I SPDL

Deep Learning for Natural Language Processing

César Bragagnini

cesarbrma91@gmail.com
@MarchBragagnini

Universidad Católica San Pablo

ML/DL meetup AQP

Outline

1. Natural Language Processing
2. Word Embeddings
3. Neural networks
    a. RNN & Vanish/Exploding Gradient
    b. LSTM, GRU.
    c. Bidirectional-RNN
    d. Recursive Neural Networks
    e. Stack RNN
    f. Leakey units/residual connections
4. Neural Language Modeling
5. Sentiment Analysis
6. Name Entity Recognition(NER)
7. Opinion Mining
8. Attention/self Attention
9. Machine Translation
    a. Encoder-Decoder Sutskever
    b. GNMT
    c. Conv2Seq
    d. Transformer
10. Image Caption
11. Question Answering/Machine Comprehension / Chat bot
12. Visual Question Answering
13. Transfer Learning in NLP
14. References

1. Natural Language Processing

NLP es la intersección entre ciencia de la computación, inteligencia artificial y lingüística.

Objetivo

Lograr que la computadoras procesan o “entiendan” el lenguaje Natural para realizar tareas que son útiles, por ejemplo:

- Atender citas, realizar compras, atender clientes(bots)
- Responder preguntas (Question Answering)

El perfecto entendimiento del lenguaje, significa una IA completa.

[ ] Natural Language Processing with Deep Learning, Christopher Manning and Richard Socher, cs224n,2017.

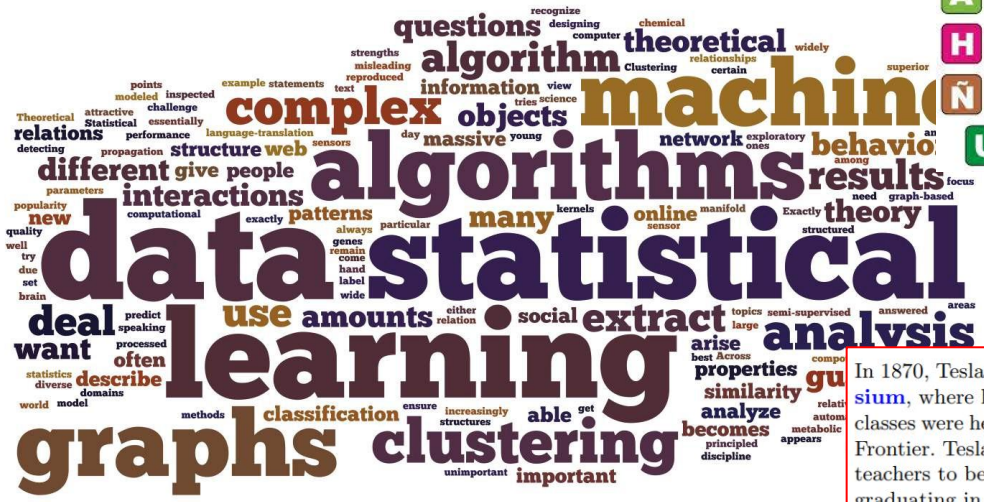
1. Natural Language Processing

En general

[ ] Natural Language Processing with Deep Learning, Christopher Manning and Richard Socher, cs224n,2017.

1. Natural Language Processing

Datos de *Natural Language Processing*



« » ‹ › ‘ ’ “ ” „ , ❝ ❞ £ ¥ € $ ¢
¬ ¶ @ § ® © ™ ° × π ± √ ‰
Ω ∞ ≈ ÷ ~ ≠ ¹ ² ³ ½ ¼ ¾ - –
— |/\ [ ] { } † ‡ … · • ● ⇧ ↩
¡ ¿ ? ⁂ ∴ ∵ ◊ ※ ← → ↑ ↓ ☜
☞ ☝ ☟ ✓ ★ ☆ ☼ ☂ ☺ ☹ ☃
✉ ✿ ✂ ✈ ✌ ✎ ♠ ♦ ♣ ♥ ♪ ♫ ♯

In 1870, Tesla moved to Karlovac, to attend school at the Higher Real Gymnasium, where he was profoundly influenced by a math teacher Martin Sekulic. The classes were held in German, as it was a school within the Austro-Hungarian Military Frontier. Tesla was able to perform integral calculus in his head, which prompted his teachers to believe that he was cheating. He finished a four-year term in three years, graduating in 1873.

[ ] Medium, Retention: machine learning at home

2. Word Embeddings

Representación de los datos - *One Hot Encoding*

$$w^{aardvark} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, w^{a} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, w^{at} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \cdots w^{zebra} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

Problema? No representa bien la relación entre 2 palabras

$$(w^{hotel})^T w^{motel} = (w^{hotel})^T w^{cat} = 0$$

[ ] Medium, Retention: machine learning at home

2. Word Embeddings

Embeddings - Base de todo de Natural Language Processing

Es una función que mapea un token, palabra o carácter de cierto lenguaje a un vector de una dimensión alta.

$$W : \text{words} \to \mathbb{R}^n$$

$$W(\text{“cat”}) = (0.2,\ \text{-}0.4,\ 0.7,\ \ldots)$$

$$W(\text{“mat”}) = (0.0,\ 0.6,\ \text{-}0.1,\ \ldots)$$

[ ] Word Embeddings, Christopher Olah.

2. Word Embeddings

Base de todo de Natural Language Processing

[ ] Natural Language Processing, Yves Peirsman.

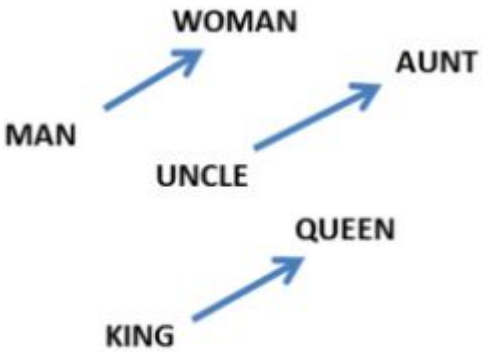
2. Word Embeddings

Base de todo de Natural Language Processing

Podemos realizar operaciones aritméticas como:

$$W(\text{“woman”}) - W(\text{“man”}) \simeq W(\text{“aunt”}) - W(\text{“uncle”})$$

$$W(\text{“woman”}) - W(\text{“man”}) \simeq W(\text{“queen”}) - W(\text{“king”})$$



From Mikolov *et al.* (2013a)

[ ] Word Embeddings, Tomas Mikolov.

2. Word Embeddings

Base de todo de Natural Language Processing

¿Qué define el significado de una palabra ?

El significado de una palabra está determinado por su contexto

1. El **perro** está ladrando y moviendo la colita
2. El **can** está ladrando y moviendo la colita.
3. El **boby** está ladrando y moviendo la colita.
4. El **Chiquito** está ladrando y moviendo la colita.

¿Qué representa la palabra boby y Chiquito ?

[ ] Word Embeddings, Tomas Mikolov.

2. Word Embeddings

Base de todo de Natural Language Processing

El significado de una palabra está determinado por su contexto

1. El **perro** está ladrando y moviendo la colita
2. El **can** está ladrando y moviendo la colita.
3. El **boby** está ladrando y moviendo la colita.
4. El **Chiquito** está ladrando y moviendo la colita.

Si un modelador de lenguaje, genera la siguiente frase:

1. El **boby** ha maullado muy fuerte, y ahora está hablando con Lari.

[ ] Word Embeddings, Tomas Mikolov.

2. Word Embeddings

Base de todo de Natural Language Processing

El significado de una palabra está determinado por su contexto

1. El **perro** está ladrando y moviendo la colita
2. El **can** está ladrando y moviendo la colita.
3. El **boby** está ladrando y moviendo la colita.
4. El **Chiquito** está ladrando y moviendo la colita.

Si un modelador de lenguaje, genera la siguiente frase:

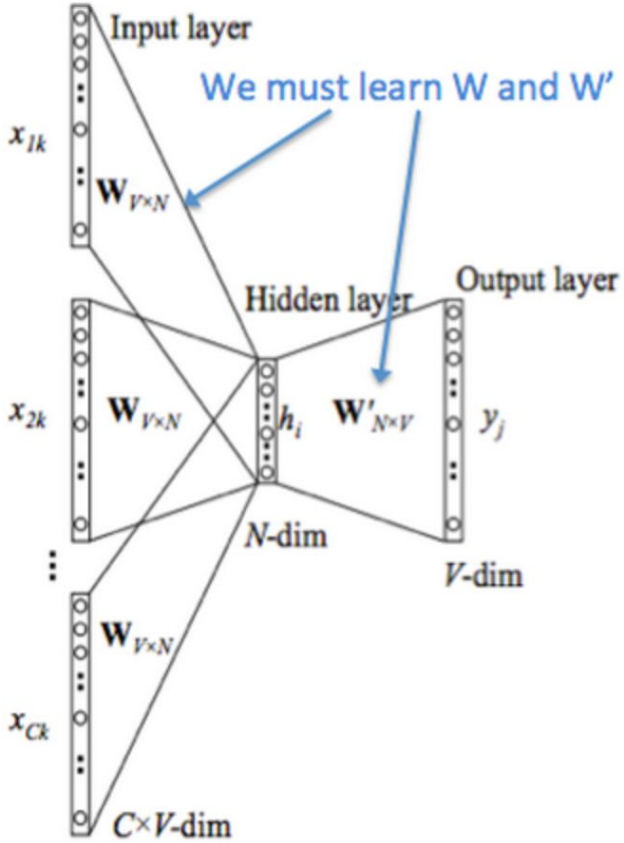
1. El **boby** ha maullado muy fuerte, y ahora está hablando con Lari.

Tal vez, sea un mal generador de lenguaje

[ ] Word Embeddings, Tomas Mikolov.

2. Word Embeddings

Word2Vec - CBOW Model



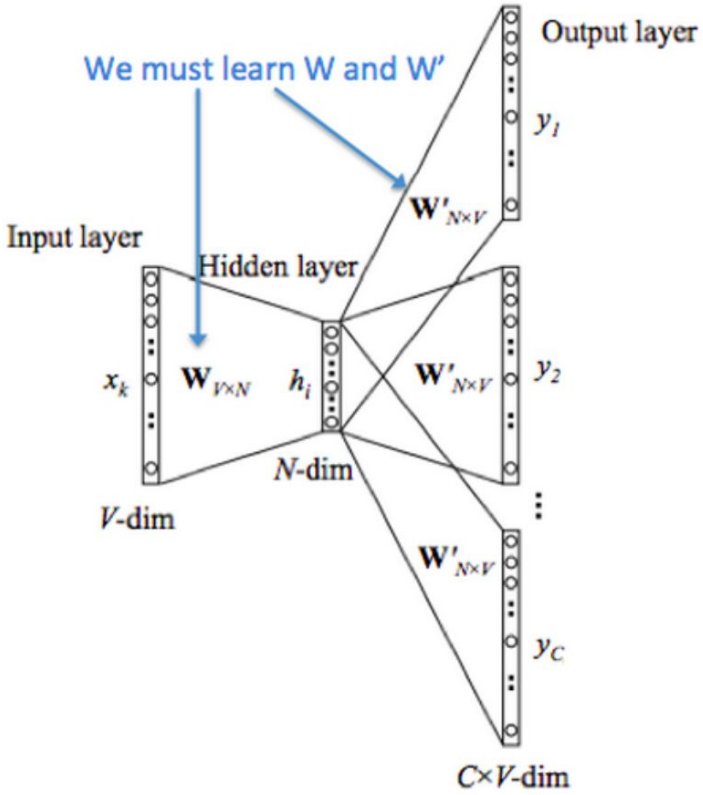
Notation for CBOW Model:

- w_i: Word i from vocabulary V
- $\mathcal{V} \in \mathbb{R}^{n\times|V|}$: Input word matrix
- v_i: i-th column of $\mathcal{V}$, the input vector representation of word w_i
- $\mathcal{U} \in \mathbb{R}^{n\times|V|}$: Output word matrix
- u_i: i-th row of $\mathcal{U}$, the output vector representation of word w_i

[ ] Natural Language Processing with Deep Learning, Manning, 2016.

2. Word Embeddings

Word2Vec - Skip Gram Model



Notation for Skip-Gram Model:

- w_i: Word i from vocabulary V
- $\mathcal{V} \in \mathbb{R}^{n\times|V|}$: Input word matrix
- v_i: i-th column of $\mathcal{V}$, the input vector representation of word w_i
- $\mathcal{U} \in \mathbb{R}^{n\times|V|}$: Output word matrix
- u_i: i-th row of $\mathcal{U}$, the output vector representation of word w_i

[ ] Natural Language Processing with Deep Learning, Manning, 2016.

2. Word Embeddings

Problema con los word embeddings

1. La visita al zoológico no se cobra, y se puede observar la cobra recién adquirida
2. Visitamos las plantaciones de coca, mientras tomábamos una coca cola
3. Poniendo tu cara bonita, difícilmente te salga cara la entrada

[ ] Natural Language Processing with Deep Learning, Manning, 2016.

3. Neural Networks

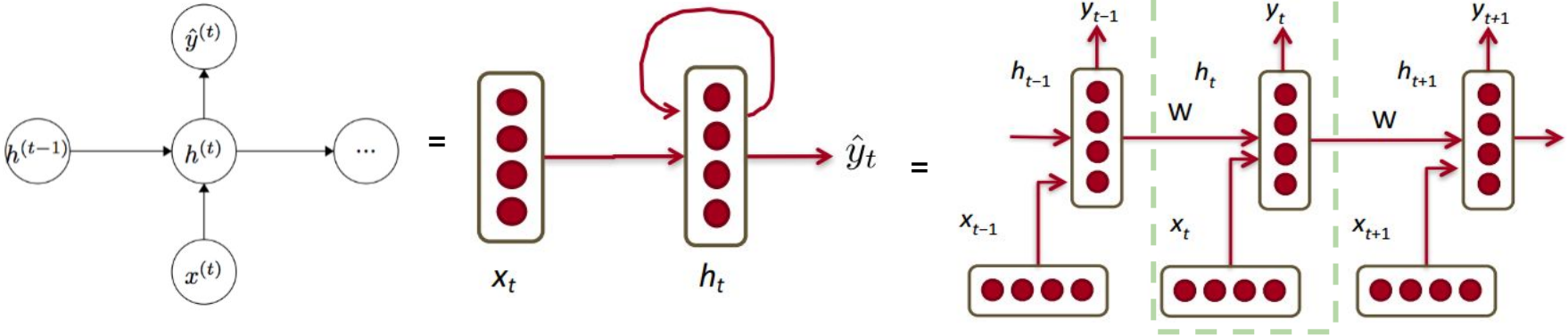
3.1 Introducción

el adobo delicioso parece me .

