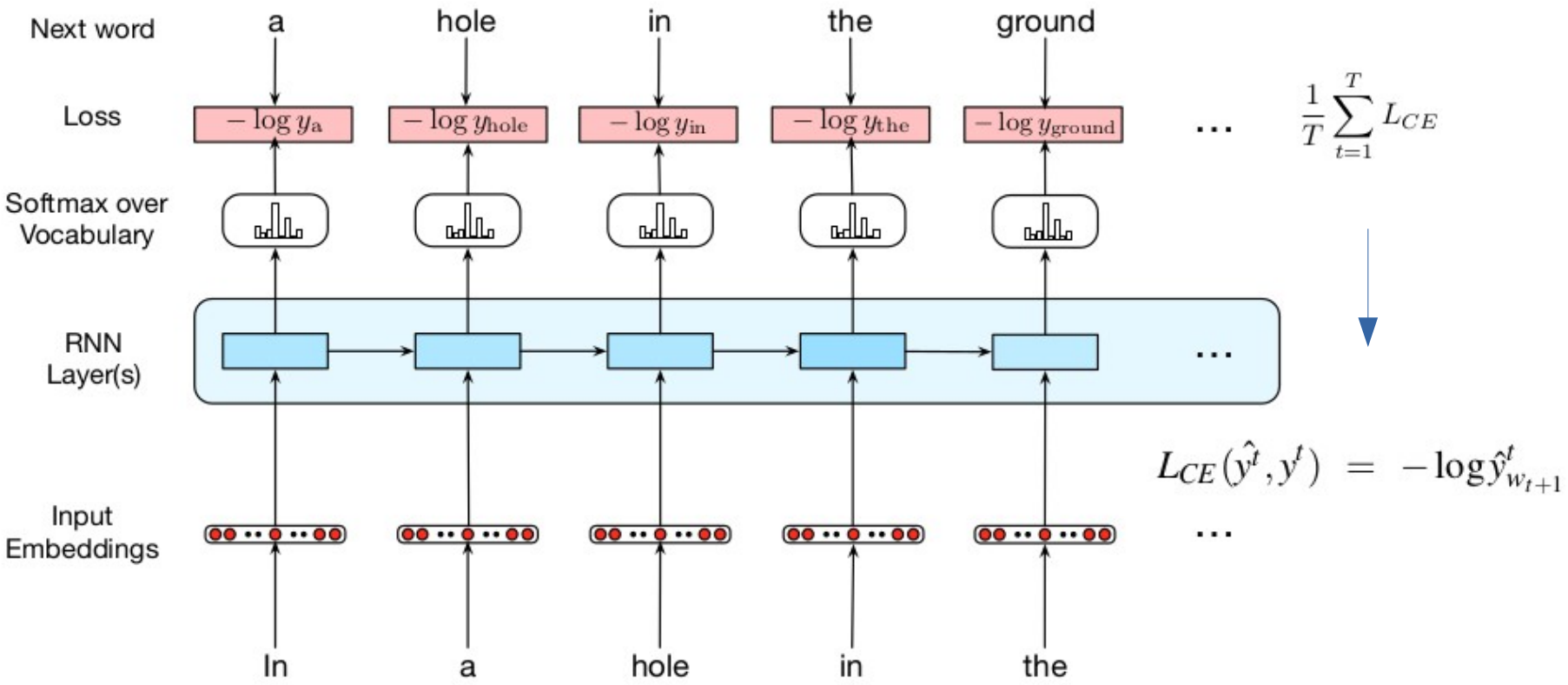
Background: Modelos de lenguaje neuronales

Red recurrente:

$$
\begin{aligned}
e_t &= E^T x_t \\
h_t &= g(U h_{t-1} + W e_t) \\
y_t &= \text{softmax}(V h_t)
\end{aligned}
$$

