Manuel Mager

Redes Neuronales para datos secuenciales II: Modelos sequence-to-sequence

July 2, 2019

Institute for Natural Language Processing
University of Stuttgart

Sequence to sequence

Encoder-Decoder

La idea básica es que usamos un modelo de lenguaje RNN para generar la salida y. El estado inicial de este modelo de lenguaje será el calculado dado otro RNN sobre la frase de origen.

Vanilla Encoder-Decoder

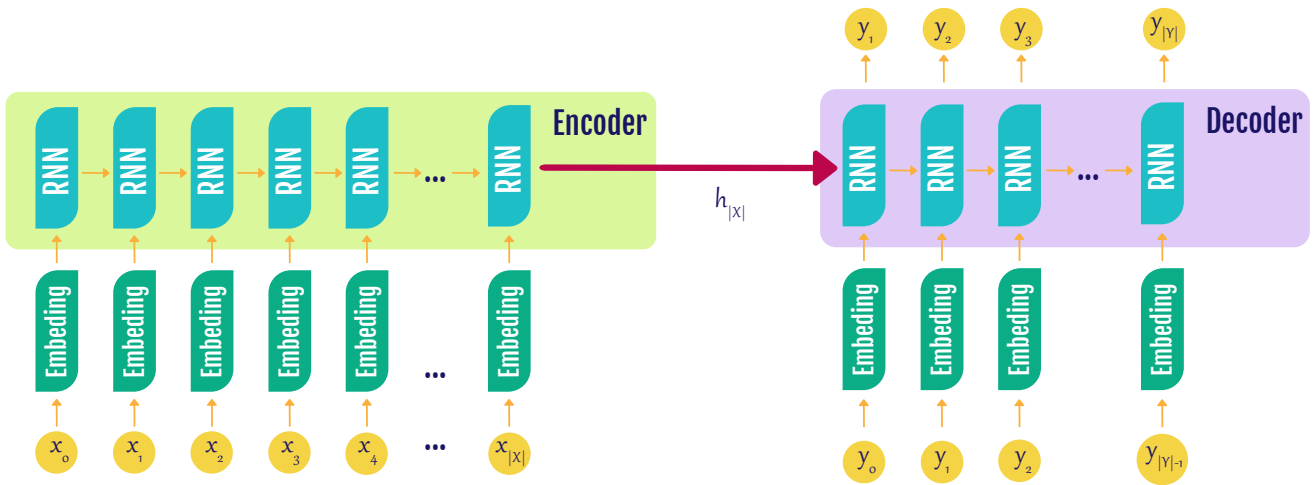

Figure: Encoder decoder

Modelo Encoder-Decoder

$$\mathrm{h}_t^{(x)} = \begin{cases} \mathrm{RNN}^{(x)}(embed(x_t), h_{t-1}^{(x)}) & t > 1 \\ 0 & \text{otherwise.} \end{cases}$$

$$\mathrm{h}_t^{(y)} = \begin{cases} \mathrm{RNN}^{(y)}(\mathrm{embed}(y_{t-1}, h_{t-1}^{(y)})) & t > 1 \\ h_{|X|}^{(f)} & \text{otherwise} \end{cases}$$

$$\mathrm{p}_t^{(y)} = \mathrm{softmax}(W_{hs} h_t^{(y)} + b_s)$$

Generación de salida

1. **Random Sampling** Genera una salida aleatoria de las probabilidades $\hat{Y} \sim P(Y|X)$
2. **1-best Search** Encuentra Y que maximiza $P(Y|X)$, tal que $\hat{Y} = \text{argmax}_Y P(Y|X)$.
3. **n-best Search** Encuentra las n salidas con la mayor propabilidad de acuerdo a $P(E|F)$.

Random Sampling

- Se utiliza en casos en que se desea obtener una variedad de salidas.
- Ejemplo en un chatbot se desea variar las respuestas en vez de siempre responder lo mismo.
- Existen algoritmos que nor permiten restringuir la aleatoriedad conservando contexto, como ancestral sampling.
- Podemos ademas guiarnos por la probabilidad de toda la salida, denotada por $P(\hat{Y}|X) = \prod_t^{|\hat{Y}|} P(\hat{y}_t|X, \hat{Y}_{t-1})$.

