

Taller de Procesamiento de Lenguaje Natural

# Modelos neuronales de etiquetado de secuencias

Escuela de Verano – EVIA 2023

David Vilares



UNIVERSIDADE DA CORUÑA



# Etiquetado de secuencias

Asignar a cada palabra de una oración una, y una única, etiqueta.

El	volcán	emitió	mucha	lava	durante	la	erupción
DET	NOUN	VERB	DET	NOUN	PREP	DEP	NOUN

Es la aproximación más simple de predicción estructurada.

Consiste en convertir texto plano en una representación con anotaciones sobre relaciones entre palabras o sus propiedades que pueden ser luego procesadas para extraer información.

# Etiquetado de secuencias

## Ventajas

Rápidos, potencialmente  $\Theta(n)$ ,  $n$  es el tamaño de la oración.

Fáciles de entender y de implementar (lo veremos hoy!).

No es un concepto limitado a PLN. También se etiquetan secuencias de ADN, por ejemplo.

# Etiquetado de secuencias

## Desventajas

Sirven para extraer información superficial de la oración.

Para extraer información y relaciones más complejas entre palabras, se emplean algoritmos de análisis de árboles y grafos.

# Tareas abordables como etiquetado de secuencias

Reconocimiento de entidades nombradas o *named-entity recognition* (NER).

Etiquetación morfológica o *part-of-speech (PoS) tagging*.

Etiquetación de roles semánticos o *semantic role labeling*.

*Multi-span question answering as sequence labeling*.

# Named-entity recognition (NER)

Reconocimiento de entidades nombradas.

Identificar información clave en texto escrito, habiendo definido previamente un conjunto de categorías (nombres de personas, localizaciones, organizaciones, etc).

Una entidad puede definirse como un concepto del que se habla frecuentemente en un dominio. Por ejemplo, nombres de actores en una película, o los platos en un menú de un restaurante.

# Named-entity recognition (NER)

Juan	lava	su	coche	rojo
B-PERSON	O	O	O	O
Barack	Obama	nació	en	Hawái
B-PERSON	I-PERSON	O	O	B-LOC

# Named-entity recognition (NER)

Suelen anotarse con el criterio de anotación BIO:

B: Indica que con esa palabra comienza una entidad.

I: Indica que con esa palabra continúa el reconocimiento de una entidad que se abrió anteriormente (etiquetando una palabra previa con una B).

O: Etiqueta usada para identificar aquellas palabras que no aportan información relevante para el dominio.

```
2   B-Rating
start I-Rating
restaurants O
with   O
inside B-Amenity
dining I-Amenity
```

# Aplicaciones de named-entity recognition (NER)

The screenshot displays a Named Entity Recognition (NER) interface. At the top, there is a legend bar with colored boxes and labels: 'Person' (blue, 'p'), 'Loc' (yellow, 'l'), 'Org' (black, 'o'), 'Event' (green, 'e'), 'Date' (red, 'd'), and 'Other' (purple, 'z'). Below the legend, a text snippet is shown with several entities highlighted in colored boxes: 'Barack Hussein Obama II' (blue), 'August 4, 1961' (red), 'American' (purple), 'the United States' (yellow), 'January 20, 2009' (red), 'January 20, 2017' (red), 'Democratic Party' (black), 'African American' (purple), 'United States Senator' (purple), 'Illinois' (yellow), and 'Illinois State Senate' (black). Each highlighted entity has a small asterisk icon to its right.

Extraída de: <https://www.analyticsvidhya.com/blog/2021/11/a-beginners-introduction-to-ner-named-entity-recognition/>

# Aplicaciones de named-entity recognition (NER)



## La Bombilla

Rúa Torreiro, 6, A Coruña

✎ Escribir una reseña

4,3 ★★★★★ 5.767 reseñas

Las reseñas no se verifican. ⓘ

Los usuarios suelen mencionar

- Todas
- emblemático 39
- plástico 34
- mítico 30
- tradición 28
- esencia 14
- caldo gallego 11
- estrella galicia 11
- milanesa 10
- servilletero 9
- clásico 8

Ordenar por

- Más relevantes
- Más recientes
- Más alta
- Más baja



5.753 fotos

★★★★★ Hace 8 meses

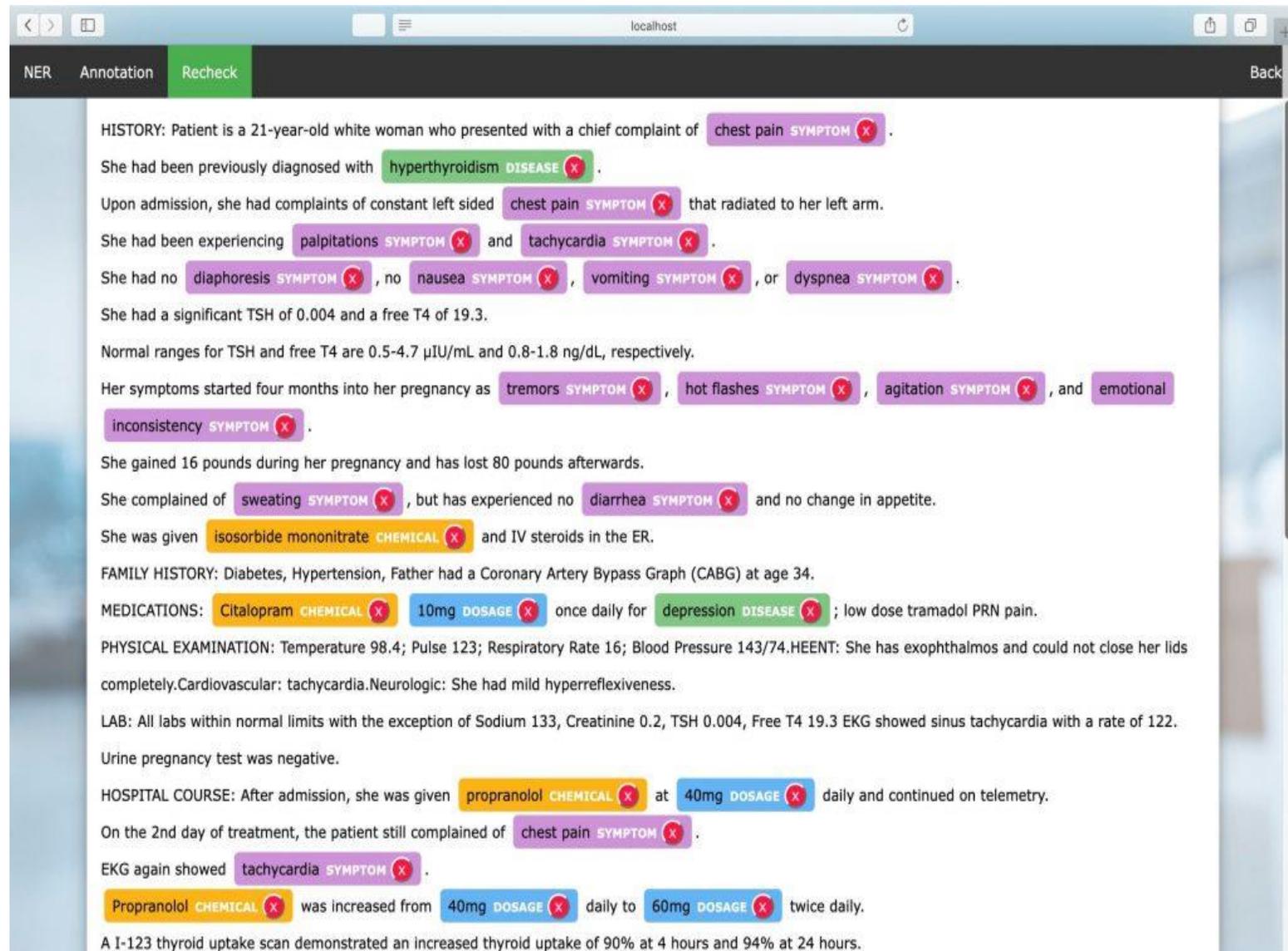
Fue una visita rápida de cañas y tapas. Una lástima que las tapas de milanesa y calamares estaban frías. Rico sabor pero mal la temperatura. A los callos les faltaba sabor y tenían muchas partes de cerdo que eran solo grasa. La ensaladilla estaba muy rica.