1. adobo delicioso me el .
2. el adobo me parece delicioso .
3. . adobo me el delicioso

3. Neural Networks

3.1 Recurrent Neural Networks & Vanish/Exploding Gradient



Given a list of words: $x_1, \ldots, x_{t-1}, x_t, x_{t+1}, \ldots, x_T$

At a single time step:

$$h_t = \sigma\left(W^{(hh)} h_{t-1} + W^{(hx)} x_{[t]}\right)$$

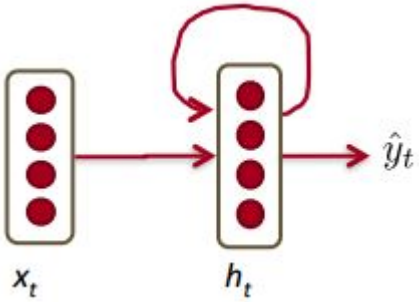
$$\hat{y}_t = \text{softmax}\left(W^{(S)} h_t\right)$$

[ ] RNN, cs224d, Richard Socher, 2016.

3. Neural Networks

3.1 Recurrent Neural Networks & Vanish/Exploding Gradient

$$h_t = W f(h_{t-1}) + W^{(hx)} x_{[t]}$$
$$\hat{y}_t = W^{(S)} f(h_t)$$



$$\frac{\partial E}{\partial W} = \sum_{t=1}^{T} \frac{\partial E_t}{\partial W}$$

$$\frac{\partial E_t}{\partial W} = \sum_{k=1}^{t} \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W}$$

$$\frac{\partial E_t}{\partial W} = \sum_{k=1}^{t} \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \boxed{\frac{\partial h_t}{\partial h_k}} \frac{\partial h_k}{\partial W}$$

$$\frac{\partial h_t}{\partial h_k} = \boxed{\prod_{j=k+1}^{t} \frac{\partial h_j}{\partial h_{j-1}}}$$

[ ] CS224N/Ling284, Lecture 8, Stanford University, Manning , 2016.

3. Neural Networks

3.1 Recurrent Neural Networks & Vanish/Exploding Gradient

Each partial is a Jacobian

$$\frac{d\mathbf{f}}{d\mathbf{x}} = \left[\frac{\partial \mathbf{f}}{\partial x_1} \quad \cdots \quad \frac{\partial \mathbf{f}}{\partial x_n}\right]$$

$$= \begin{bmatrix} \partial x_1 & & \partial x_n \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Analyzing the norms of the Jacobians, yields:

$$\left\| \frac{\partial h_j}{\partial h_{j-1}} \right\| \leq \|W^T\| \|\text{diag}[f'(h_{j-1})]\| \leq \beta_W \beta_h$$

Where we defined β's as upper bounds of the norms

The gradient is a product of Jacobian matrices, each associated with a step in the forward computation.

$$\left\| \frac{\partial h_t}{\partial h_k} \right\| = \left\| \prod_{j=k+1}^{t} \frac{\partial h_j}{\partial h_{j-1}} \right\| \leq (\beta_W \beta_h)^{t-k}$$

This can become very small or very large quickly

[ ] CS224N/Ling284, Lecture 8, Stanford University, Manning , 2016.
[ ] Learning Long-Term Dependencies with Gradient Descent is Difficult, Bengio, 1994.

3. Neural Networks

3.1 Recurrent Neural Networks & Vanish/Exploding Gradient

The basic problem is that gradients propagated over many stages tend to either vanish (most of the time) or explode (rarely, but with much damage to the optimization)

$$\boldsymbol{h}^{(t)} = \boldsymbol{W}^{\top} \boldsymbol{h}^{(t-1)}$$

$$\boldsymbol{h}^{(t)} = \left(\boldsymbol{W}^{t}\right)^{\top} \boldsymbol{h}^{(0)}, \quad \boldsymbol{W} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{\top}$$

$$\boldsymbol{h}^{(t)} = \boldsymbol{Q}^{\top}\boldsymbol{\Lambda}^{t}\boldsymbol{Q}\boldsymbol{h}^{(0)}$$

The eigenvalues are raised to the power of ***t***, causing eigenvalues with magnitude less than one to decay to zero and eigenvalues with magnitude greater than one to explode.

Any component of ***h(0)*** that is not aligned with the largest eigenvector will eventually be discarded

Problem: Long Term Dependencies, information whose outputs depend on very early entries can not be remembered.

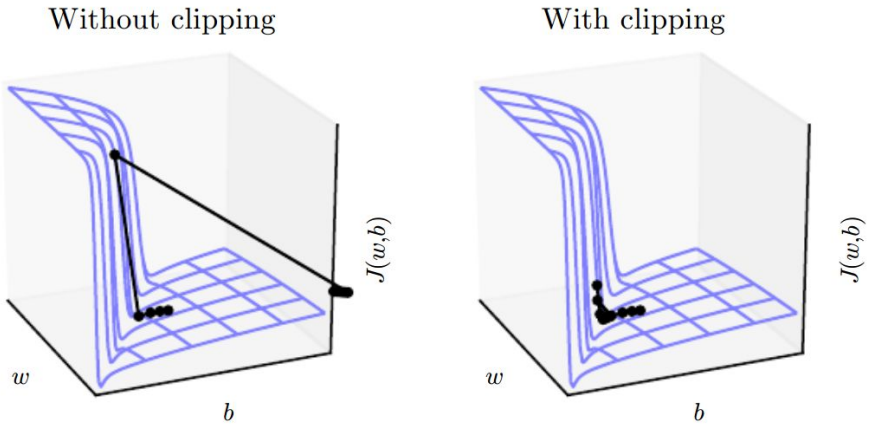
Input: Teo es la mascota de Lari, es un perrito pequeño y bien cariñoso, siempre sale a pasear todos los domingos, le gusta ladrar a la gente y a las perritas. A su mascota no le gustan las chalinas

Can RNN predict the missing word? : La mascota de Lari se llama ____

[ ] Deep Learning, Ian Goodfellow, Yoshua Bengio, Aaron Courville, 2016.

3. Neural Networks

3.1 Recurrent Neural Networks & Vanish/Exploding Gradient



Example of the effect of gradient clipping, in a recurrent network with two parameters w and b. For solving exploding gradient.

* **(Left)**Gradient descent without gradient clipping overshoots the bottom of this small ravine, then receives a very large gradient from the cliff face. The large gradient catastrophically propels the parameters outside the axes of the plot.

* **(Right)**Gradient descent with gradient clipping has a more moderate reaction to the cliff.

[ ] Deep Learning, Ian Goodfellow, Yoshua Bengio, Aaron Courville, 2016.

3. Neural Networks

3.1 Recurrent Neural Networks & Vanish/Exploding Gradient

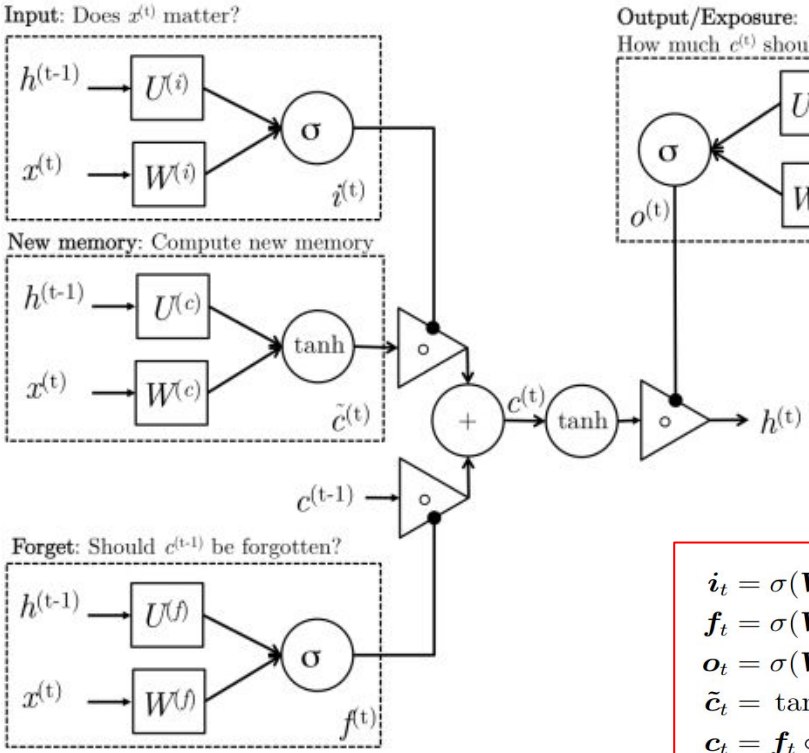
Algorithm 1 Pseudo-code for norm clipping the gradients whenever they explode

$\hat{\mathbf{g}} \leftarrow \frac{\partial \mathcal{E}}{\partial \theta}$
if $\|\hat{\mathbf{g}}\| \geq threshold$ **then**
  $\hat{\mathbf{g}} \leftarrow \frac{threshold}{\|\hat{\mathbf{g}}\|}\hat{\mathbf{g}}$
end if

[ ] Deep Learning, Ian Goodfellow, Yoshua Bengio, Aaron Courville, 2016.
[ ] Advanced in Optimizing Recurrent Networks, Ian Goodfellow, Yoshua Bengio, Aaron Courville, 2016.

3. Neural Networks

3.2 Long Short Term Memory(LSTM) & ...



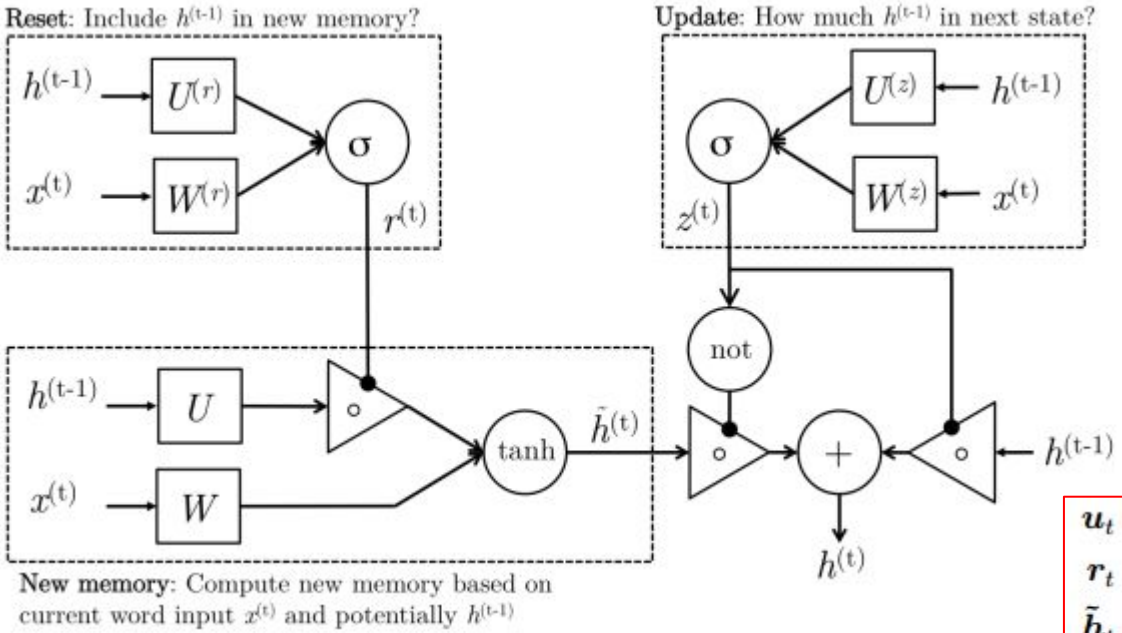
LSMT architecture.

$$
\begin{aligned}
\boldsymbol{i}_t &= \sigma(\boldsymbol{W}_i[\boldsymbol{x}_t; \boldsymbol{h}_{t-1}] + \boldsymbol{b}_i) && \text{Input Gate} \\
\boldsymbol{f}_t &= \sigma(\boldsymbol{W}_f[\boldsymbol{x}_t; \boldsymbol{h}_{t-1}] + \boldsymbol{b}_f) && \text{Forget Gate} \\
\boldsymbol{o}_t &= \sigma(\boldsymbol{W}_o[\boldsymbol{x}_t; \boldsymbol{h}_{t-1}] + \boldsymbol{b}_o) && \text{Output Gate} \\
\tilde{\boldsymbol{c}}_t &= \tanh(\boldsymbol{W}_{\hat{h}}[\boldsymbol{x}_t; \boldsymbol{h}_{t-1}] + \boldsymbol{b}_{\hat{h}}) && \text{New Cell Memory} \\
\boldsymbol{c}_t &= \boldsymbol{f}_t \circ \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \circ \tilde{\boldsymbol{c}}_t && \text{Final Memory} \\
\boldsymbol{h}_t &= \boldsymbol{o}_t \circ \tanh(\boldsymbol{c}_{t-1}) && \text{Hidden State}
\end{aligned}
$$

[ ] Lecture note 04 in Deep Learning for Natural Language Processing, Mohammadi, 2015.
[ ] Recurrent Neural Network Regularization, Wojciech Zarembam, Ilya Sutskever and Oriol Vinyals, 2014.

3. Neural Networks

3.2 Gated Recurrent Unit(GRU)



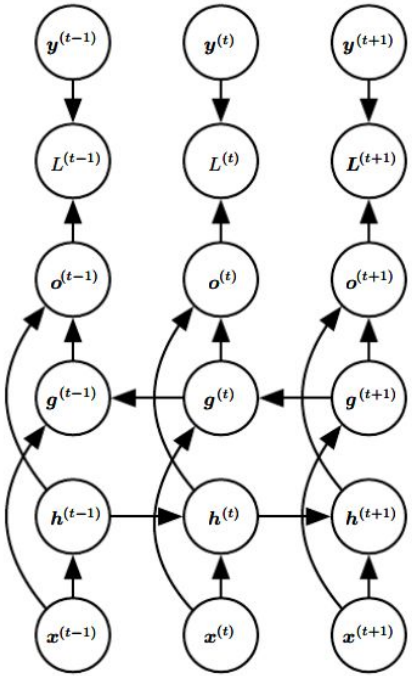
GRU architecture.

$$
\begin{aligned}
\boldsymbol{u}_t &= \sigma(\boldsymbol{W}_u[\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_u) && \text{Update Gate} \\
\boldsymbol{r}_t &= \sigma(\boldsymbol{W}_r[\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_r) && \text{Restart Gate} \\
\tilde{\boldsymbol{h}}_t &= \tanh(\boldsymbol{W}_h[\boldsymbol{r}_t \circ \boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_h) && \text{New Memory} \\
\boldsymbol{h}_t &= \boldsymbol{u}_t \circ \tilde{\boldsymbol{h}}_t + (\vec{1} - \boldsymbol{u}_t) \circ \boldsymbol{h}_{t-1} && \text{Hidden State}
\end{aligned}
$$

[ ] Learning phrase representations using RNN encoder-decoder for statistical machine translation, Cho, 2014.
[ ] Lecture note 04 in Deep Learning for Natural Language Processing, Mohammadi, 2015.