Entrenamiento autoregresivo:

$$P(w_{t+1} = i | w_{1:t}) = y_t^i$$



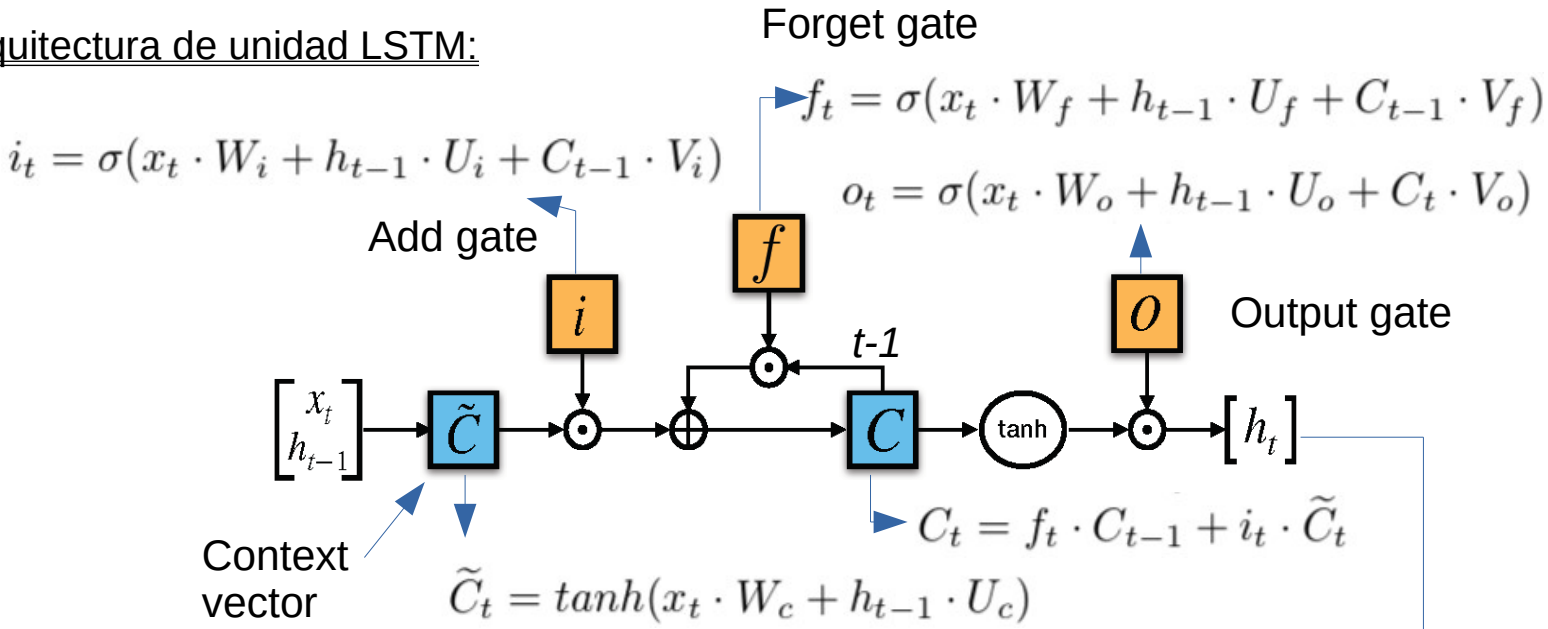
- DEPENDENCIA DEL CONTEXTO -

Extensión a LSTM

Long-short term memory (LSTM): dividen el problema en dos sub-problemas: a) remover información innecesaria, b) agregar información necesaria para resolver una tarea.

Las LSTM hacen lo anterior agregando unidades de control de flujo de información. Cada unidad consiste de una capa FF seguida por una sigmoid (activación). Combinando la salida de la sigmoid con productos, se obtiene un efecto similar al de una máscara binaria.

Arquitectura de unidad LSTM:



"Si x se parece al pasado, f se alinea (mask 1). En otro caso, f hace mask 0."