👍 2

Extraída de Google Reviews

# Aplicaciones de named-entity recognition (NER)



The screenshot displays a web-based NER application interface. At the top, there are navigation tabs for 'NER', 'Annotation', and 'Recheck', with 'Recheck' currently selected. The main content area shows a medical history text with several entities highlighted in colored boxes and labeled with terms like SYMPTOM, DISEASE, and CHEMICAL. The text is as follows:

HISTORY: Patient is a 21-year-old white woman who presented with a chief complaint of chest pain SYMPTOM . She had been previously diagnosed with hyperthyroidism DISEASE . Upon admission, she had complaints of constant left sided chest pain SYMPTOM that radiated to her left arm. She had been experiencing palpitations SYMPTOM and tachycardia SYMPTOM . She had no diaphoresis SYMPTOM , no nausea SYMPTOM , vomiting SYMPTOM , or dyspnea SYMPTOM . She had a significant TSH of 0.004 and a free T4 of 19.3. Normal ranges for TSH and free T4 are 0.5-4.7  $\mu$ IU/mL and 0.8-1.8 ng/dL, respectively. Her symptoms started four months into her pregnancy as tremors SYMPTOM , hot flashes SYMPTOM , agitation SYMPTOM , and emotional inconsistency SYMPTOM . She gained 16 pounds during her pregnancy and has lost 80 pounds afterwards. She complained of sweating SYMPTOM , but has experienced no diarrhea SYMPTOM and no change in appetite. She was given isosorbide mononitrate CHEMICAL and IV steroids in the ER. FAMILY HISTORY: Diabetes, Hypertension, Father had a Coronary Artery Bypass Graph (CABG) at age 34. MEDICATIONS: Citalopram CHEMICAL 10mg DOSAGE once daily for depression DISEASE ; low dose tramadol PRN pain. PHYSICAL EXAMINATION: Temperature 98.4; Pulse 123; Respiratory Rate 16; Blood Pressure 143/74. HEENT: She has exophthalmos and could not close her lids completely. Cardiovascular: tachycardia. Neurologic: She had mild hyperreflexiveness. LAB: All labs within normal limits with the exception of Sodium 133, Creatinine 0.2, TSH 0.004, Free T4 19.3 EKG showed sinus tachycardia with a rate of 122. Urine pregnancy test was negative. HOSPITAL COURSE: After admission, she was given propranolol CHEMICAL at 40mg DOSAGE daily and continued on telemetry. On the 2nd day of treatment, the patient still complained of chest pain SYMPTOM . EKG again showed tachycardia SYMPTOM . Propranolol CHEMICAL was increased from 40mg DOSAGE daily to 60mg DOSAGE twice daily. A I-123 thyroid uptake scan demonstrated an increased thyroid uptake of 90% at 4 hours and 94% at 24 hours.

Extraída de: <https://www.persistent.com/blogs/building-named-entity-recognition-models-for-healthcare/>

# Modelos de etiquetado de secuencia

Metodología estándar para la creación de los modelos.

Fase entrenamiento/desarrollo:

- Un conjunto de entrenamiento (train.txt) y otro de desarrollo (dev.txt), comúnmente en texto plano, con anotaciones a nivel de palabra.
- Un modelo supervisado capaz de aprender a partir de dichos datos.

Evaluación:

- Un conjunto de test (test.txt) que el modelo no haya visto antes, para estimar su rendimiento en entornos reales.