Algoritmo básico de búsqueda

$i \leftarrow 10$
for t in 1 to $|X|$ **do**
  Calculate $h_t^{(x)}$
end for
$\hat{y} \leftarrow$ <BOS>
$t \leftarrow 0$
while $\hat{y} \neq$ <EOS> **do**
  Calculate $h_t^{(y)}$ y $p_t^{(y)}$ de $\hat{y}_{t-1}$
  Muestrea $\hat{y}_t$ de acuerdo a $p_t^{(y)}$
end while

Greedy 1-best Search

La mayoría de aplicaciones van a requerir este algoritmo de generación. la idea es buscar por el grafo probabilístico con búsqueda voraz. Para cada tiempo se escoge la mejor probabilidad. Esto lo podemos ver cambiando el algoritmo pasado:

$i \leftarrow 10$
for t in 1 to $|X|$ **do**
  Calculate $h_t^{(x)}$
end for
$\hat{y} \leftarrow$ <BOS>
$t \leftarrow 0$
while $\hat{y} \neq$ <EOS> **do**
  Calculate $h_t^{(y)}$ y $p_t^{(y)}$ de $\hat{y}_{t-1}$
  $\hat{y}_t = \text{argmax}_y p_t^{(y)}$ de acuerdo a $p_t^{(y)}$
end while

Beam search (n-best search)

Aproxima la solución al problema de encontrar la máxima probabilidad de $\hat{Y}$ de todo el enunciado.

Atención!

Problemas con el modelo Encoder-Decoder

- El modelo puede modelar perfectamente funciones del tipo $P(y_t|X, y_{t-1})$. Sin embargo, para lograr el óptimo rendimiento necesita con una cantidad de datos no existente en la realidad.
- No funciona muy bien modelando dependencias largas.
- Las representaciones intermedias h_t son del mismo tamañano del encoder y el decoder, a pesar provenir de tamaños de sequencias diferentes.