3. Neural Networks

3.4 Bidirectional - RNN



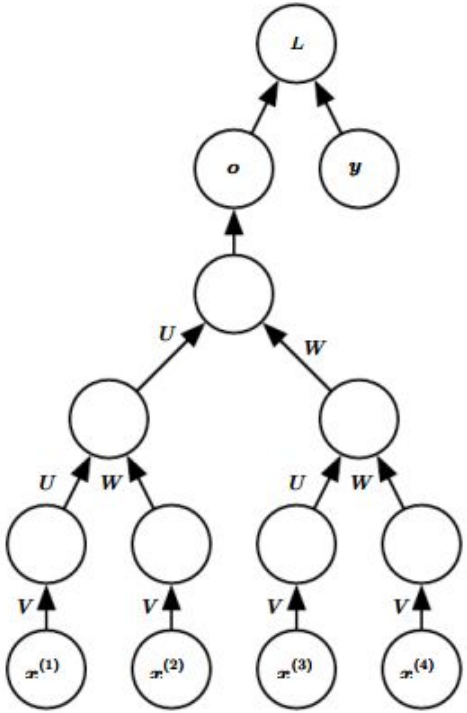
$$
\begin{aligned}
\boldsymbol{h}^{(t-1)} &= f(\boldsymbol{W}_h[\boldsymbol{h}^{(t-2)}, \boldsymbol{x}^{(t-1)}] + \boldsymbol{b}_h) \\
\boldsymbol{h}^{(t)} &= f(\boldsymbol{W}_h[\boldsymbol{h}^{(t-1)}, \boldsymbol{x}^{(t)}] + \boldsymbol{b}_h) \\
\boldsymbol{g}^{(t+1)} &= f(\boldsymbol{W}_g[\boldsymbol{g}^{(t+2)}, \boldsymbol{x}^{(t+1)}] + \boldsymbol{b}_g) \\
\boldsymbol{g}^{(t)} &= f(\boldsymbol{W}_g[\boldsymbol{g}^{(t+1)}, \boldsymbol{x}^{(t)}] + \boldsymbol{b}_g)
\end{aligned}
$$

$$
\boldsymbol{o}^{(t)} = \hat{\boldsymbol{y}}_{(t)} = \mathrm{Softmax}(\boldsymbol{W}_o[\boldsymbol{h}^{(t)}, \boldsymbol{g}^{(t)}] + \boldsymbol{b}_o)
$$

[ ] Deep Learning, Ian Goodfellow, Yoshua Bengio, Aaron Courville, 2016.

3. Neural Networks

3.5 Recursive Neural Networks

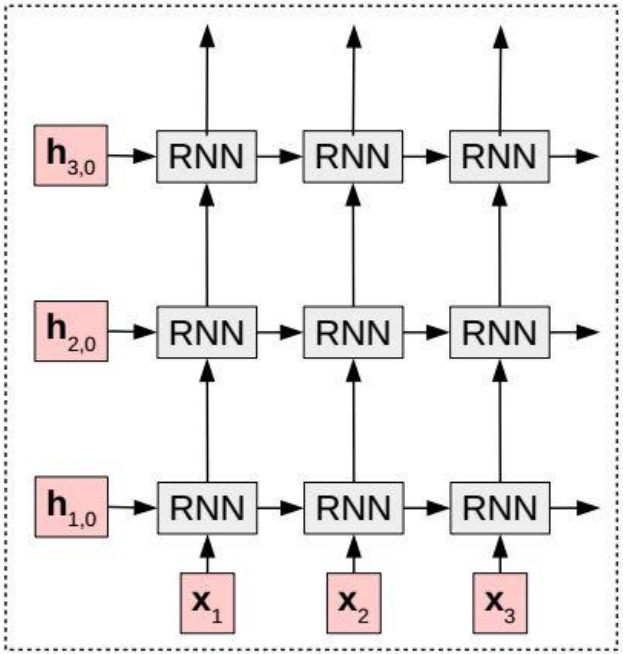


A recursive network has a computational graph that generalizes that of the recurrent network from a chain to a tree. A variable-size sequence x(1), x (2), . . . , x(t) can be mapped to a fixed-size representation (the output o), with a fixed set of parameters (the weight matrices U, V , W).

[ ] Deep Learning, Ian Goodfellow, Yoshua Bengio, Aaron Courville, 2016.

3. Neural Networks

3.6 Stack RNN



$$\boldsymbol{h}_{3,t} = \mathrm{RNN}_3(\boldsymbol{h}_{2,t}, \boldsymbol{h}_{3,t-1}, \boldsymbol{b}_3)$$

$$\boldsymbol{h}_{2,t} = \mathrm{RNN}_2(\boldsymbol{h}_{1,t}, \boldsymbol{h}_{2,t-1}, \boldsymbol{b}_2)$$

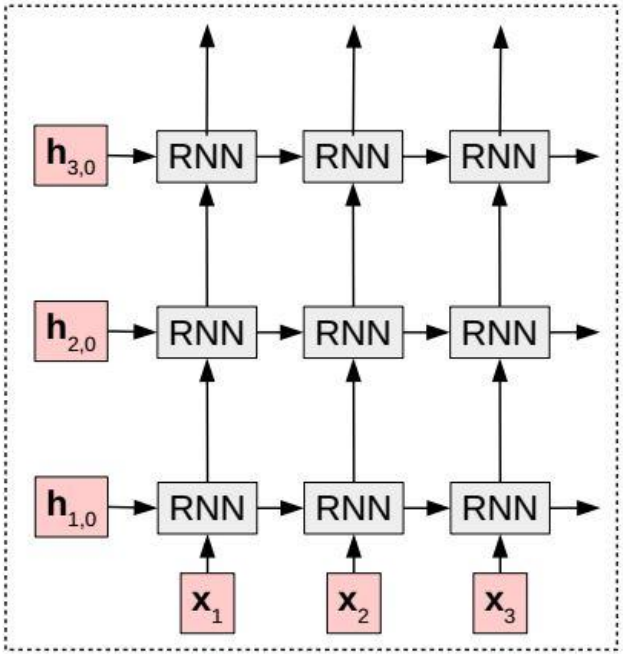
$$\boldsymbol{h}_{1,t} = \mathrm{RNN}_1(\boldsymbol{x}_t, \boldsymbol{h}_{1,t-1}, \boldsymbol{b}_1)$$

[ ] Neural Machine Translation and Sequence-to-sequence Models: A Tutorial, Graham Neubig, 2017.

3. Neural Networks

3.6 Stack RNN

Trouble?



$$\boldsymbol{h}_{3,t} = \mathrm{RNN}_3(\boldsymbol{h}_{2,t}, \boldsymbol{h}_{3,t-1}, \boldsymbol{b}_3)$$

$$\boldsymbol{h}_{2,t} = \mathrm{RNN}_2(\boldsymbol{h}_{1,t}, \boldsymbol{h}_{2,t-1}, \boldsymbol{b}_2)$$

$$\boldsymbol{h}_{1,t} = \mathrm{RNN}_1(\boldsymbol{x}_t, \boldsymbol{h}_{1,t-1}, \boldsymbol{b}_1)$$

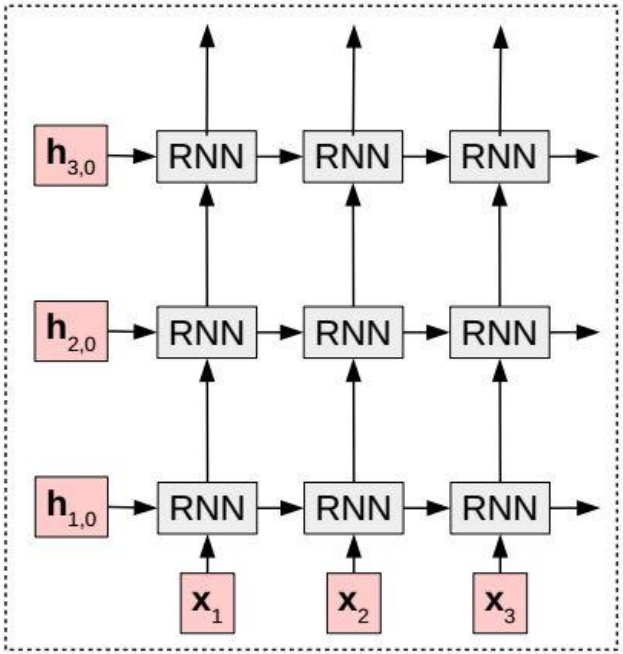
[ ] Neural Machine Translation and Sequence-to-sequence Models: A Tutorial, Graham Neubig, 2017.

3. Neural Networks

3.6 Stack RNN

Trouble?

Maybe appear gradient vanish problem in vertical



$$\boldsymbol{h}_{3,t} = \mathrm{RNN}_3(\boldsymbol{h}_{2,t}, \boldsymbol{h}_{3,t-1}, \boldsymbol{b}_3)$$

$$\boldsymbol{h}_{2,t} = \mathrm{RNN}_2(\boldsymbol{h}_{1,t}, \boldsymbol{h}_{2,t-1}, \boldsymbol{b}_2)$$

$$\boldsymbol{h}_{1,t} = \mathrm{RNN}_1(\boldsymbol{x}_t, \boldsymbol{h}_{1,t-1}, \boldsymbol{b}_1)$$

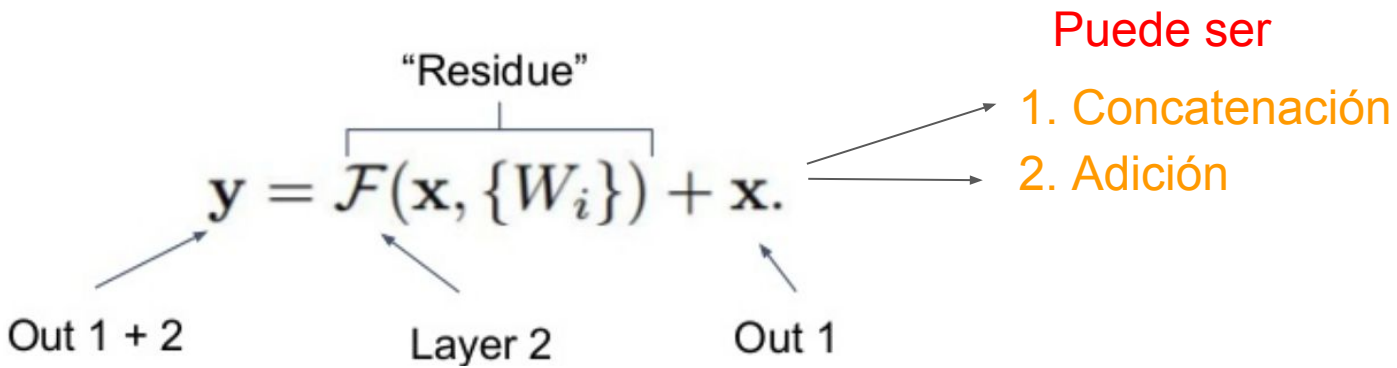
[ ] Neural Machine Translation and Sequence-to-sequence Models: A Tutorial, Graham Neubig, 2017.

3. Neural Networks

3.6 Residual Connections/Skip Connections & ...

Gradients may vanish or explode exponentially with respect to the number of time steps.

To introduce of recurrent connections with a time-delay of ***d*** to mitigate this problem. Gradients now diminish exponentially as a function of ***T/d*** rather than ***T***.



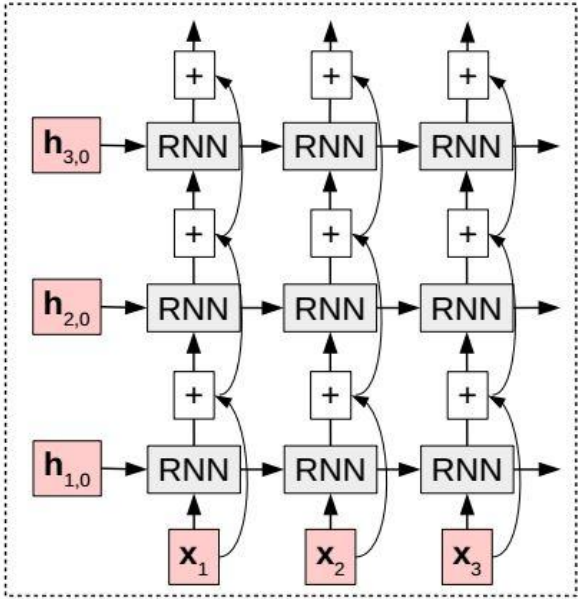
[ ] Neural Machine Translation and Sequence-to-sequence Models: A Tutorial, Graham Neubig, 2017.

3. Neural Networks

3.6 Residual Connections/Skip Connections & ...

Gradients may vanish or explode exponentially with respect to the number of time steps.

To introduce of recurrent connections with a time-delay of ***d*** to mitigate this problem. Gradients now diminish exponentially as a function of ***T/d*** rather than ***T***.



$$\boldsymbol{h}_{3,t} = \mathrm{RNN}(\boldsymbol{h}_{2,t}, \boldsymbol{h}_{3,t-1}, \boldsymbol{b}_3) + \boldsymbol{h}_{2,t}$$

$$\boldsymbol{h}_{2,t} = \mathrm{RNN}(\boldsymbol{h}_{1,t}, \boldsymbol{h}_{2,t-1}, \boldsymbol{b}_2) + \boldsymbol{h}_{1,t}$$

$$\boldsymbol{h}_{1,t} = \mathrm{RNN}(\boldsymbol{x}_t, \boldsymbol{h}_{1,t-1}, \boldsymbol{b}_1) + \boldsymbol{x}_t$$

[ ] Neural Machine Translation and Sequence-to-sequence Models: A Tutorial, Graham Neubig, 2017.