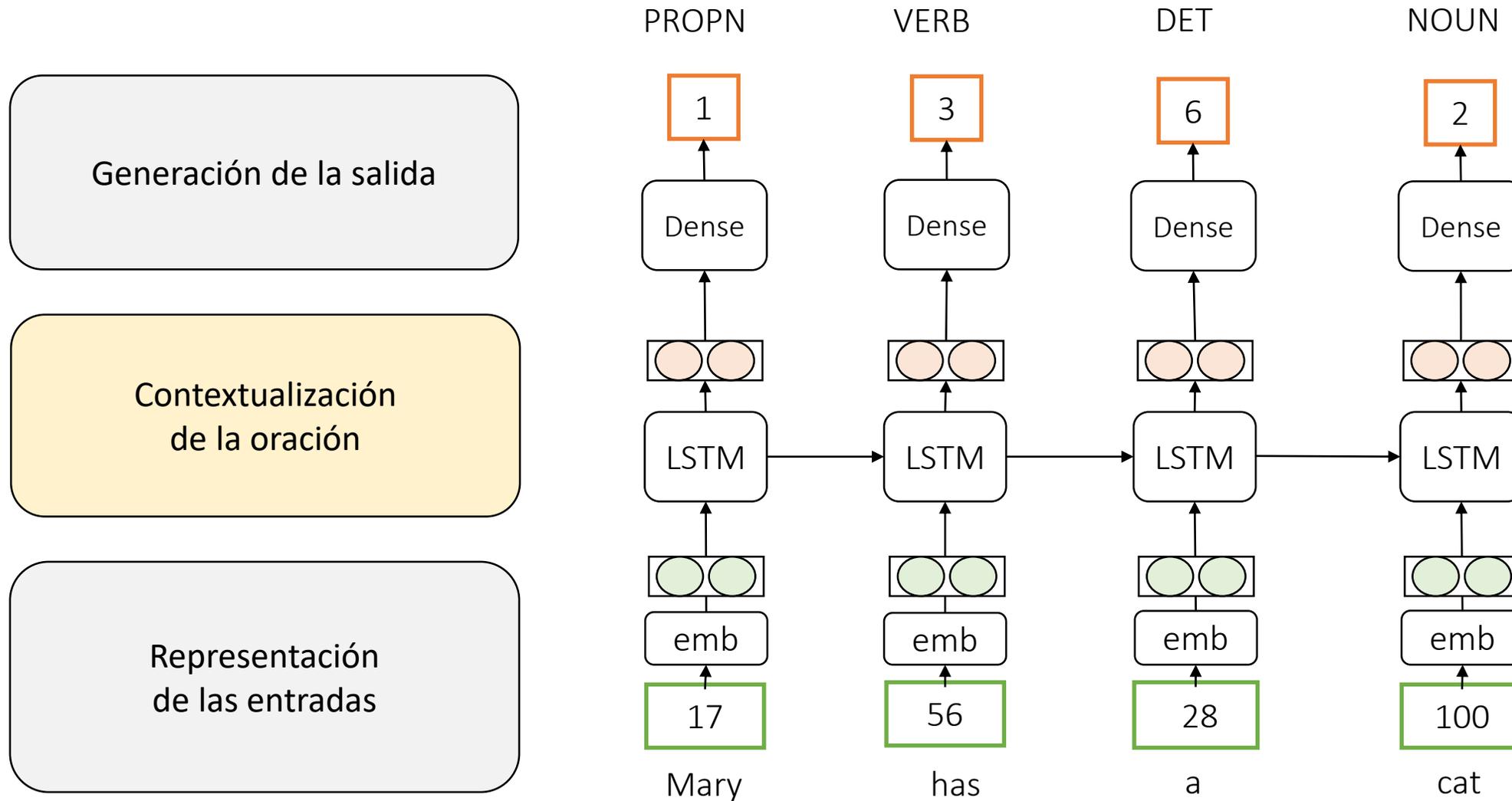
# Modelos de etiquetado de secuencia

**Snippets y feed-forward networks:** Modelos con contexto limitado.

**Long short-term memory networks (LSTMs):** Modelos que pueden recordar u olvidar información sobre secuencias largas, lo que permite capturar dependencias lejanas en los datos.

**Transformers:** Modelos con una excelente capacidad para capturar, mediante técnicas de self-attention, el contexto relevante para una palabra (independientemente de la distancia a la tarea objetivo) y para transferir conocimiento una vez han sido pre-entrenados como modelos de lenguaje sobre grandes cantidades de texto (BERT, RoBERTa, GPT, LLaMa).

# Long short-term memory networks



# Long short-term memory networks – Representación de las entradas

Mary has a cat

PROPN VERB DET NOUN



17 56 28 100

1 3 6 2

Spiderman exists

PROPN VERB



7 324

1 3

I am real

PRONOUN VERB ADJ



98 76 3

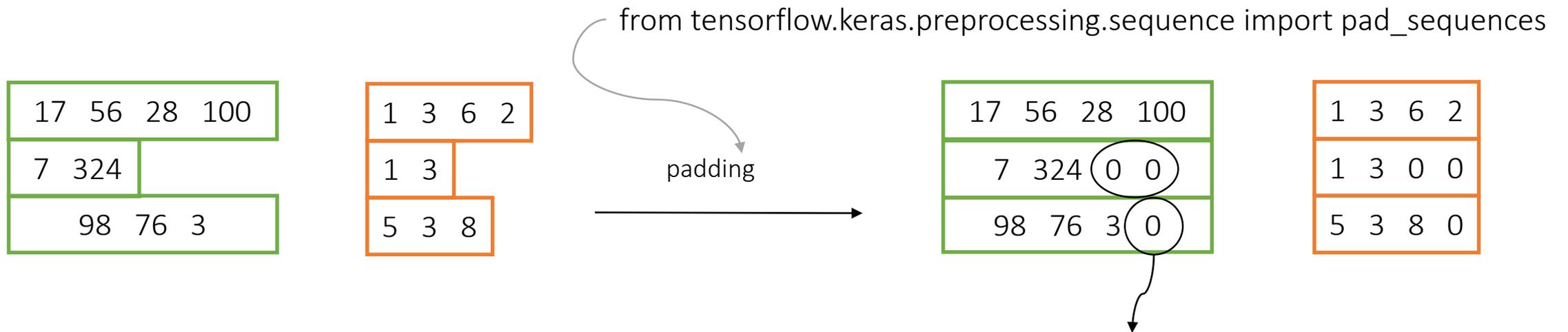
5 3 8

# Long short-term memory networks – Representación de las entradas

Tendremos oraciones con diferentes longitudes.

Todas las oraciones en un batch de entrenamiento deben tener la misma longitud para que puedan ser enviadas a un modelo de Keras

Solución: incluir padding.



Debemos ignorar estos elementos del cálculo de la loss usando `mask_zero=True` cuando definamos la capa de Embeddings

# Long short-term memory networks – Representación de las entradas

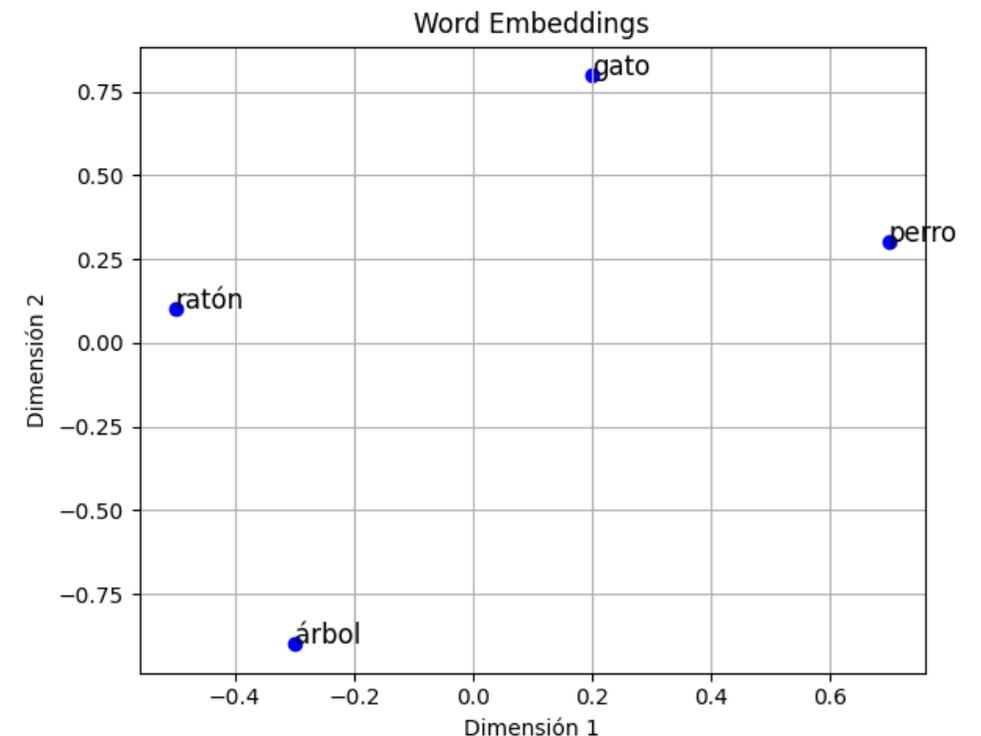
*Embeddings*: representación de una palabra (elemento discreto) como un vector de baja dimensionalidad. Palabras que ocurren en contextos similares tendrán *embeddings* similares.