Encoder-Decoder con Atención

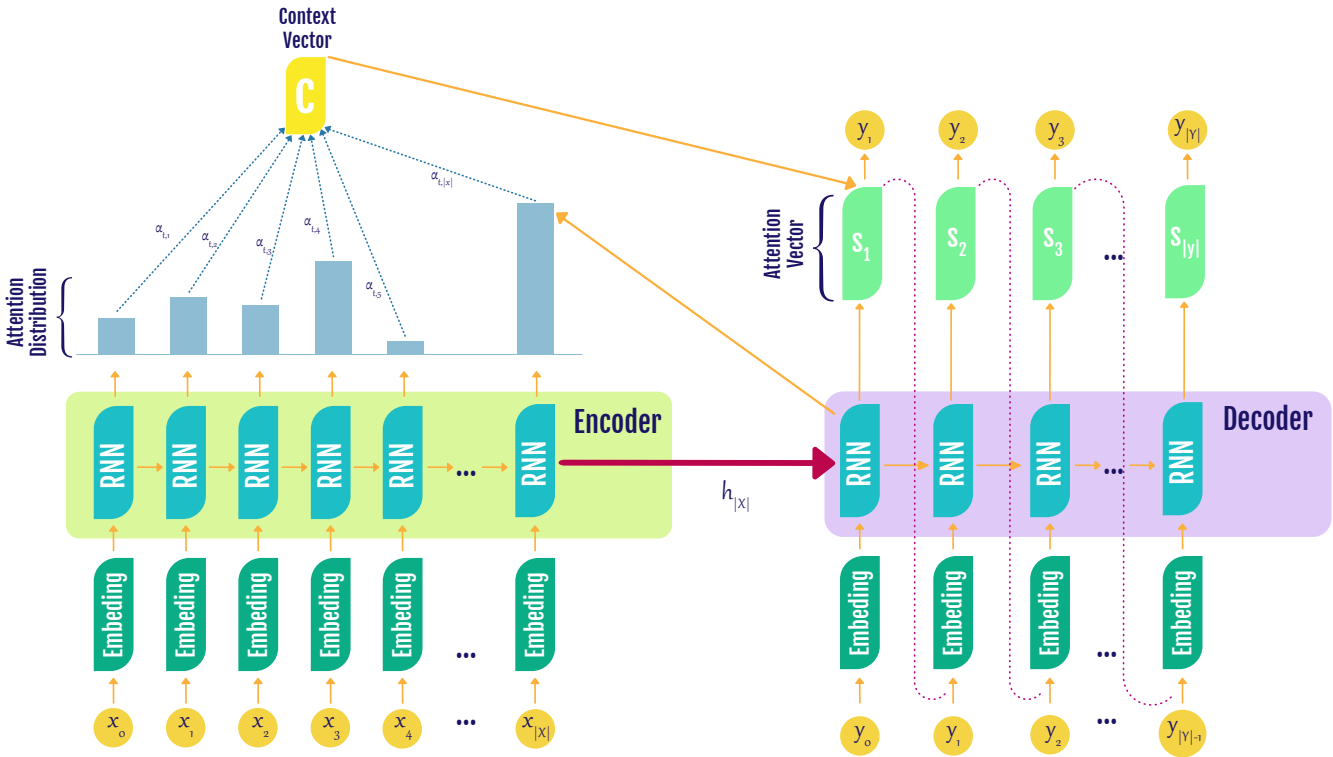

Figure: Encoder decoder con atención

Explicando el mecanismo de atención

Introducido por Bahdanau et al. (2014). Dado las ecuaciones del modelo encoder-decoder, definimos una nueva capa en el decoder:

$$s_i = f(h_{t-1}^{(y)}, y_{t-1}, c_t) \tag{1}$$

El vector de contexto c_i es computado como la suma de los pesos de las anotaciones de h_j.

$$c_i = \sum_{j=1}^{|X|} \alpha_{ij} h_j \tag{2}$$

Explicando...

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{|X|} \exp(e_{ik})} \tag{3}$$

donde

$$e_{ij} = a(h_{i-1}^{(y)}, h_j^{(x)}) \tag{4}$$

es un modelo de alineación que asigna valores sobre que tan bien las entradas alrededor de la posición j y las salidas alrededor de i corresponden. Los valoraciones son hechas basadas en el estado oculto del RNN s_{i-1} justo antes de emitir y_i y la anotacion de h_j de la secuencia de entrada.

Explicando...

Parametrizamos el modelo de alineación a como una capa feed forward.