3. Neural Networks

3.6 Leaky Units

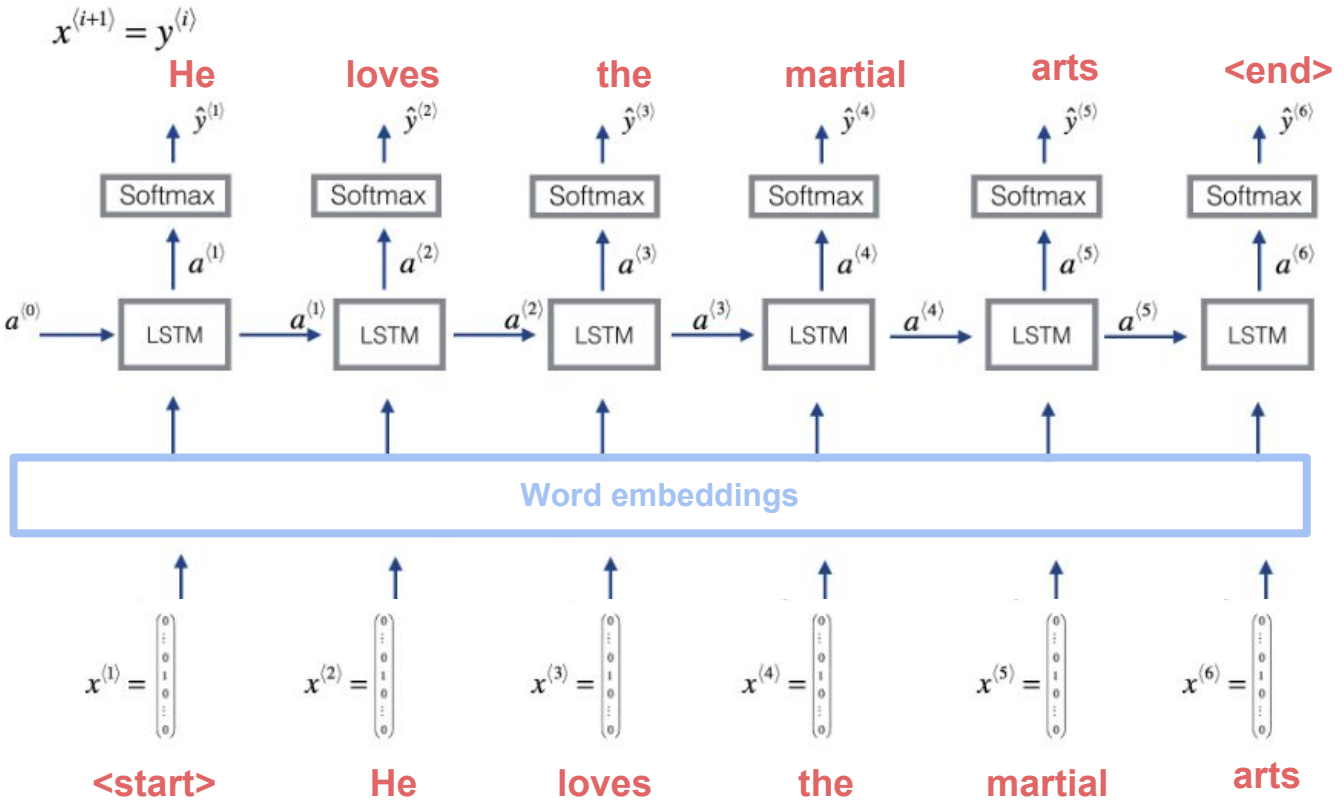
$$\boldsymbol{y} = \alpha * \mathcal{F}(\boldsymbol{x}, \{\boldsymbol{W}_i\}) + (1 - \alpha) * \boldsymbol{x}$$

$$0 \leq \alpha \leq 1$$

[ ] Deep Learning, Ian Goodfellow, Yoshua Bengio, Aaron Courville, 2016.

4. Neural Language Modeling

Language Modeling Train

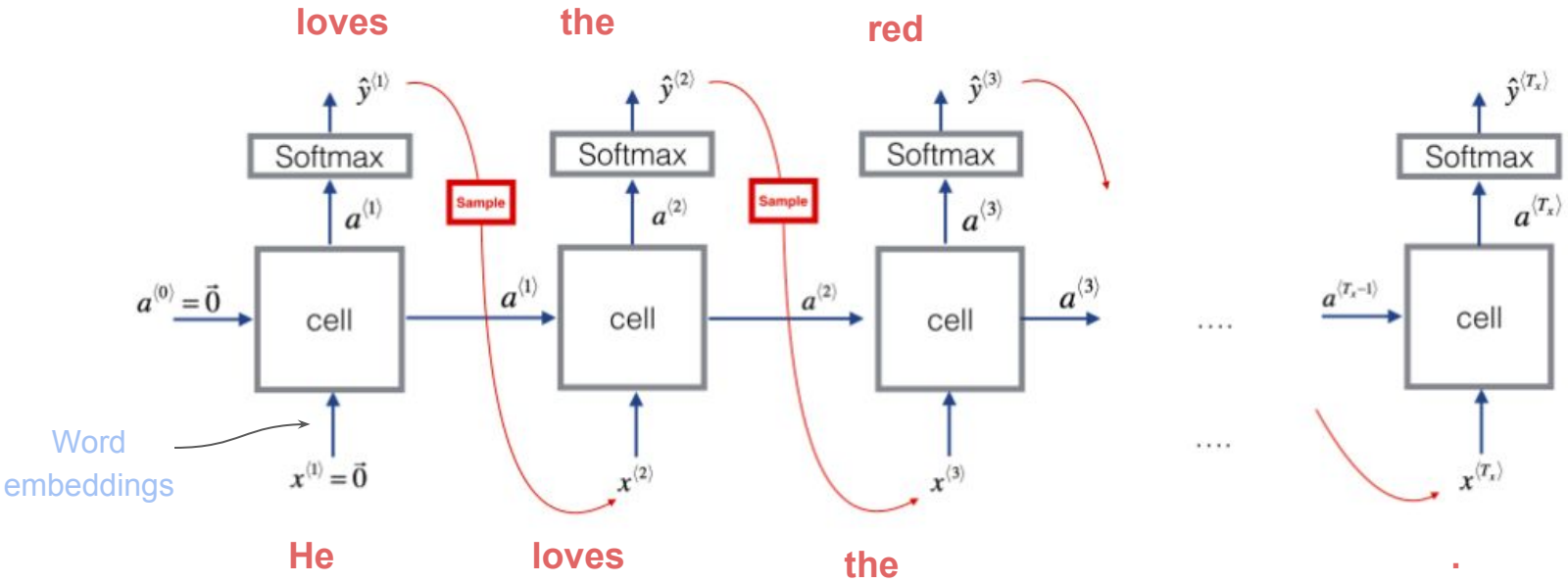


[ ] Deep Learning Specialization, Sequence Models, Andrew Ng, 2016.

4. Neural Language Modeling

Application

Language Modeling Testing

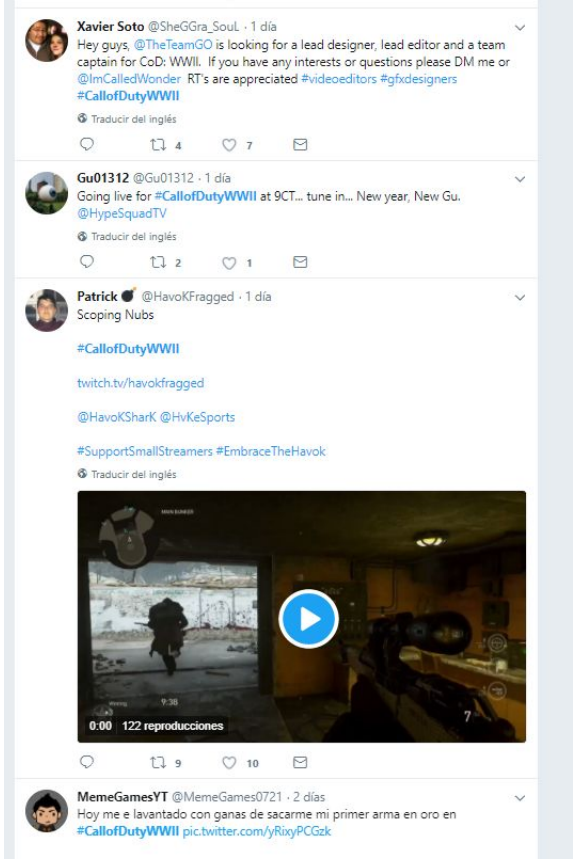


[ ] Deep Learning Specialization, Sequence Models, Andrew Ng, 2016.

5. Sentiment Analysis

Representa un sentimiento positivo, negativo o neutro ??

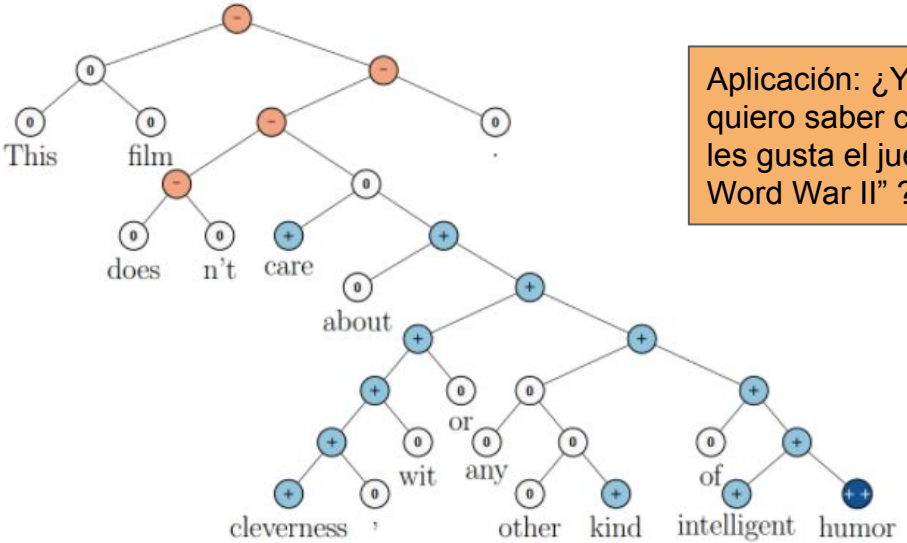
Aplicación: ¿Yo como empresa quiero saber cuantas personas les gusta el juego “Call of Dutty Word War II” ?



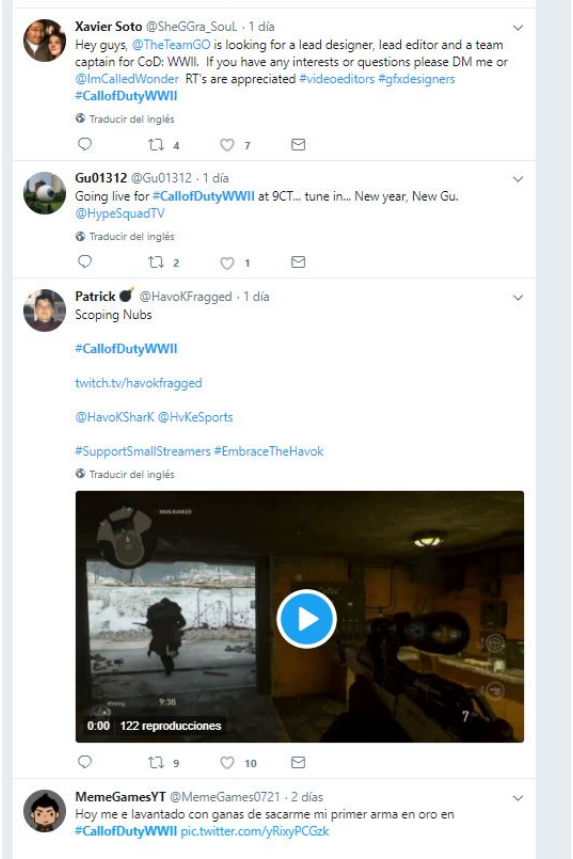
[ ] Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, Socher, 2013.

5. Sentiment Analysis

Representa un sentimiento positivo, negativo o neutro ??



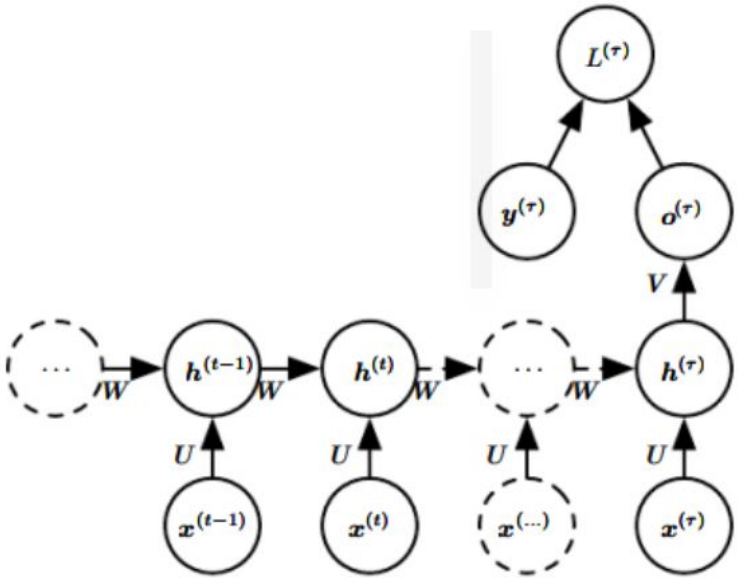
Aplicación: ¿Yo como empresa quiero saber cuantas personas les gusta el juego “Call of Dutty Word War II” ?



[ ] Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, Socher, 2013.

5. Sentiment Analysis

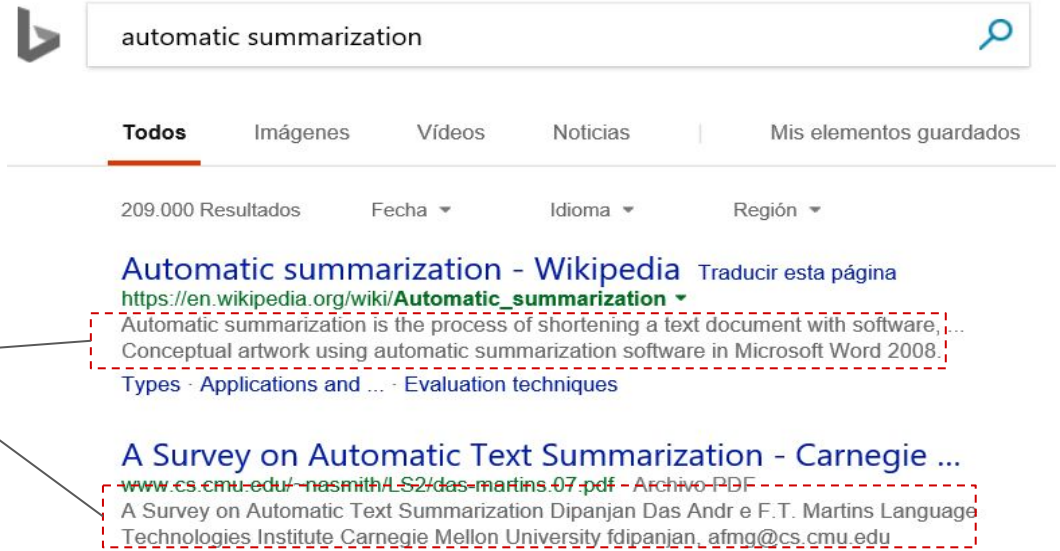
RNN for many-to-one



(a) RNN model to mapping an input sequence $(\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(\tau)})$ to a fixed vector $\boldsymbol{o}^{\tau}$.

[ ] Deep Learning, Ian Goodfellow, Yoshua Bengio and Aaron Courville, 2016.