Dependencia del contexto

FLAIR

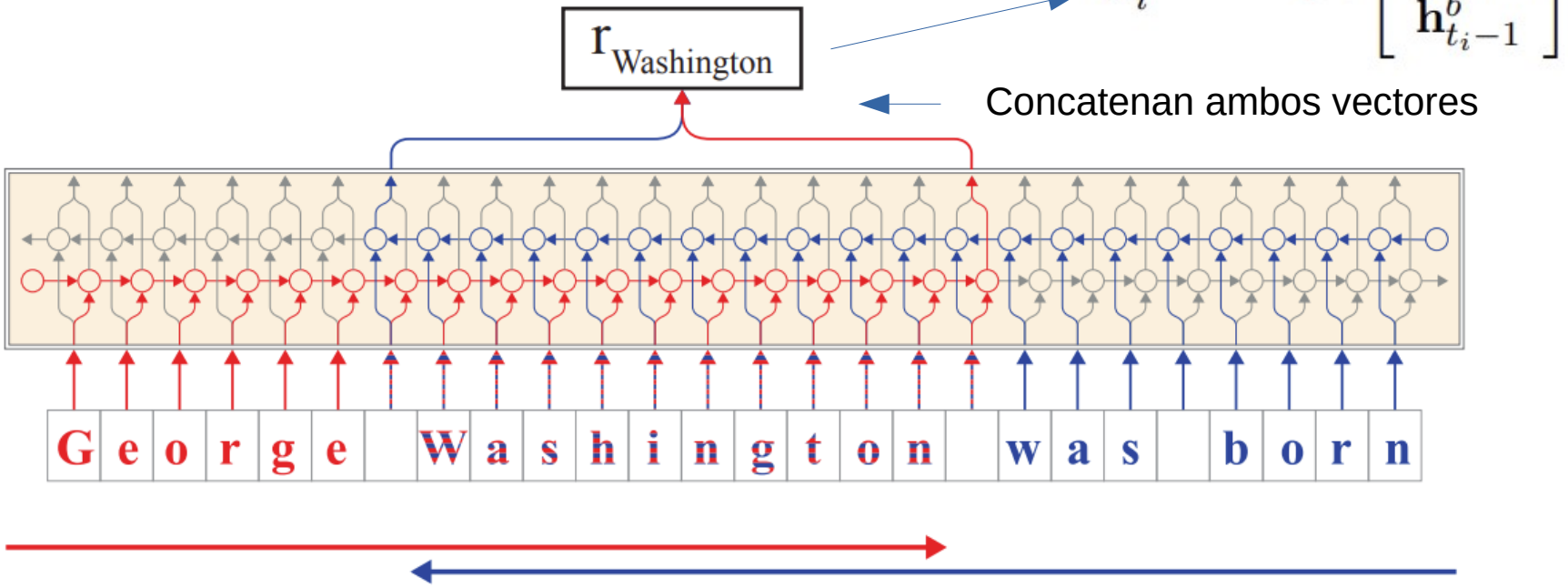
Dada una secuencia de n símbolos, se calculan dos modelos de lenguajes:

LSTM

$$p(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} p(t_k \mid t_1, t_2, \ldots, t_{k-1}).$$ Forward LM.

$$p(t_1, t_2, \ldots, t_N) = \prod_{k=1}^{N} p(t_k \mid t_{k+1}, t_{k+2}, \ldots, t_N).$$ Backward LM.