$$e_{ij} = a(h_{i-1}^{(y)}, h_j^{(x)}) \tag{5}$$

$$= V\tanh(W_1 h_{i-1}^{(y)}, W_2 h_j^{(x)}) \tag{6}$$

representación gráfica

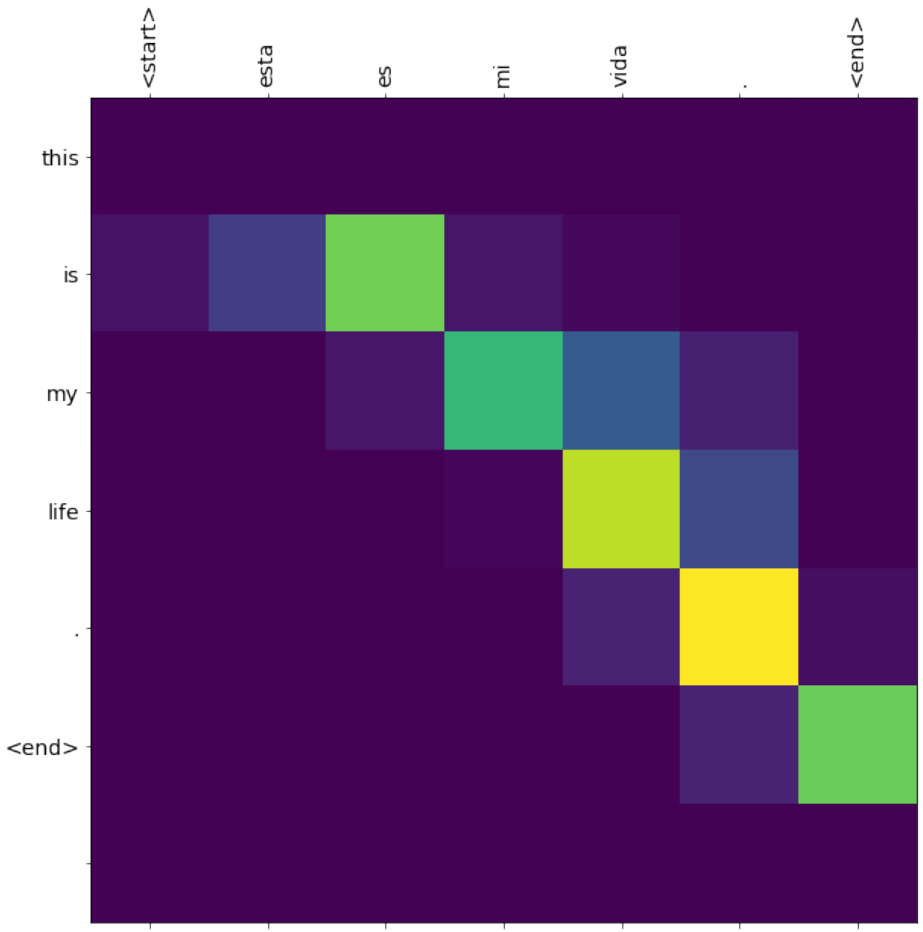

Figure: Attention matrix.

Mejorando modelos seq2seq

- Vocabulario Largo
- Traducción multi lingüe
- Introducir ruido artificial
- Entrenamiento multi tareas
- Datos artificiales
- Transfer learning

Atención

- **Hard attention** En el modelo de atención vimos el uso de una combinación suave de contenidos. Si se usara una función de relación binaria se permitiría centrarnos en contextos específicos.
- **Supervised Attention** Si se cuentan con datos alineados es posible entrenar la función de atención directamente.