6. Text Summarization



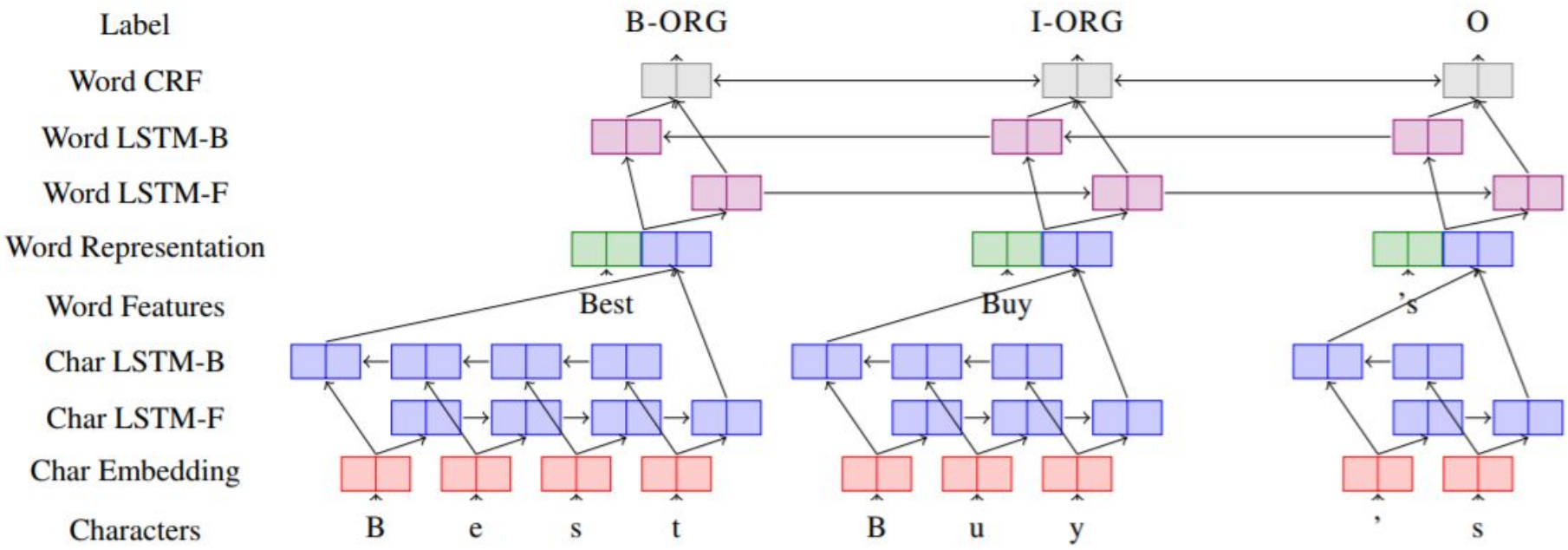
Pueden ser resumenes

Tipo Extractivo: Si todo el resumen está conformado por palabras presente en el texto.

Tipo abstracto: Si el resumen no necesariamente usa palabras dentro del texto. Extrapolación.

http://www.bing.com

6. Name Entity Recognition(NER)

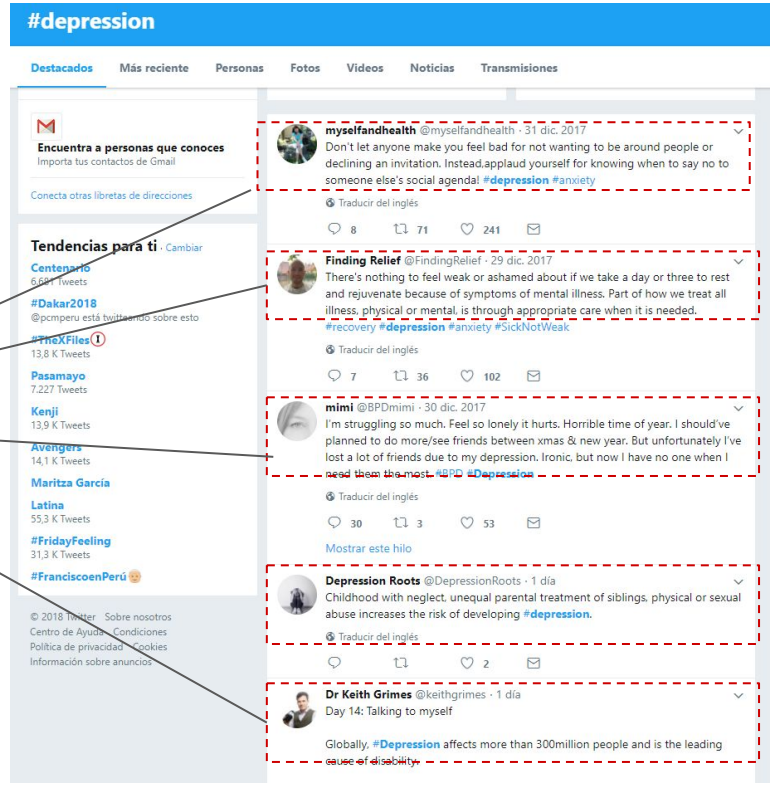


Word+Character level NN architecture for NER.

[ ] Deep Learning, Ian Goodfellow, Yoshua Bengio and Aaron Courville, 2016.

7. Opinion Mining

¿ Realmente tiene depresion ?



http://www.twitter.com

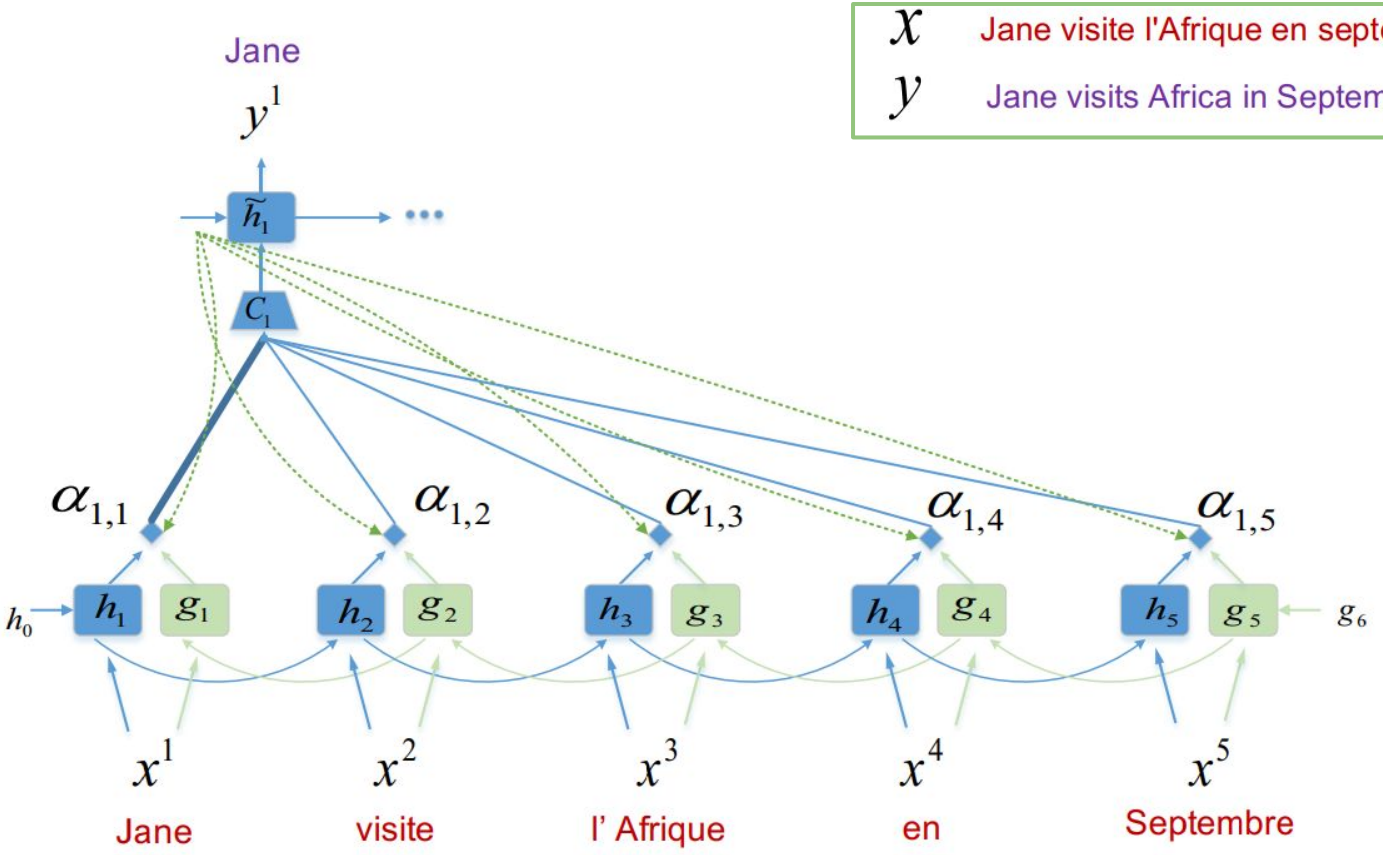
8. Attention / Self Attention

$$y = f(x)$$

x Jane visite l'Afrique en septembre

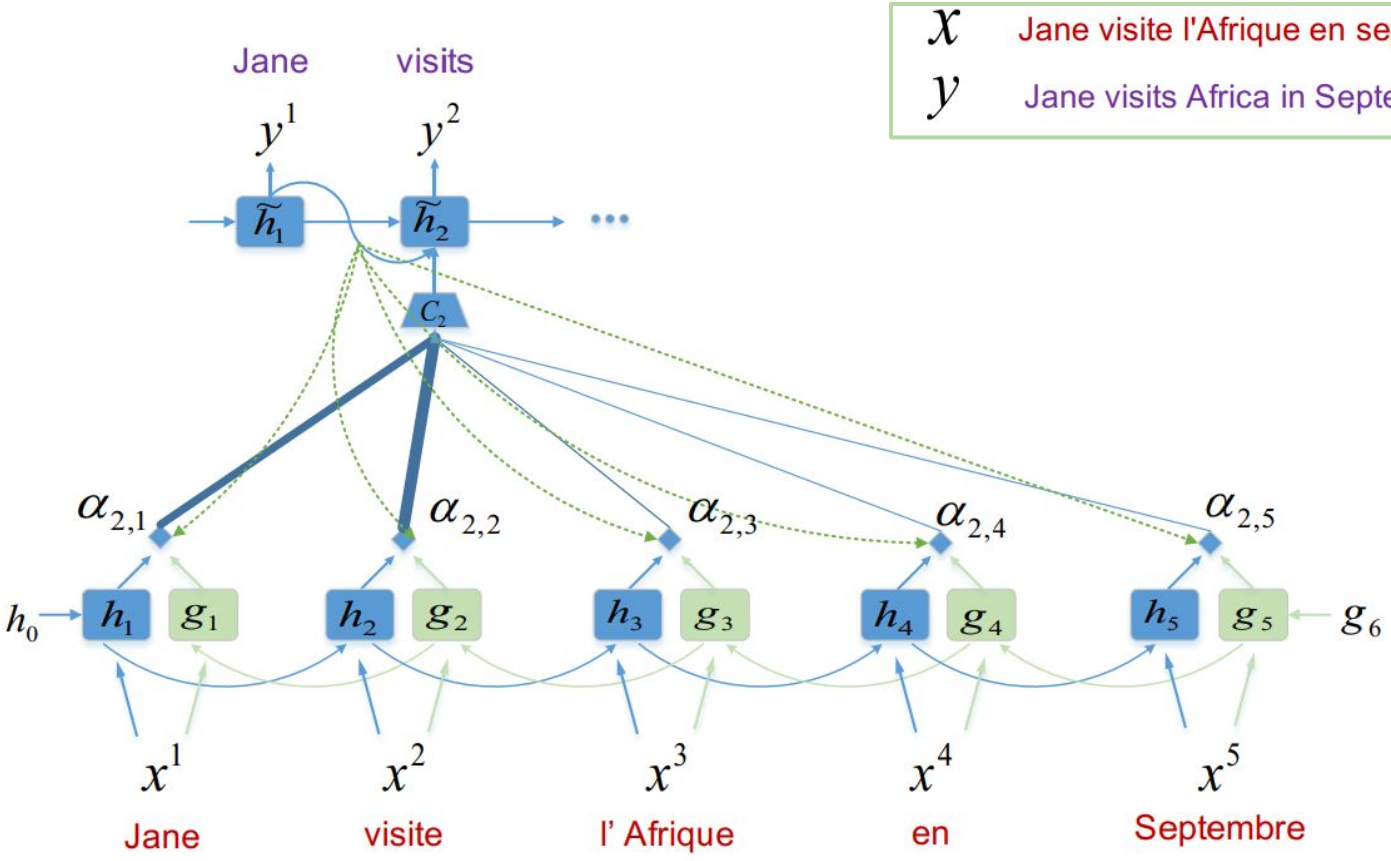
y Jane visits Africa in September

8. Attention / Self Attention



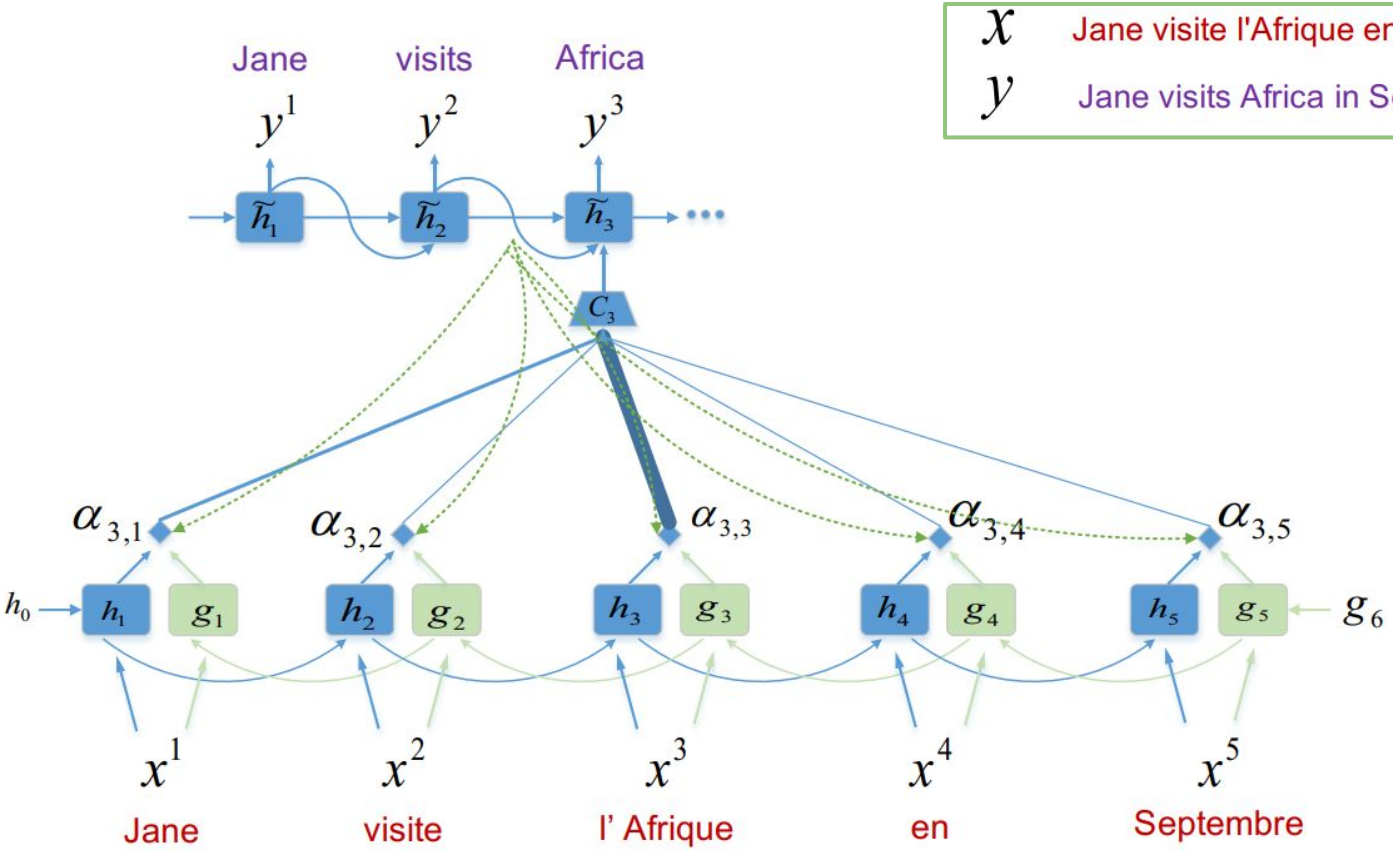
Atención en *Machine Translation* en el paso t=1