"gato": [0.2, 0.8]

"perro": [0.7, 0.3]

"ratón": [-0.5, 0.1]

"árbol": [-0.3, -0.9]



# Long short-term memory networks – Representación de las entradas

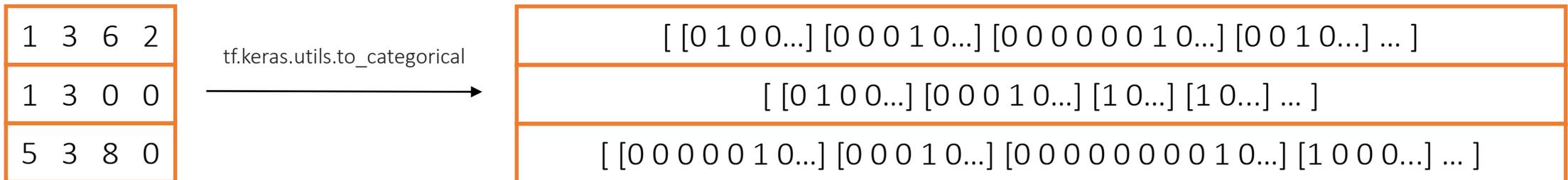
**Embeddings inicializadas aleatoriamente** y entrenadas conjuntamente con el resto de la red para la tarea objetivo. Palabras con contextos similares deberían tener embeddings cercanas al final del entrenamiento.

**Embeddings estáticas:** calculadas previamente (word2vec, GloVe) y enchufables a la red. Una misma palabra tiene siempre la misma embedding, independiente del contexto en el que aparezca.

Embeddings **contextualizadas:** calculadas a través de un modelo de lenguaje que ha sido pre-entrenado (BERT, GPT) y que se conectan al resto de la red.

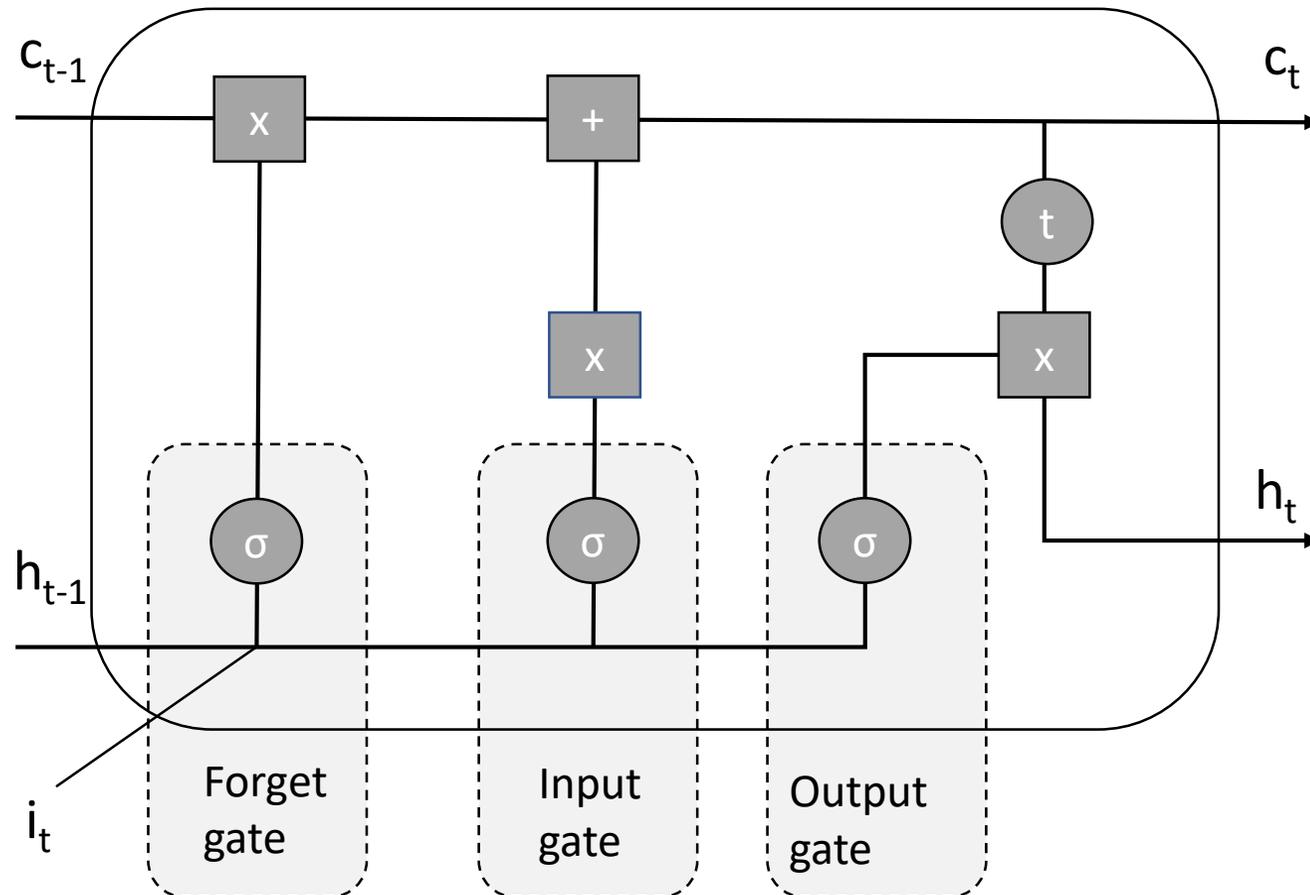
# Long short-term memory networks – Representación de etiquetas de salida

Necesitaremos transformar las etiquetas de salida (inicialmente representadas como una cadena de caracteres) en un one-hot vector que identifica la clase que se debe predecir.



# Long short-term memory networks

## Celda LSTM



**Cell state:** La "memoria" de la LSTM. En cada *timestep* (cada palabra en el caso de problemas de PLN) se actualiza en base a la forget y la input gate. Se almacena en  $c_t$ .

**Input gate:** Determina que valores de la entrada en un *timestep* dado añadir al cell state.

**Forget gate:** Determina que valores eliminar del cell state para un *timestep* dado.

**Output gate:** Determina la cantidad de cell state que transmitir a la salida, que se manifiesta en el hidden output state ( $h_t$ ).