Modelos nuevos

Pointer-Net

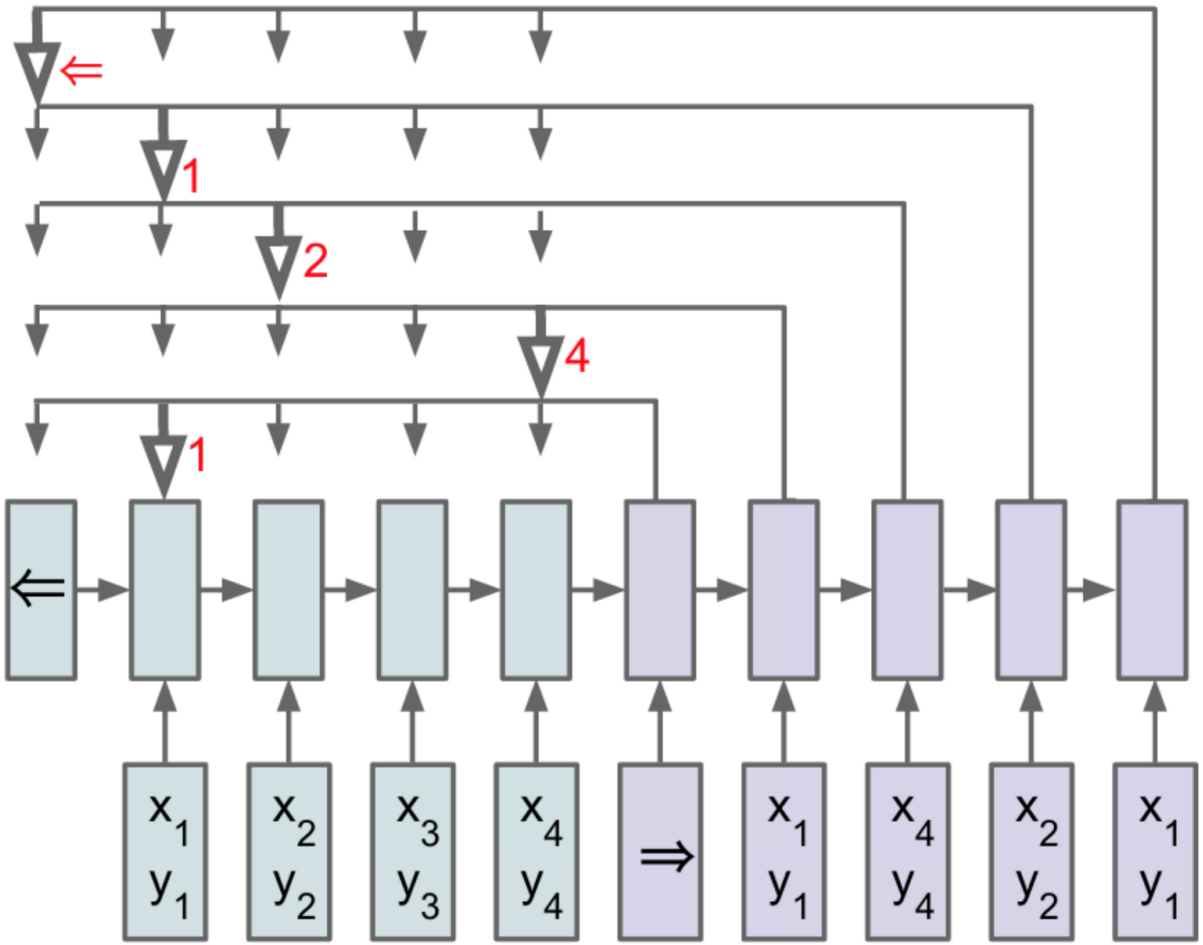

Figure: Fuente: Vinyals (2016)