8. Attention / Self Attention



Atención en *Machine Translation* en el paso t=1

8. Attention / Self Attention



Atención en *Machine Translation* en el paso t=2

8. Attention / Self Attention

Formulación matemática

Score value

$$\boldsymbol{h}_1^c = [\boldsymbol{h}_1, \boldsymbol{g}_1]$$

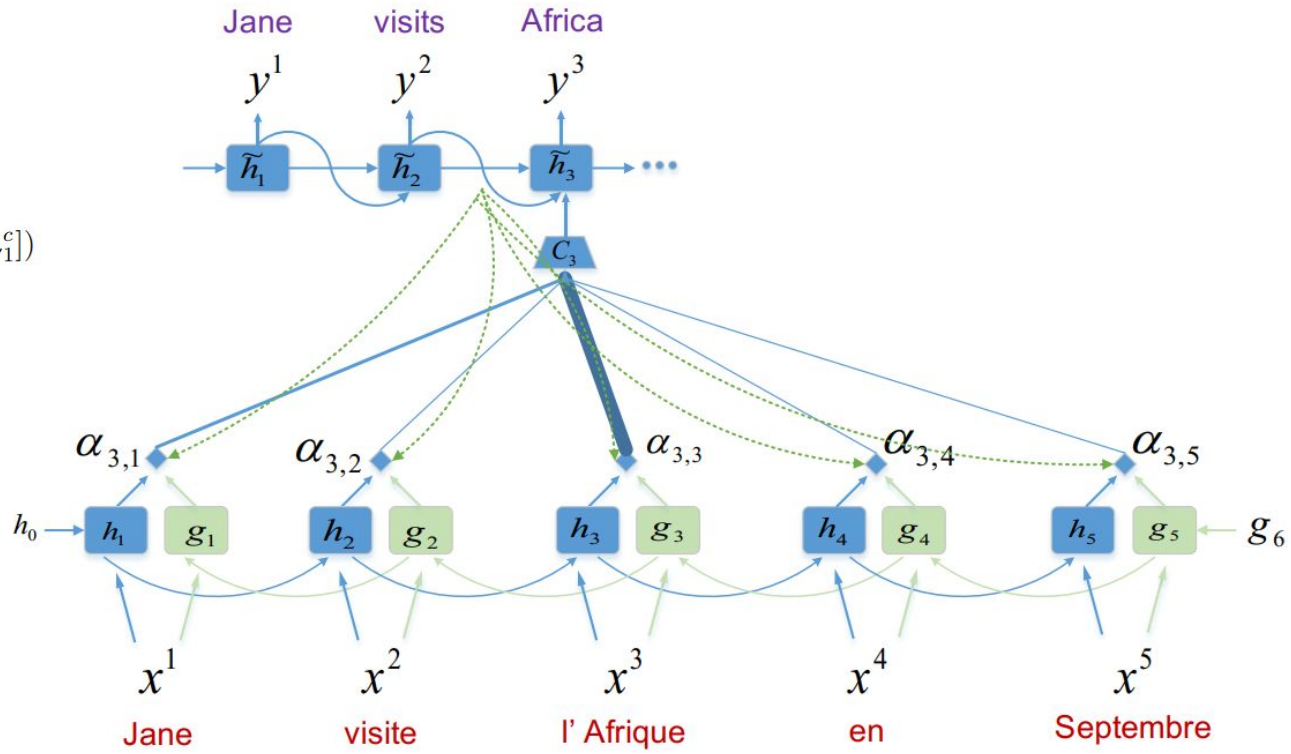
$$\boldsymbol{e}_{3,1} = v_a^{\mathrm{T}} \tanh(\boldsymbol{W}_a[\tilde{\boldsymbol{h}}_2, \boldsymbol{h}_1^c])$$

$$\boldsymbol{\alpha}_{3,1} = \frac{\exp(\boldsymbol{e}_{3,1})}{\sum_{k=1}^{5} \exp(\boldsymbol{e}_{3,k})}$$

Score attention

Recordando

$$\sum_{k=1}^{T_x} \boldsymbol{\alpha}_{i,k} = 1$$



Atención en *Machine Translation* en el paso t=3

[ ] Neural Machine Translation by Jointly Learning to Align and Translate, Bahdanau, 2014

8. Attention / Self Attention

Formulación matemática

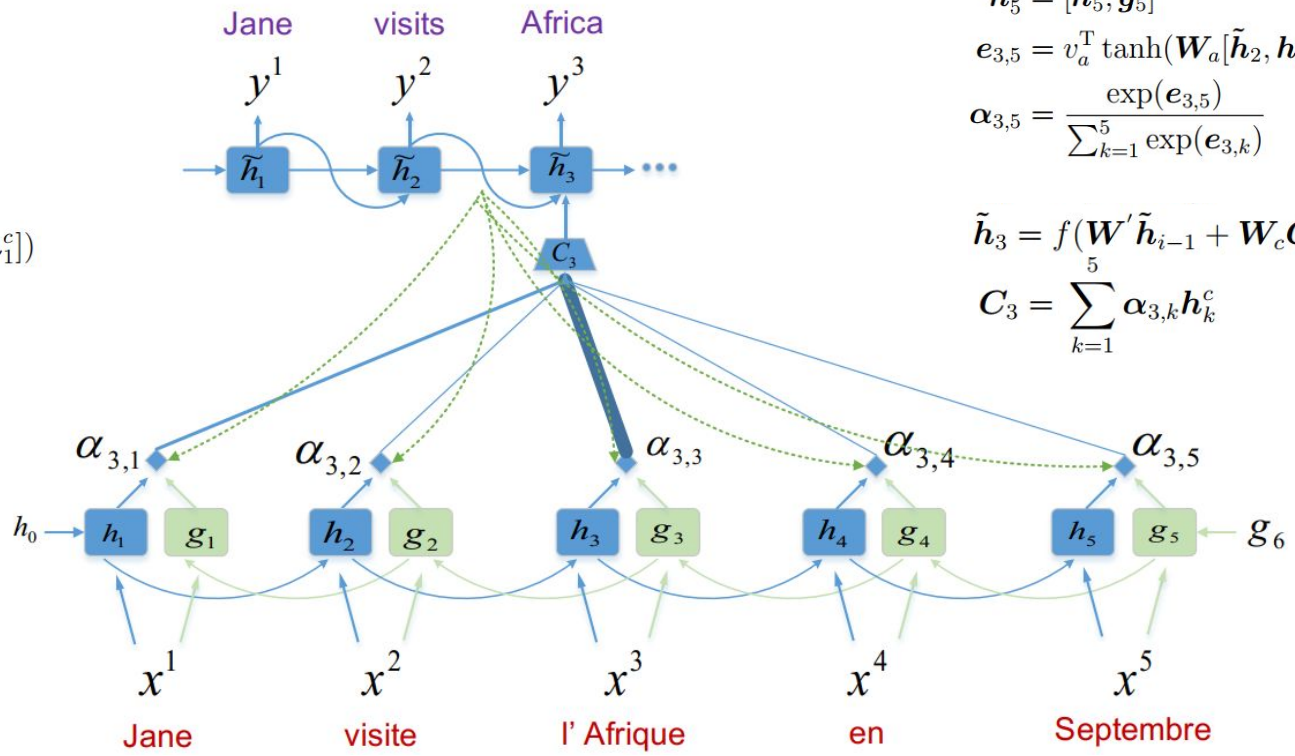
Score value

$$h_1^c = [h_1, g_1]$$
$$e_{3,1} = v_a^{\mathrm{T}} \tanh(W_a[\tilde{h}_2, h_1^c])$$
$$\alpha_{3,1} = \frac{\exp(e_{3,1})}{\sum_{k=1}^{5} \exp(e_{3,k})}$$

Score attention

$$h_5^c = [h_5, g_5]$$
$$e_{3,5} = v_a^{\mathrm{T}} \tanh(W_a[\tilde{h}_2, h_5^c])$$
$$\alpha_{3,5} = \frac{\exp(e_{3,5})}{\sum_{k=1}^{5} \exp(e_{3,k})}$$

$$\tilde{h}_3 = f(W' \tilde{h}_{i-1} + W_c C_3 + b_{eh})$$
$$C_3 = \sum_{k=1}^{5} \alpha_{3,k} h_k^c$$



Atención en *Machine Translation* en el paso t=3

[ ] Neural Machine Translation by Jointly Learning to Align and Translate, Bahdanau, 2014

8. Attention / Self Attention

Formulación matemática

En general para el paso t = 3

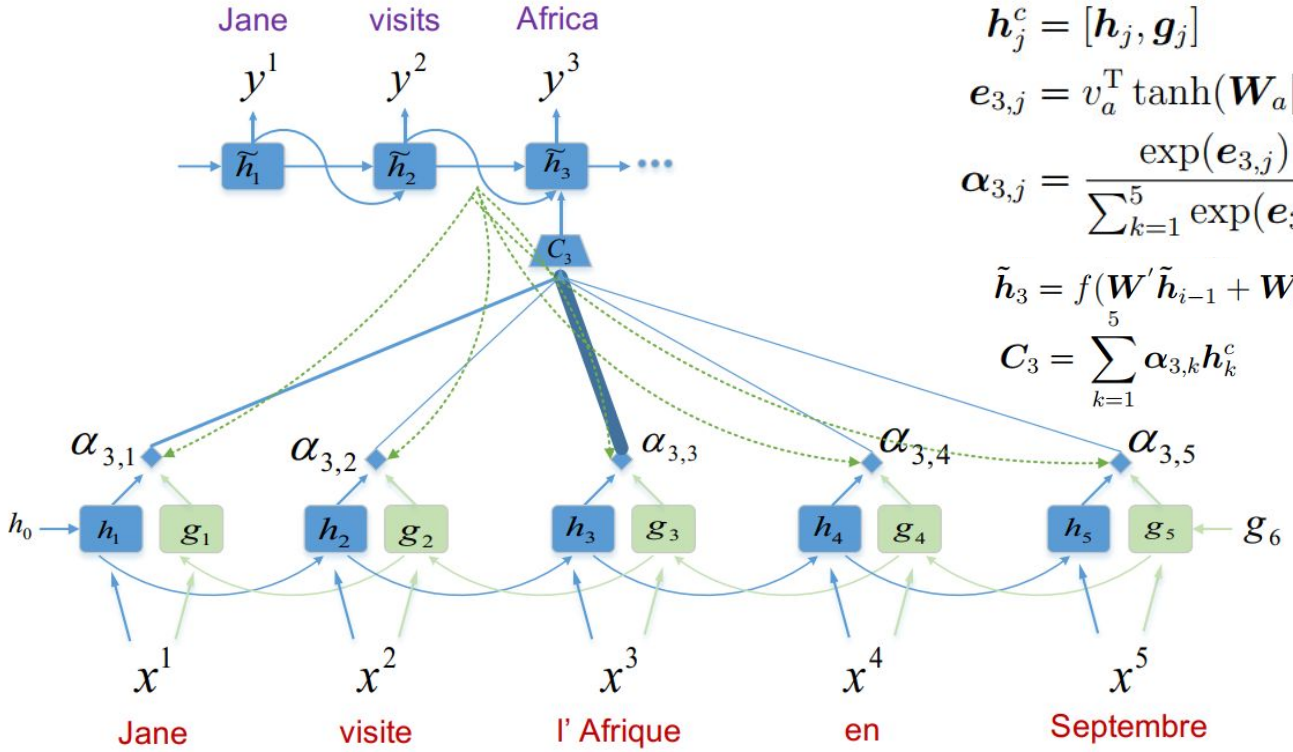
$$\boldsymbol{h}_j^c = [\boldsymbol{h}_j, \boldsymbol{g}_j]$$

$$\boldsymbol{e}_{3,j} = v_a^{\mathrm{T}} \tanh(\boldsymbol{W}_a[\tilde{\boldsymbol{h}}_2, \boldsymbol{h}_j^c])$$

$$\boldsymbol{\alpha}_{3,j} = \frac{\exp(\boldsymbol{e}_{3,j})}{\sum_{k=1}^{5} \exp(\boldsymbol{e}_{3,k})}$$

$$\tilde{\boldsymbol{h}}_3 = f(\boldsymbol{W}' \tilde{\boldsymbol{h}}_{i-1} + \boldsymbol{W}_c \boldsymbol{C}_3 + \boldsymbol{b}_{eh})$$

$$\boldsymbol{C}_3 = \sum_{k=1}^{5} \boldsymbol{\alpha}_{3,k} \boldsymbol{h}_k^c$$

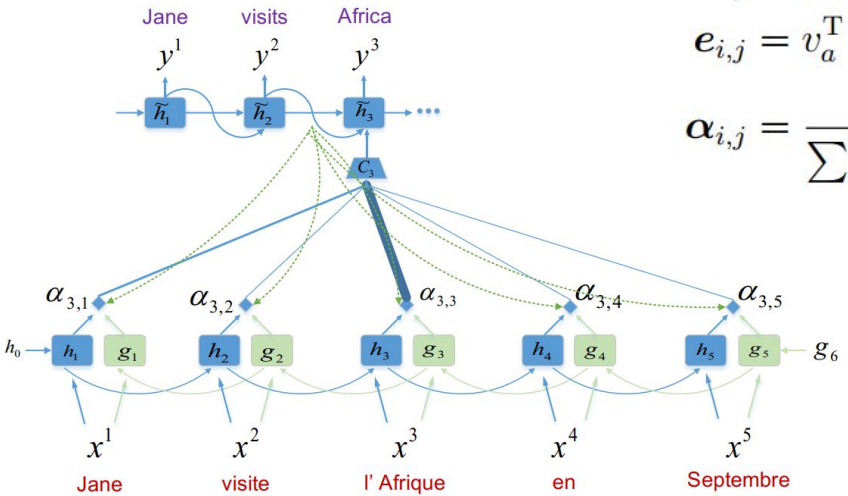


Atención en *Machine Translation* en el paso t=2

[ ] Neural Machine Translation by Jointly Learning to Align and Translate, Bahdanau, 2014