# Long short-term memory networks – Definición de la red

Capa de entrada: [https://www.tensorflow.org/api\\_docs/python/tf/keras/Input](https://www.tensorflow.org/api_docs/python/tf/keras/Input)

Capa de embeddings: [https://www.tensorflow.org/api\\_docs/python/tf/keras/layers/Embedding](https://www.tensorflow.org/api_docs/python/tf/keras/layers/Embedding)

Capa LSTM: [https://www.tensorflow.org/api\\_docs/python/tf/keras/layers/LSTM](https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM)

Capa bidireccional: [https://www.tensorflow.org/api\\_docs/python/tf/keras/layers/Bidirectional](https://www.tensorflow.org/api_docs/python/tf/keras/layers/Bidirectional)

Capa Dense: [https://www.tensorflow.org/api\\_docs/python/tf/keras/layers/Dense](https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dense)

Capa TimeDistributed [importante!]:  
[https://www.tensorflow.org/api\\_docs/python/tf/keras/layers/TimeDistributed](https://www.tensorflow.org/api_docs/python/tf/keras/layers/TimeDistributed)

# tf.keras.layers.TimeDistributed

Un wrapper para aplicar una capa a cada timestep de una entrada temporal (como por ejemplo una oración).

La dimension en el eje 1 es la que se considera como la dimension temporal.

En nuestro caso, la forma de la entrada para nuestro etiquetador es (batch\_size, max\_sentence\_length, word\_embedding\_size).

Dada una capa TimeDistributed, aplicará la capa 'envuelta' a todos los elementos de entrada de la secuencia.

# tf.keras.layers.TimeDistributed

En este caso usaremos la capa TimeDistributed para aplicar la capa de clasificación de salida Dense sobre la representación oculta de cada palabra.

Se aplicará a cada uno de los vectores de salida generados por la LSTM.

Permitirá obtener las etiquetas de salida para cada token de entrada de manera transparente.

# tf.keras.layers.TimeDistributed

model = ...

...

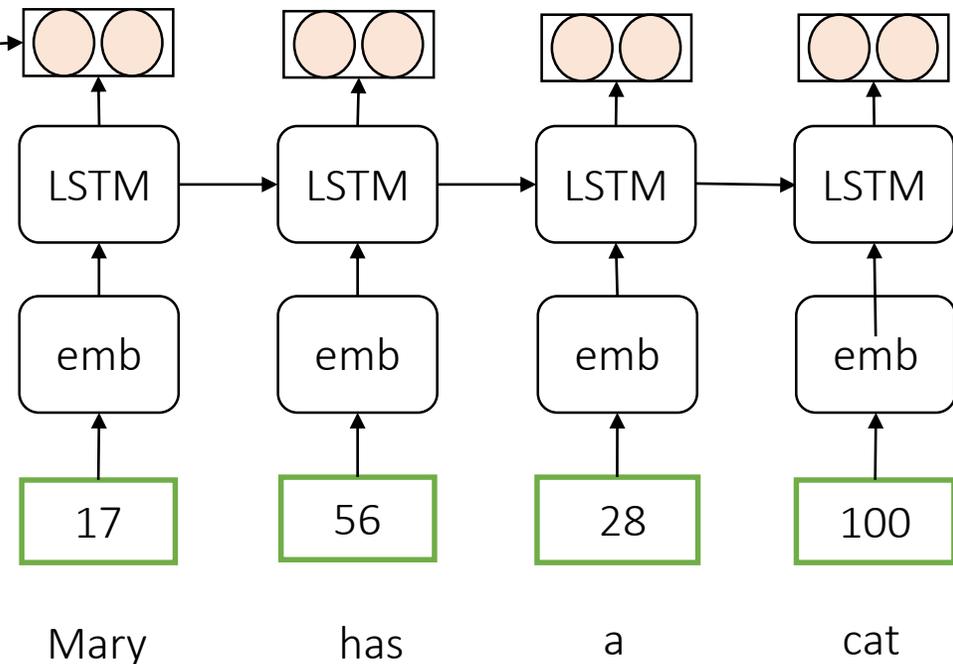
Añadir la capa de entrada

Añadir la capa de Embeddings

Añadir la capa de LSTMs

...

La salida de la LSTM debe ser un vector para cada palabra (chequear el parámetro `return_sequences` de la capa LSTM).



# tf.keras.layers.TimeDistributed

model = ...

...

Añadir la capa de entrada

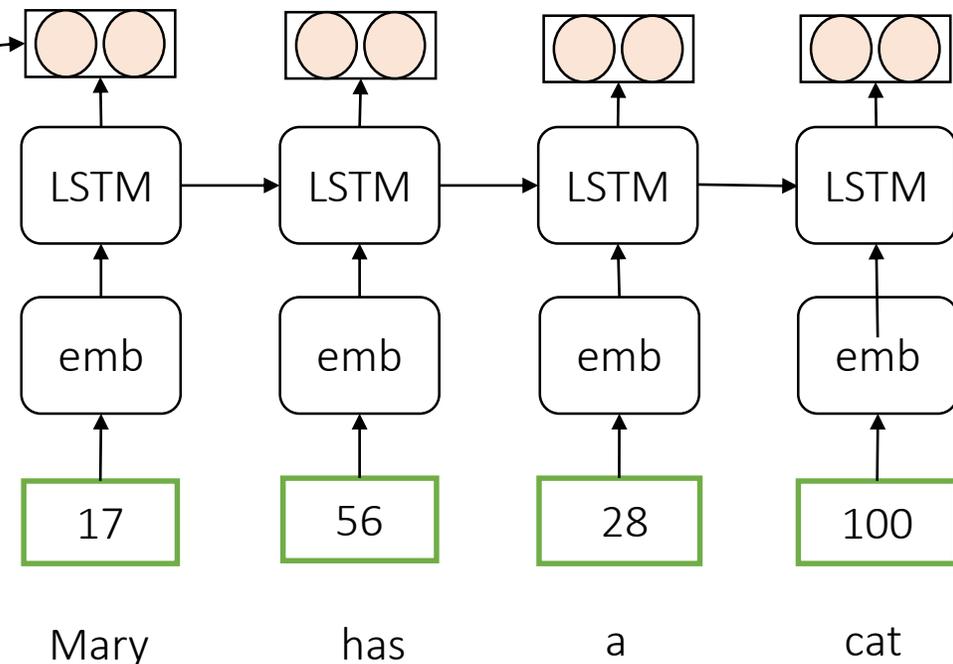
Añadir la capa de Embeddings

Añadir la capa de LSTMs

...

La salida de la LSTM debe ser un vector para cada palabra (chequear el parámetro `return_sequences` de la capa LSTM).

Sin embargo, la capa Dense (redes feed-forward) que viene a continuación no está pensada para ser aplicada sobre secuencias, ¿cómo podemos obtener entonces fácilmente la etiqueta para todos los elementos de entrada 'de una vez'? Usando el wrapper `TimeDistributed`, que hace justamente eso.