Pointer Generator-Net

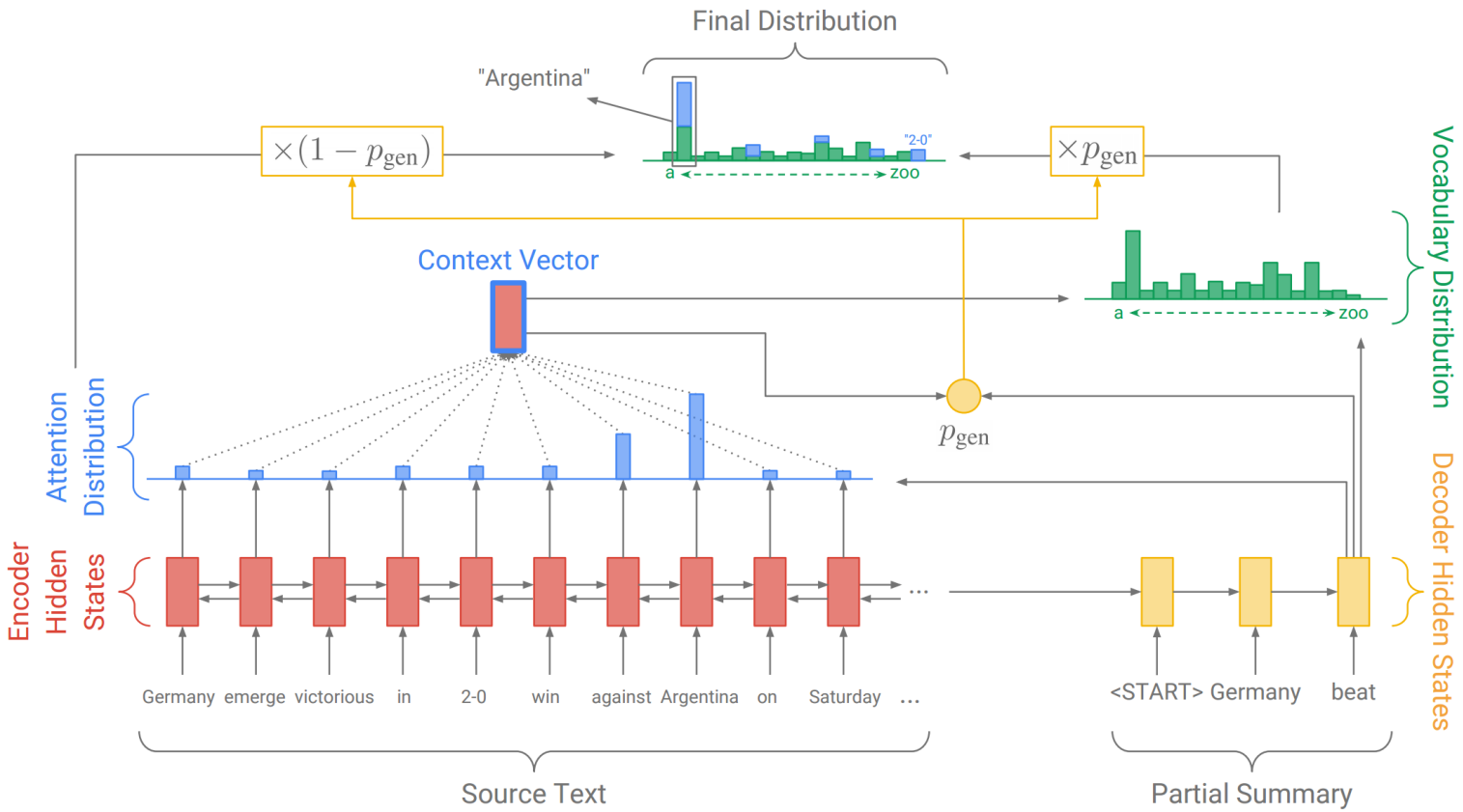

Figure: Fuente: See et al. (2016)

Transformers

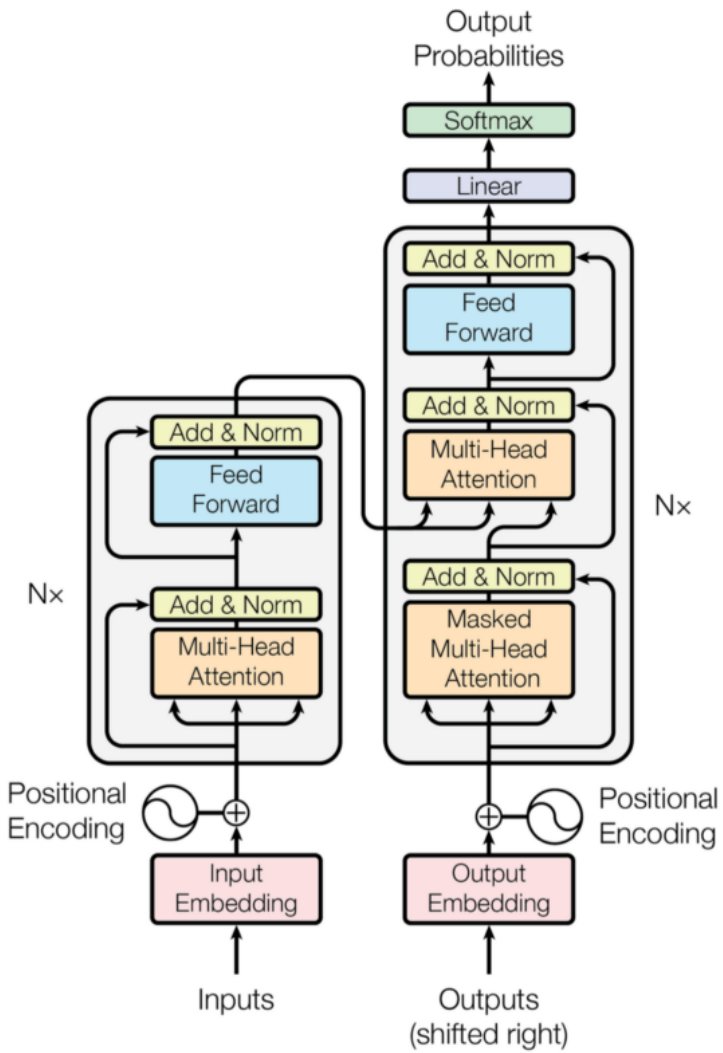

Figure 1: The Transformer - model architecture.