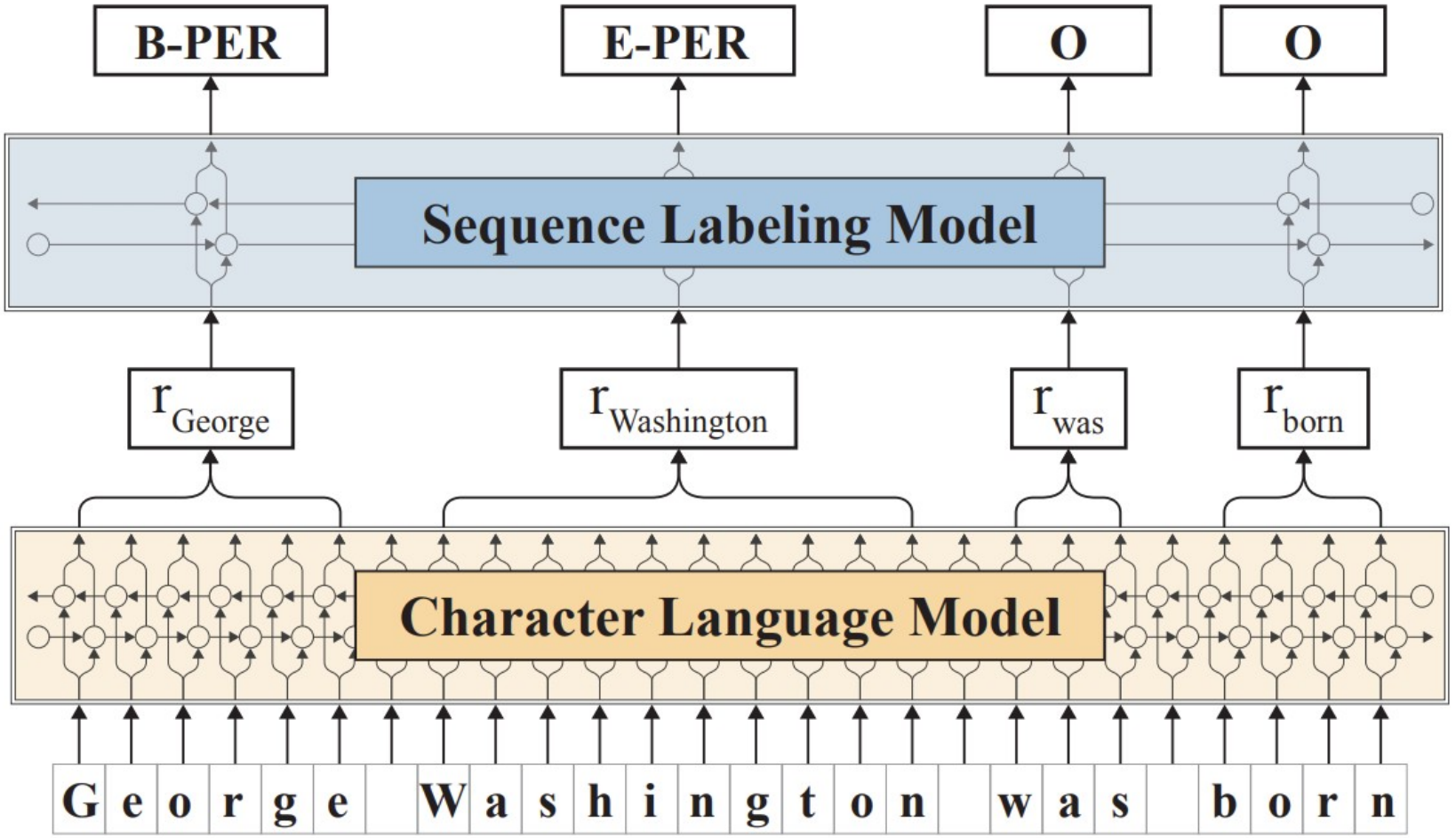
$$\mathbf{w}_i^{CharLM} := \begin{bmatrix} \mathbf{h}^f_{t_{i+1}-1} \\ \mathbf{h}^b_{t_i-1} \end{bmatrix}$$



Dependencia del contexto

FLAIR

Luego usan los *string embeddings* en NER:



Dependencia del contexto

Embeddings basados en modelos de lenguaje (ELMo)

Idea similar a FLAIR pero con multicapa y modelos basados en palabras:

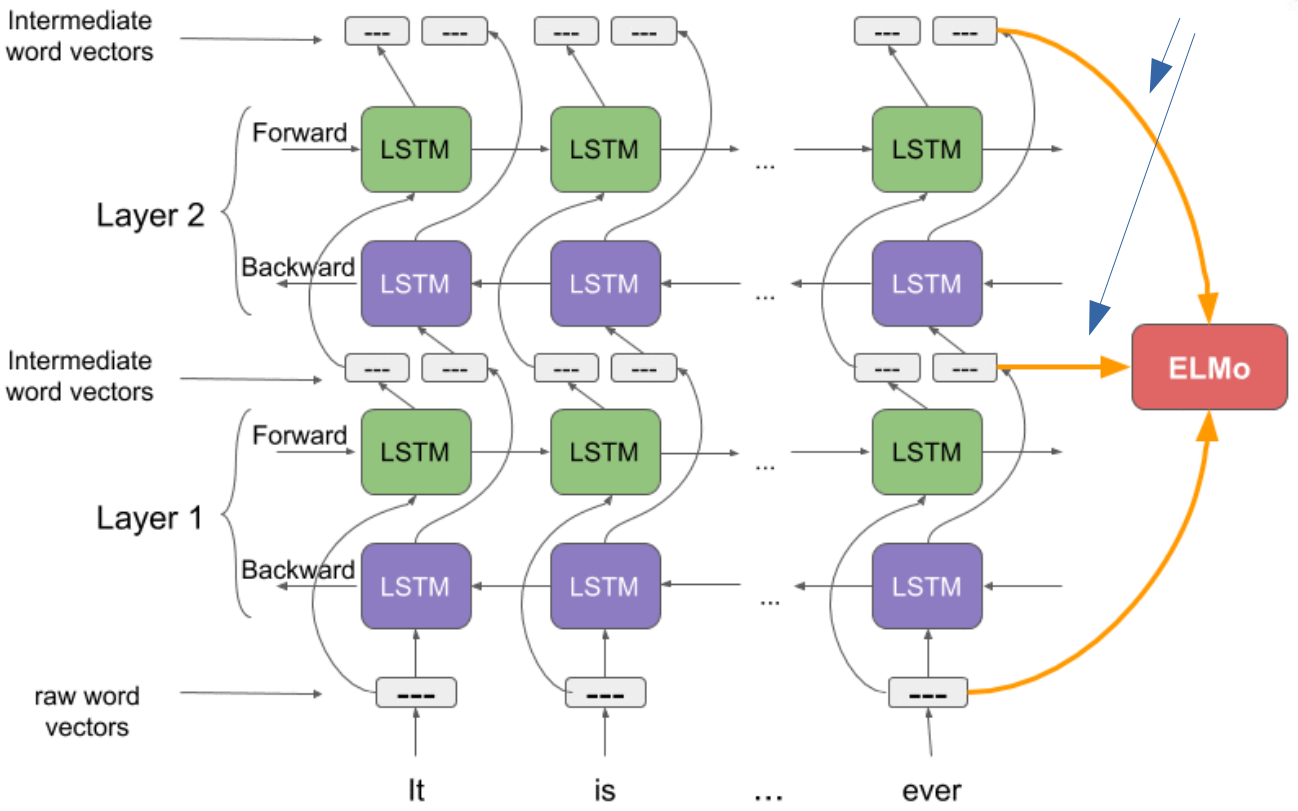
Parámetros de la representación

Parámetros de las LSTM

$$R_k = \{\mathbf{x}_k^{LM}, \overrightarrow{\mathbf{h}}_{k,j}^{LM}, \overleftarrow{\mathbf{h}}_{k,j}^{LM} \mid j = 1, \ldots, L\}$$
$$= \{\mathbf{h}_{k,j}^{LM} \mid j = 0, \ldots, L\},$$

Para representar a t_k

Arquitectura:



Dependencia del contexto

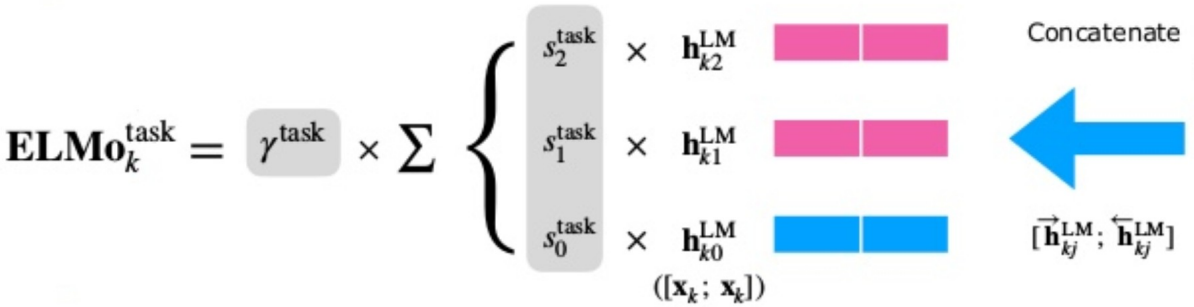
Embeddings basados en modelos de lenguaje (ELMo)