# tf.keras.layers.TimeDistributed

model = ...

...

Añadir la capa de entrada

Añadir la capa de Embeddings

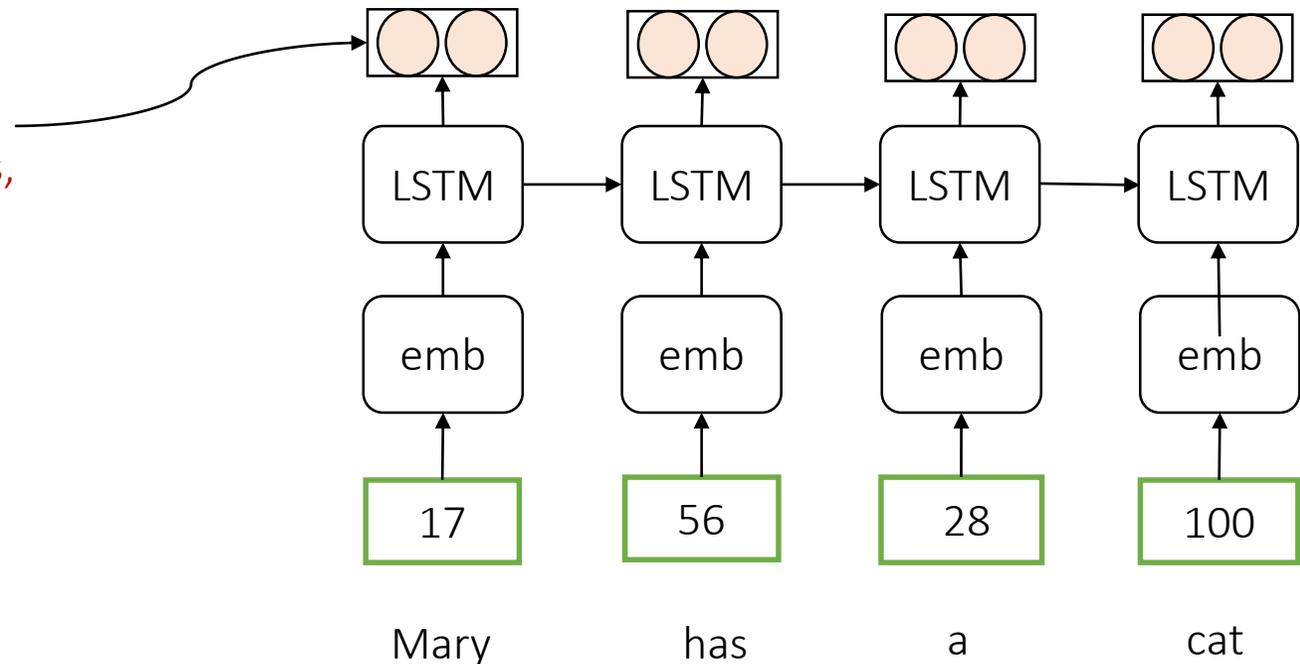
Añadir la capa de LSTMs

...

```
model.add(TimeDistributed(Dense(nlabels,  
activation='softmax')))
```

...

model.fit(...)



# tf.keras.layers.TimeDistributed

model = ...

...

Añadir la capa de entrada

Añadir la capa de Embeddings

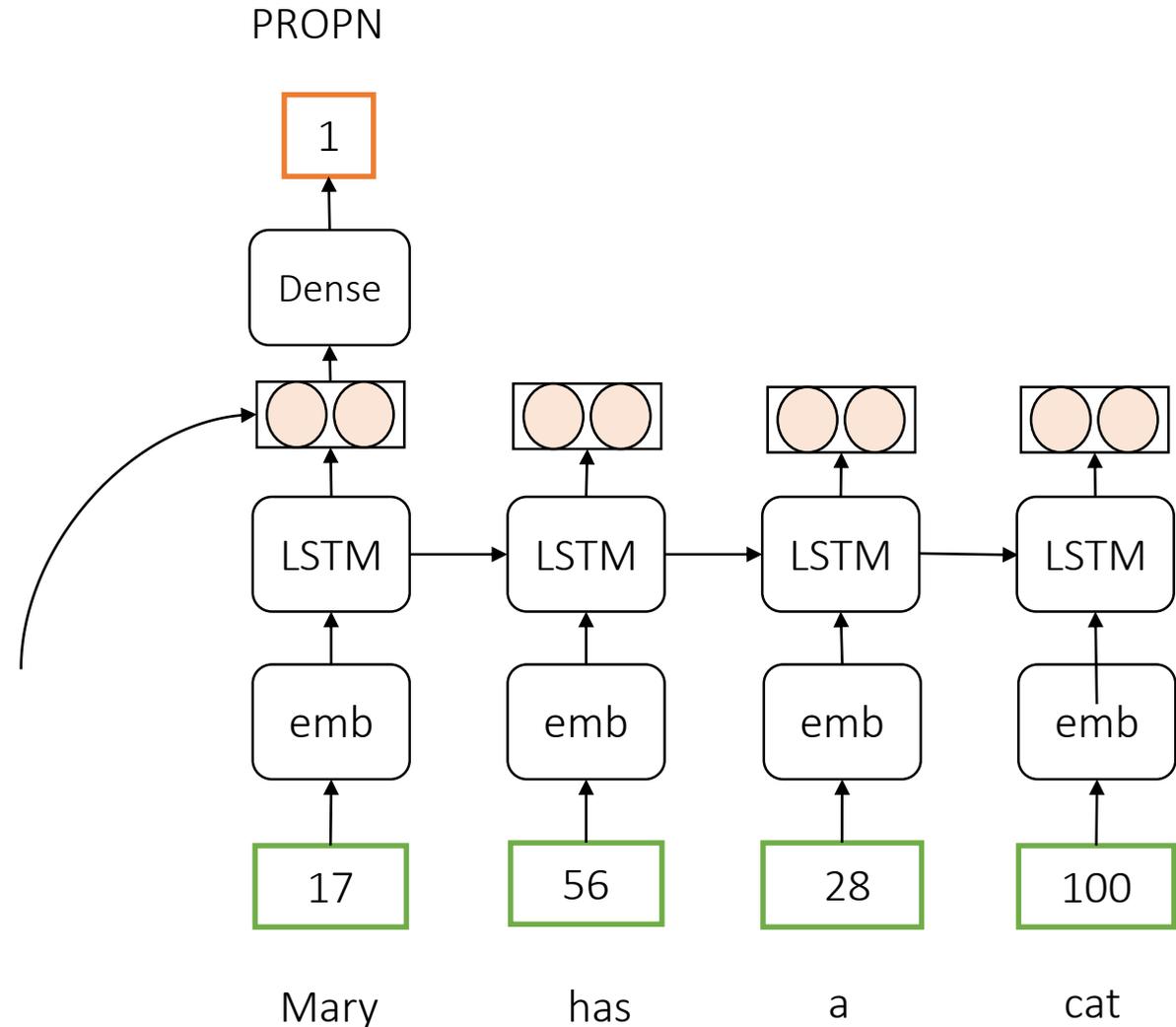
Añadir la capa de LSTMs

...

```
model.add(TimeDistributed(Dense(nlabels,  
activation='softmax')))
```

...

model.fit(...)