8. Attention / Self Attention

Formulación matemática



Atención en *Machine Translation* en el paso t=3

Para una sentencia x de tamaño Tx

$$\boldsymbol{h}_j^c = [\boldsymbol{h}_j, \boldsymbol{g}_j]$$

$$\boldsymbol{e}_{i,j} = v_a^{\mathrm{T}} \tanh(\boldsymbol{W}_a[\tilde{\boldsymbol{h}}_{i-1}, \boldsymbol{h}_j^c])$$ → **Score value**

$$\boldsymbol{\alpha}_{i,j} = \frac{\exp(\boldsymbol{e}_{i,j})}{\sum_{k=1}^{T_x} \exp(\boldsymbol{e}_{i,k})}$$ → **Score attention**

$$\tilde{\boldsymbol{h}}_3 = f(\boldsymbol{W}' \tilde{\boldsymbol{h}}_{i-1} + \boldsymbol{W}_c \boldsymbol{C}_3 + \boldsymbol{b}_{eh})$$

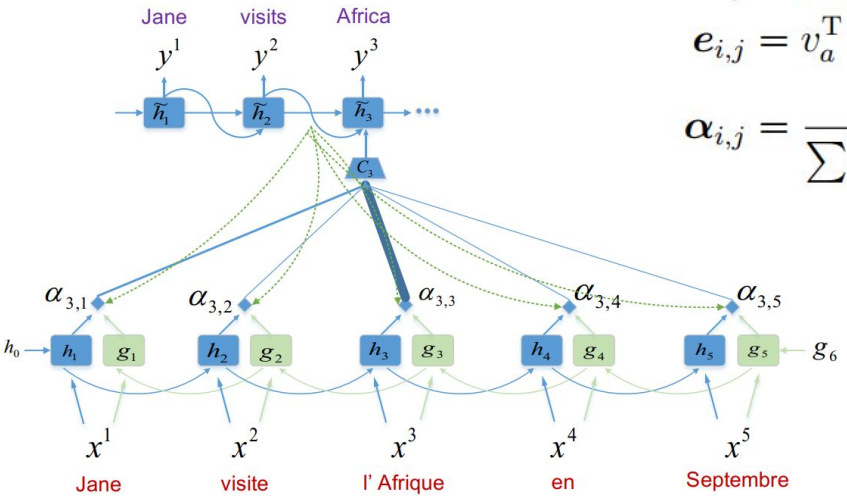
$$\boldsymbol{C}_i = \sum_{k=1}^{T_x} \boldsymbol{\alpha}_{i,k} \boldsymbol{h}_k^c$$

Cual *score* aprende a quién dar más atención ??

[ ] Neural Machine Translation by Jointly Learning to Align and Translate, Bahdanau, 2014

8. Attention / Self Attention

Formulación matemática



Atención en *Machine Translation* en el paso t=3

Para una sentencia x de tamaño Tx

$$\boldsymbol{h}_j^c = [\boldsymbol{h}_j, \boldsymbol{g}_j]$$

$$\boldsymbol{e}_{i,j} = v_a^{\mathrm{T}} \tanh(\boldsymbol{W}_a[\tilde{\boldsymbol{h}}_{i-1}, \boldsymbol{h}_j^c])$$ → **Score value**

$$\boldsymbol{\alpha}_{i,j} = \frac{\exp(\boldsymbol{e}_{i,j})}{\sum_{k=1}^{T_x} \exp(\boldsymbol{e}_{i,k})}$$ → **Score attention**

$$\tilde{\boldsymbol{h}}_3 = f(\boldsymbol{W}'\tilde{\boldsymbol{h}}_{i-1} + \boldsymbol{W}_c\boldsymbol{C}_3 + \boldsymbol{b}_{eh})$$

$$\boldsymbol{C}_i = \sum_{k=1}^{T_x} \boldsymbol{\alpha}_{i,k}\boldsymbol{h}_k^c$$

Cual *score* aprende a quién dar más atención ??

El *score value*

[ ] Neural Machine Translation by Jointly Learning to Align and Translate, Bahdanau, 2014

8. Attention / Self Attention

Distintos *attention* ~ distintas formas de calcular *score value*

En resumen un score value es calculado a partir de 2 valores, en nuestro ejemplo $\tilde{\boldsymbol{h}}_{i-1}$ y $\boldsymbol{h}_j^c$

$$\boldsymbol{e}_{i,j} = \mathcal{F}(\tilde{\boldsymbol{h}}_{i-1}, \boldsymbol{h}_j^c)$$

Mecanismo	Score value
Bahdanau	$\boldsymbol{e}_{i,j} = v_a^{\mathrm{T}} \tanh(\boldsymbol{W}_a[\tilde{\boldsymbol{h}}_{i-1}, \boldsymbol{h}_j^c])$
Luong	$\boldsymbol{e}_{i,j} = \tilde{\boldsymbol{h}}_{i-1}^{\prime\mathrm{T}} \boldsymbol{h}_j^c$
Bahdanau Normado	$\boldsymbol{e}_{i,j} = g \frac{v_a}{\|v_a\|} \tanh(\boldsymbol{W}_a[\tilde{\boldsymbol{h}}_{i-1}, \boldsymbol{h}_j^c])$
Luong Escalado*	$\boldsymbol{e}_{i,j} = g \tilde{\boldsymbol{h}}_{i-1}^{\prime\mathrm{T}} \boldsymbol{h}_j^c$

Considerar, por fines didácticos usamos $i-1$

[ ] Neural Machine Translation by Jointly Learning to Align and Translate, Bahdanau, 2014.
[ ] Online and linear-time attention by enforcing monotonic alignments, Raffel, 2017.
[ ] Effective Approaches to Attention-based Neural Machine Translation, Luong, 2015
[ ] Attention is all you need, Vaswani, 2017.
[ ] Tensorflow Attention https://www.tensorflow.org/api_docs/python/tf/contrib/seq2seq/AttentionWrapper

8. Attention / Self Attention

Distintos *attention* ~ distintas formas de calcular *score value*

En resumen un score value es calculado a partir de 2 valores, en nuestro ejemplo $\tilde{\boldsymbol{h}}_{i-1}$ y $\boldsymbol{h}_j^c$

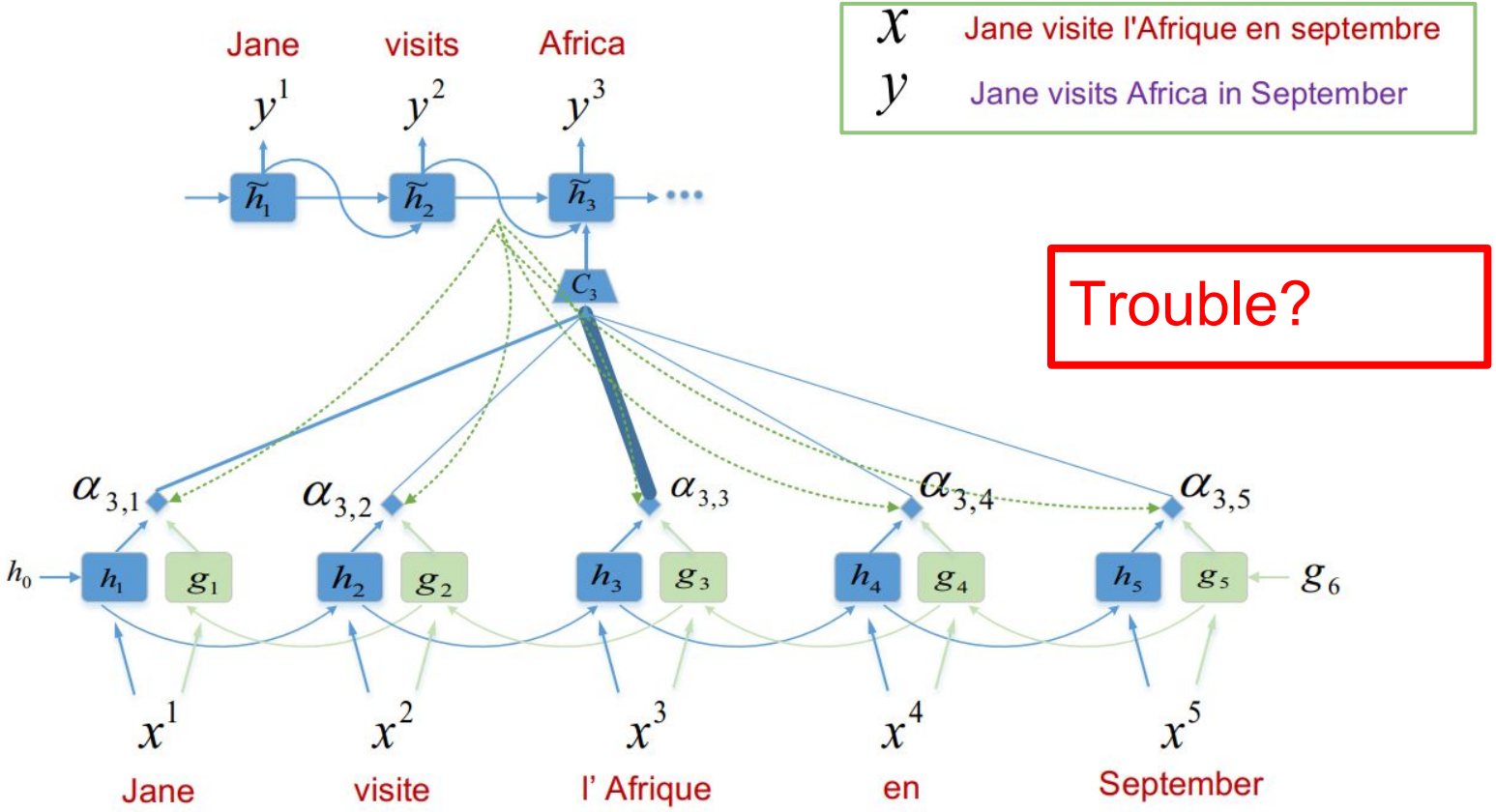
$$\boldsymbol{e}_{i,j} = \mathcal{F}(\tilde{\boldsymbol{h}}_{i-1}, \boldsymbol{h}_j^c)$$

Mecanismo	Score value
Bahdanau	$\boldsymbol{e}_{i,j} = v_a^{\mathrm{T}} \tanh(\boldsymbol{W}_a[\tilde{\boldsymbol{h}}_{i-1}, \boldsymbol{h}_j^c])$
Luong	$\boldsymbol{e}_{i,j} = \tilde{\boldsymbol{h}}_i'^{\mathrm{T}} \boldsymbol{h}_j^c$
Bahdanau Normado	$\boldsymbol{e}_{i,j} = g\frac{v_a}{\|v_a\|} \tanh(\boldsymbol{W}_a[\tilde{\boldsymbol{h}}_{i-1}, \boldsymbol{h}_j^c])$
Luong Escalado*	$\boldsymbol{e}_{i,j} = g\tilde{\boldsymbol{h}}_i'^{\mathrm{T}} \boldsymbol{h}_j^c$

Siendo exactos con la literatura

[ ] Neural Machine Translation by Jointly Learning to Align and Translate, Bahdanau, 2014.
[ ] Online and linear-time attention by enforcing monotonic alignments, Raffel, 2017.
[ ] Effective Approaches to Attention-based Neural Machine Translation, Luong, 2015
[ ] Attention is all you need, Vaswani, 2017.
[ ] Tensorflow Attention https://www.tensorflow.org/api_docs/python/tf/contrib/seq2seq/AttentionWrapper

8. Attention / Self Attention



Atención en *Machine Translation* en el paso t=3

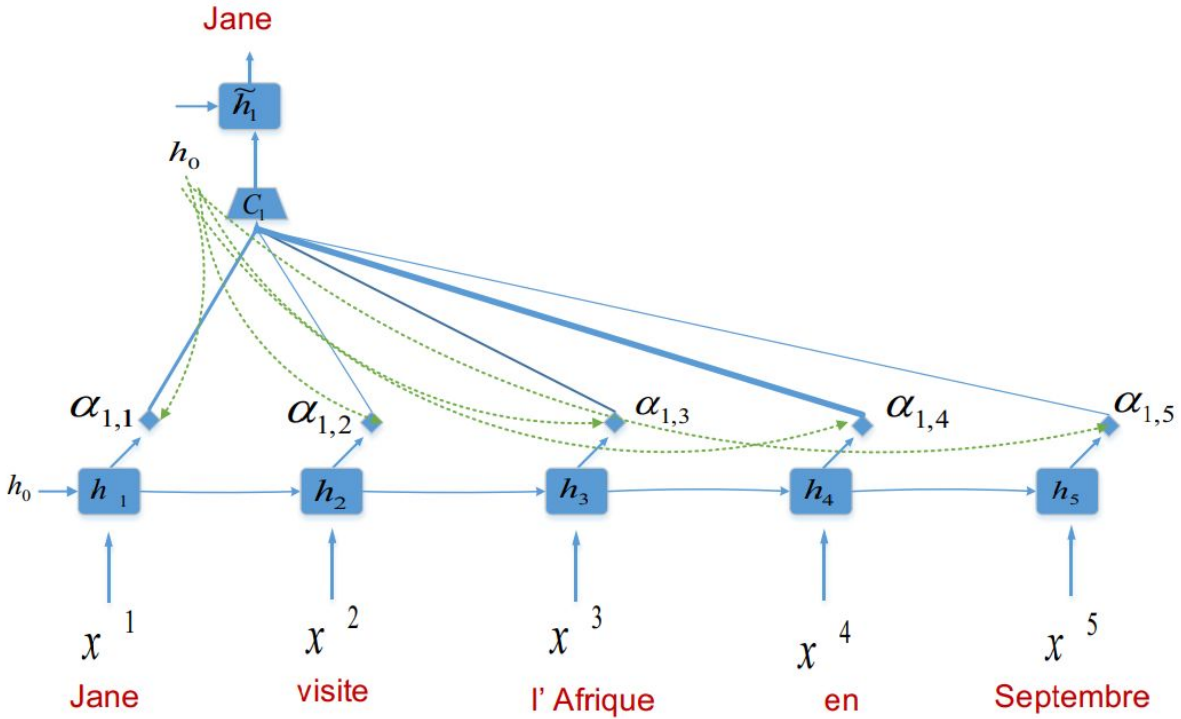
8. Attention / Self Attention

Expresiones	
Expresión	Add insult to injury.
Significado	To make a bad situation worse.
Traducción equivocada	Agregar un insulto a la herida.
Traducción correcta	De mal en peor.
Expresión	Barking up the wrong tree.
Significado	To look for a solution in the wrong place.
Traducción equivocada	Ladrar al árbol equivocado.
Traducción correcta	Buscas algo en donde no existe
Expresión	Bite the bullet.
Significado	To get something over with because it is inevetable.
Traducción equivocada	Morder la bala.
Traducción correcta	Hacer algo si o si.