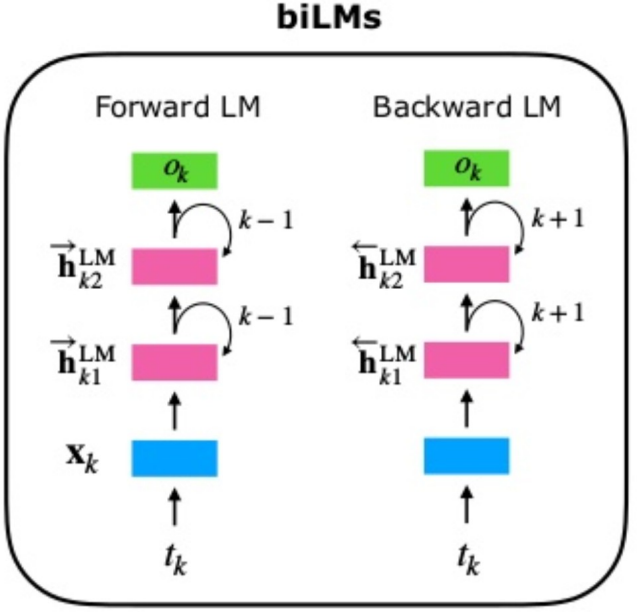
Vector de ELMo

ELMo: Se calculan pesos específicos para *downstream tasks* que producen un vector ad-hoc a la tarea.

$$\mathbf{ELMo}_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}^{LM} .$$

El parámetro $\mathbf{s}^{task}$ corresponde a pesos de la softmax. γ^{task} es un factor de escala que se introduce para efectos de optimización. En algunos casos se puede hacer normalización a nivel de capas antes de combinar los parámetros $\mathbf{h}_{k,j}^{LM}$.



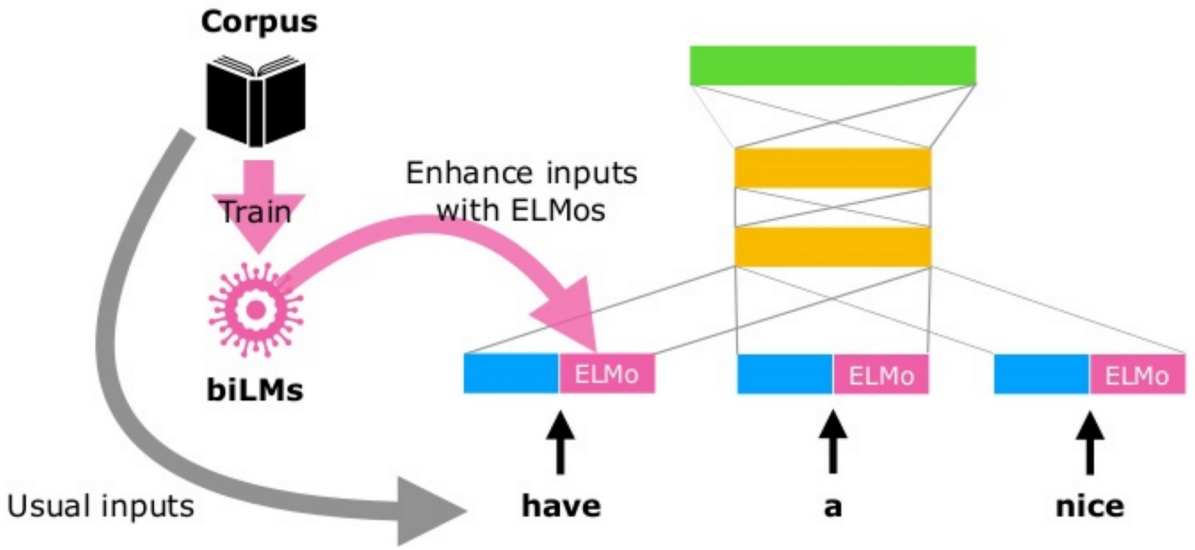


- En aprendizaje no supervisado, los vectores *h* se promedian.
- Al regularizar, el efecto que se produce es que la agregación aproxima al promedio de vectores *h*.

Dependencia del contexto

Embeddings basados en modelos de lenguaje (ELMo)

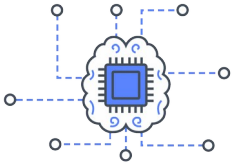
ELMo: se puede combinar con otras representaciones según la tarea específica:



El modelo final de ELMo (pre-entrenado) usa dos capas LSTM con vectores de dimensionalidad 512. Para la capa de representación, ELMo usa char n-grams y una capa de proyección de dimensionalidad 512.

ELMo fue entrenado sobre el **1 Billion Word Benchmark**.

Fine tuning para *downstream tasks* implica disminución en *perplexity* y mejoras en desempeño.



Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robin-son. 2014. One billion word benchmark for measuring progress in statistical language modeling. In INTERSPEECH.