# tf.keras.layers.TimeDistributed

model = ...

...

Añadir la capa de entrada

Añadir la capa de Embeddings

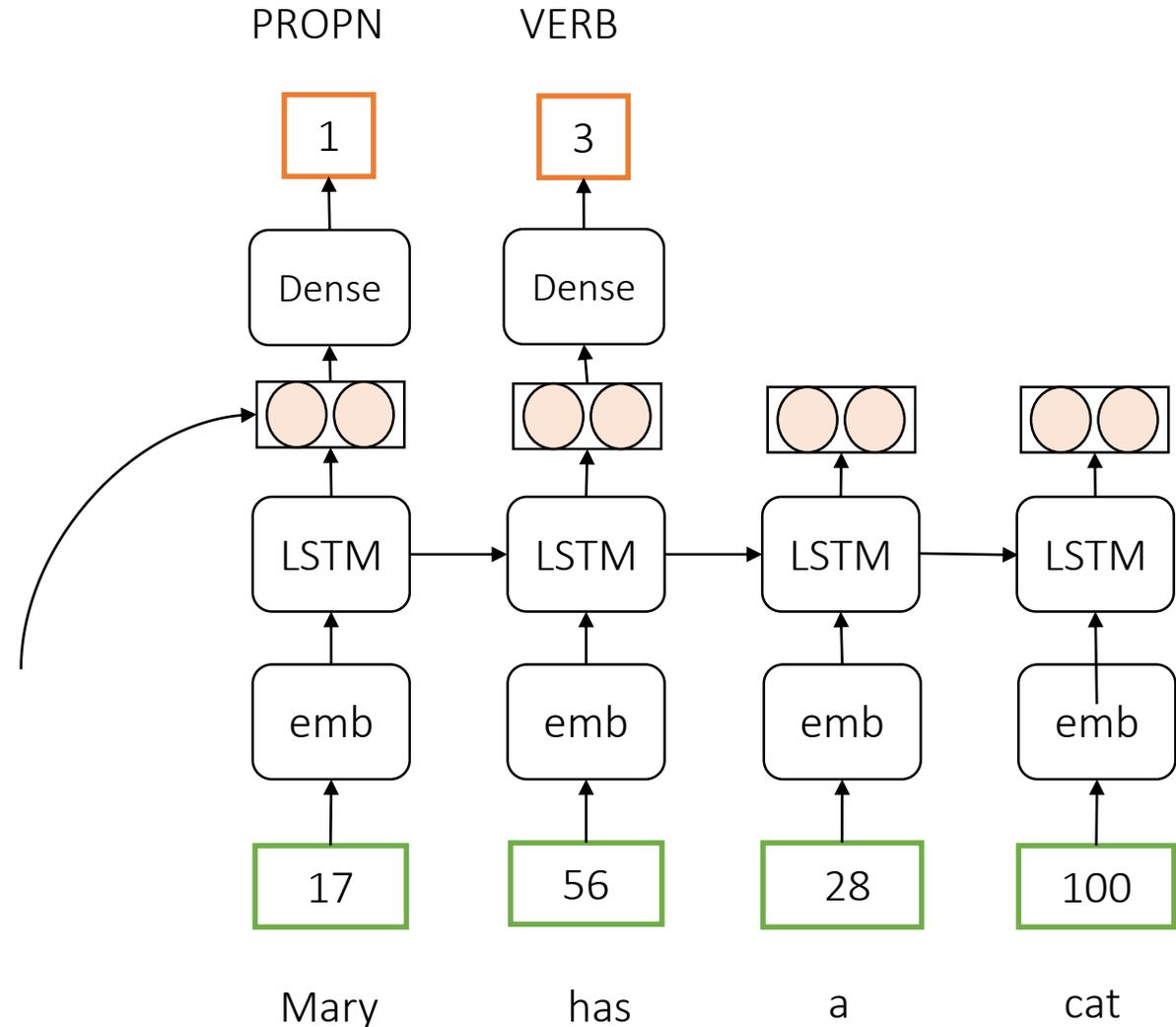
Añadir la capa de LSTMs

...

```
model.add(TimeDistributed(Dense(nlabels,  
activation='softmax')))
```

...

model.fit(...)



# tf.keras.layers.TimeDistributed

model = ...

...

Añadir la capa de entrada

Añadir la capa de Embeddings

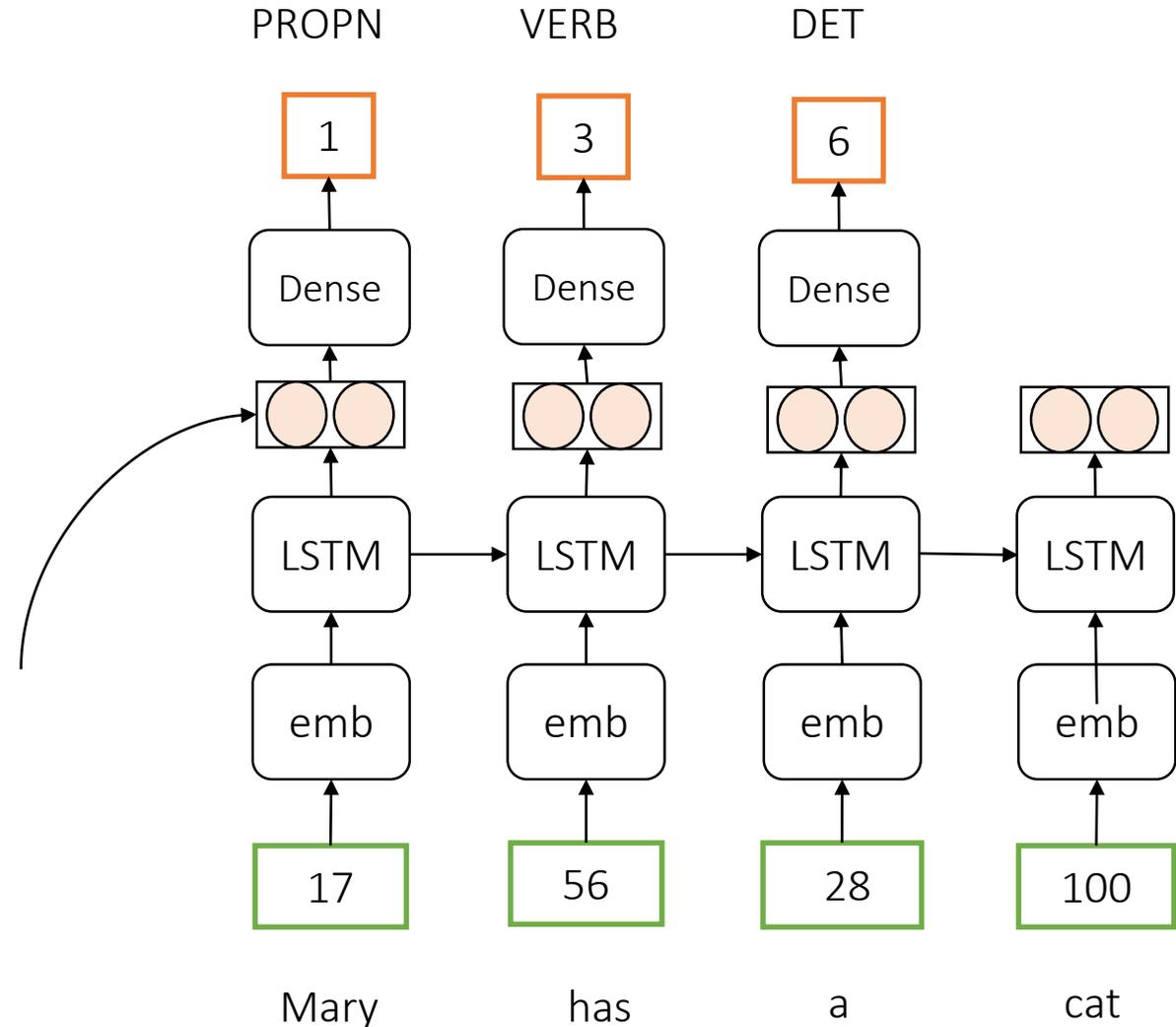
Añadir la capa de LSTMs

...

```
model.add(TimeDistributed(Dense(nlabels,  
activation='softmax')))
```

...

model.fit(...)



# tf.keras.layers.TimeDistributed

model = ...

...

Añadir la capa de entrada

Añadir la capa de Embeddings

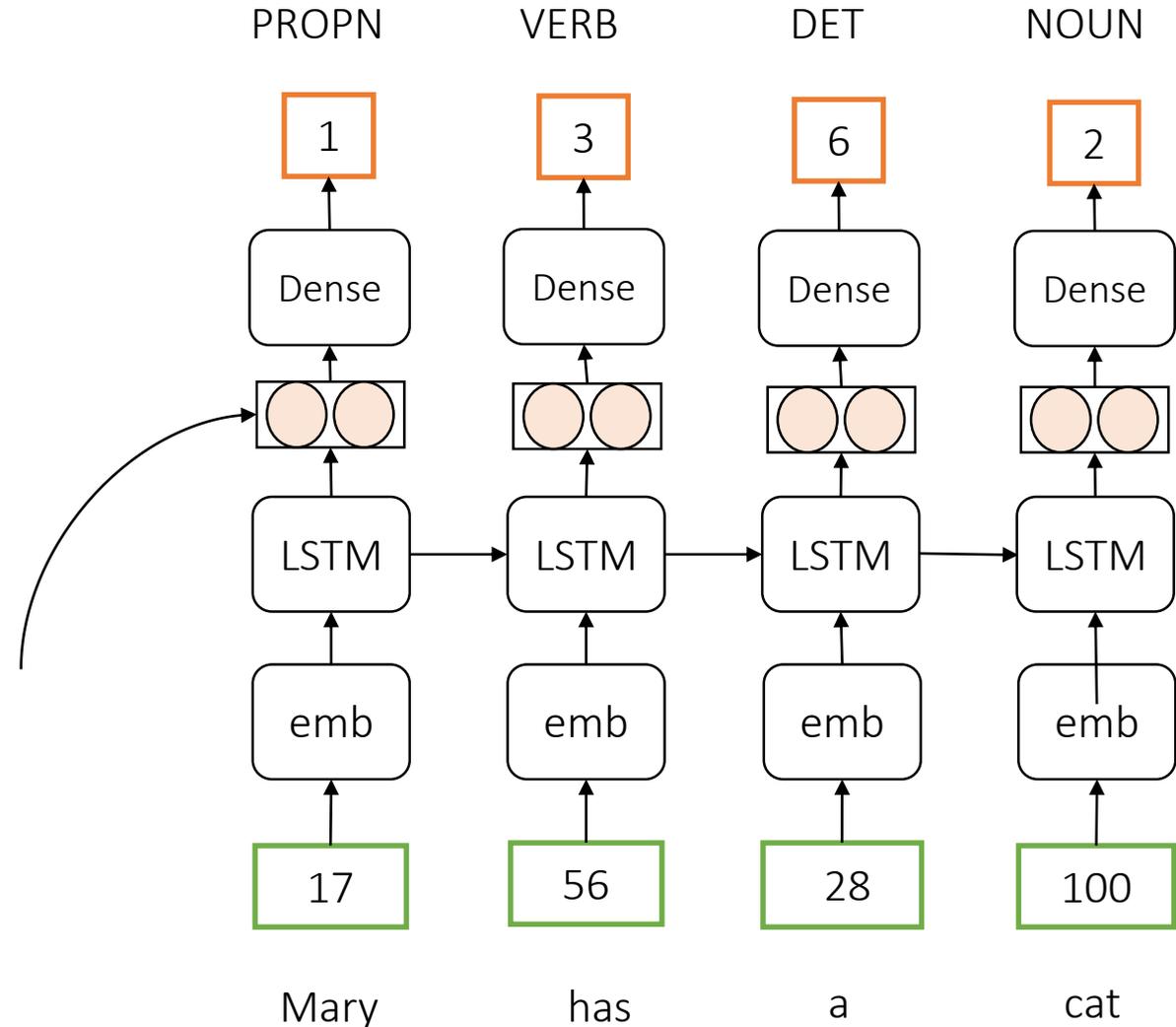
Añadir la capa de LSTMs

...

```
model.add(TimeDistributed(Dense(nlabels,  
activation='softmax')))
```

...

model.fit(...)



# Parámetros de entrenamiento

Es un problema de clasificación multi-clase.

Optimizaremos el modelo con `categorical_crossentropy` loss:  
[https://www.tensorflow.org/api\\_docs/python/tf/keras/metrics/categorical\\_crossentropy](https://www.tensorflow.org/api_docs/python/tf/keras/metrics/categorical_crossentropy)

Alternativamente a `categorical_crossentropy` loss también podemos usar una `sparse categorical_crossentropy` loss (en ese caso no necesitaremos la función `to_categorical`):  
[https://www.tensorflow.org/api\\_docs/python/tf/keras/losses/SparseCategoricalCrossentropy](https://www.tensorflow.org/api_docs/python/tf/keras/losses/SparseCategoricalCrossentropy)

Como optimizador, podemos usar SGD, Adam, etc.