Ejemplo donde fallaría atención para *Machine Translation*

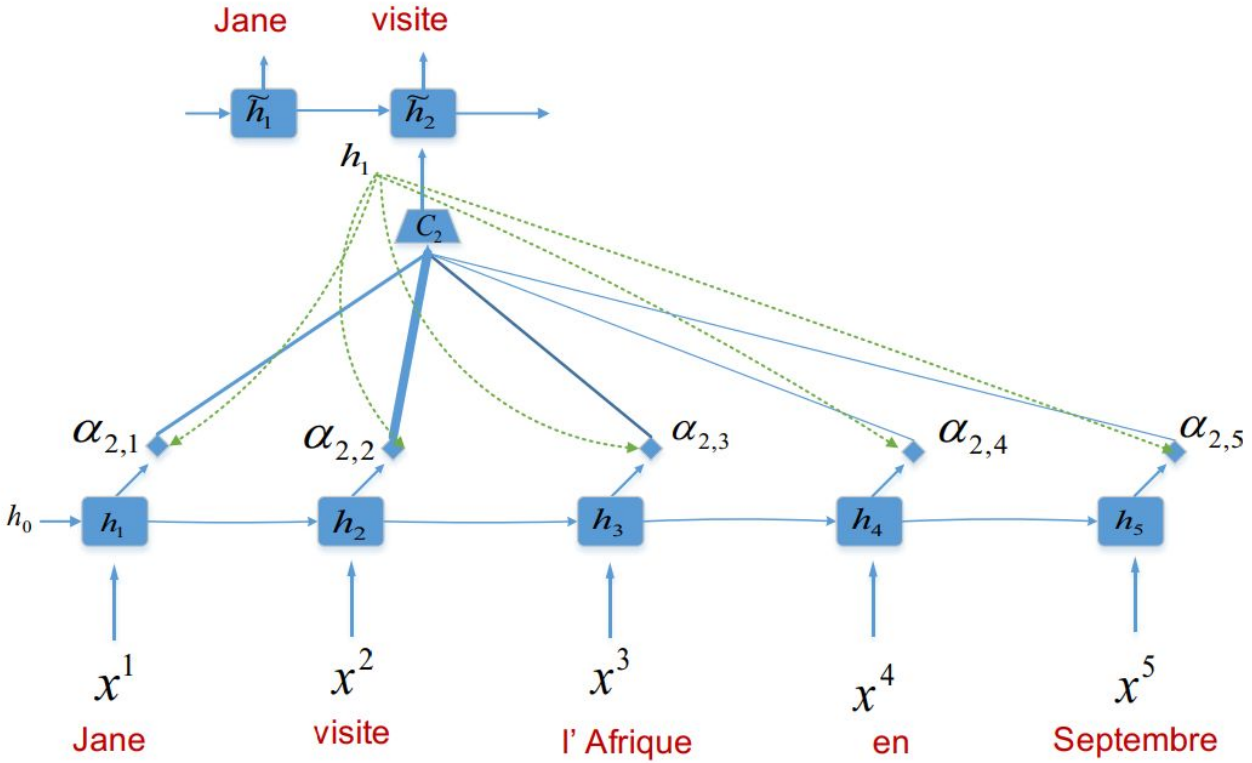
[ ] Neural Machine Translation for Low Resource Languages, Bragagnini, 2018.

8. Attention / Self Attention



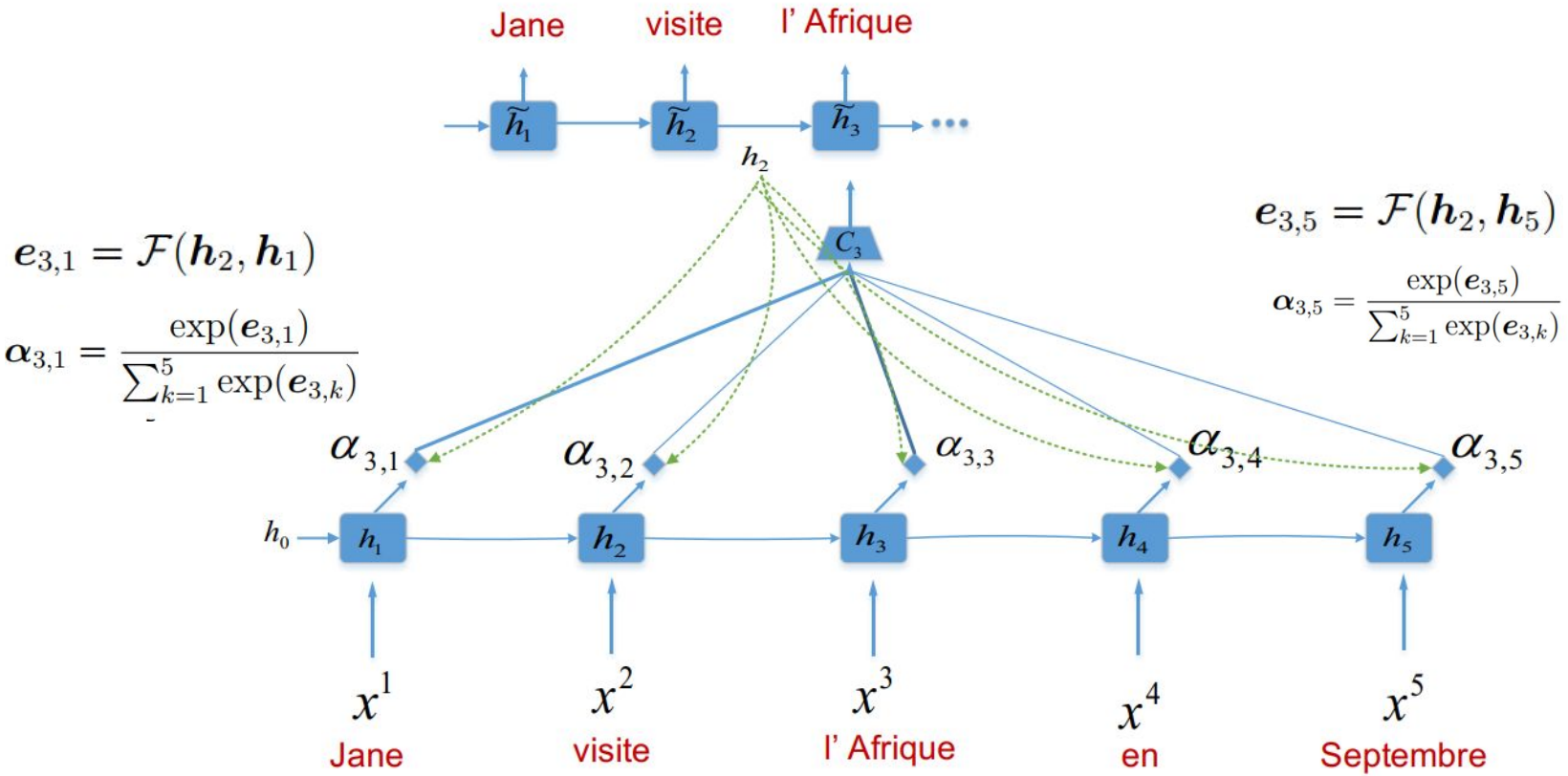
Self Attention en el paso t=1

8. Attention / Self Attention



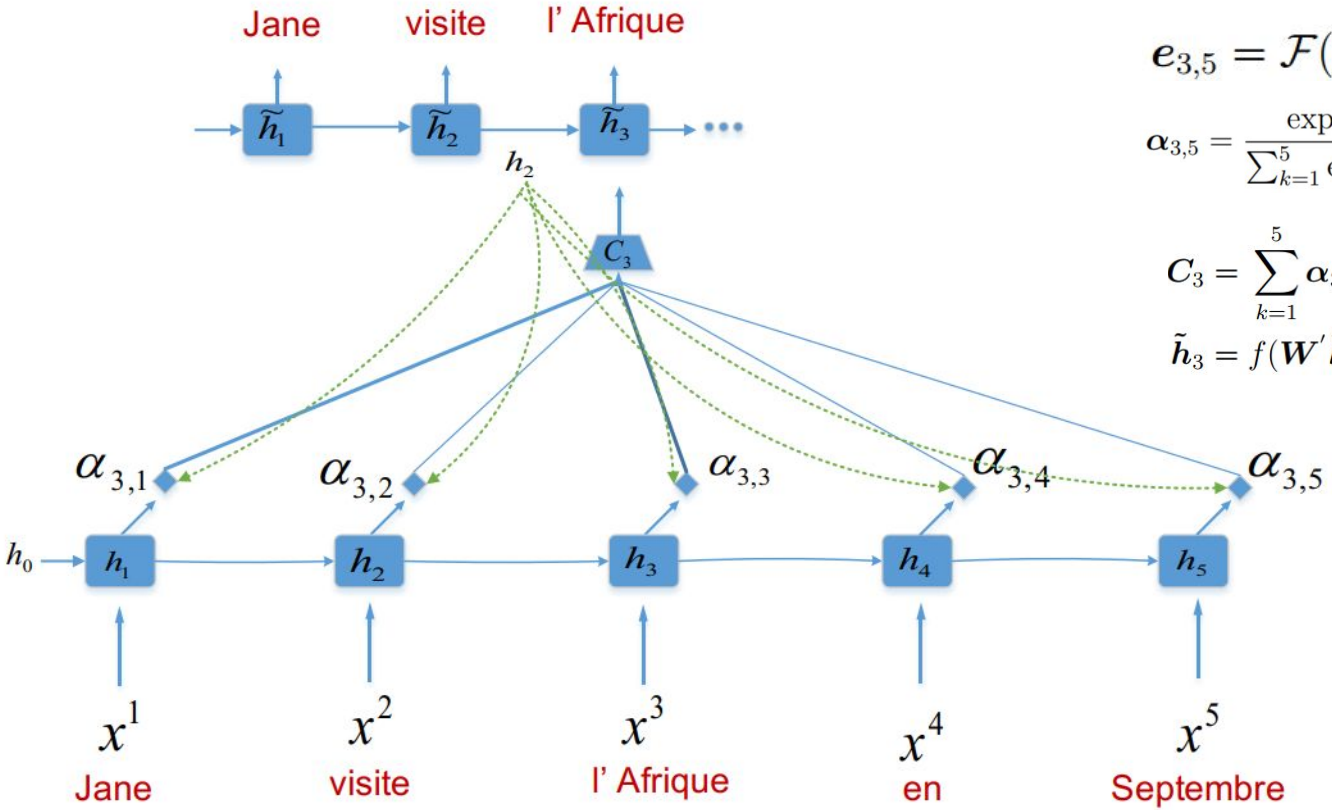
Self Attention en el paso t=2

8. Attention / Self Attention



Self Attention en el paso t=3

8. Attention / Self Attention



$$e_{3,5} = \mathcal{F}(h_2, h_5)$$

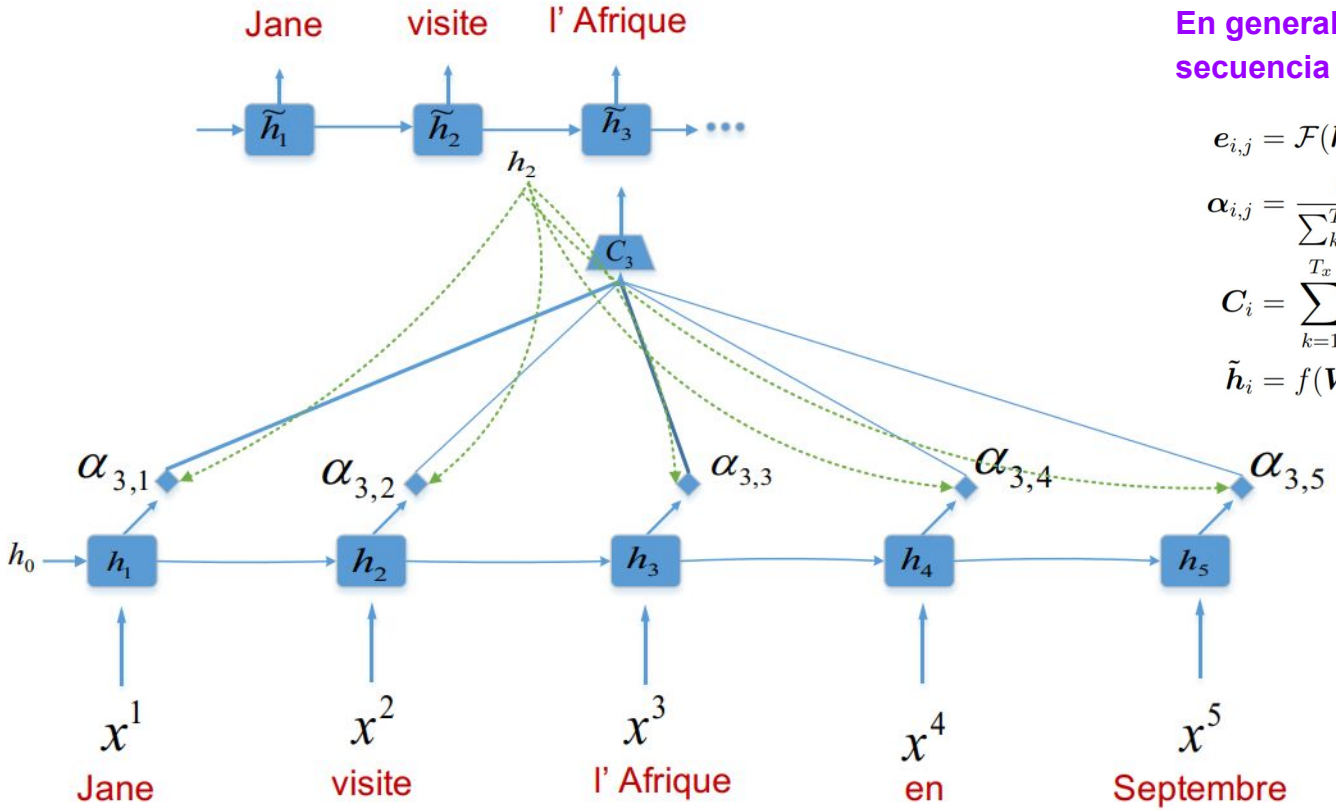
$$\alpha_{3,5} = \frac{\exp(e_{3,5})}{\sum_{k=1}^{5} \exp(e_{3,k})}$$

$$C_3 = \sum_{k=1}^{5} \alpha_{3,k} h_k$$

$$\tilde{h}_3 = f(W' \tilde{h}_2 + W_c C_3 + b_{eh})$$

Self Attention en el paso t=3

8. Attention / Self Attention



Self Attention en el paso t=3

En general para t = i y una secuencia Tx

$$e_{i,j} = \mathcal{F}(h_{i-1}, h_j)$$

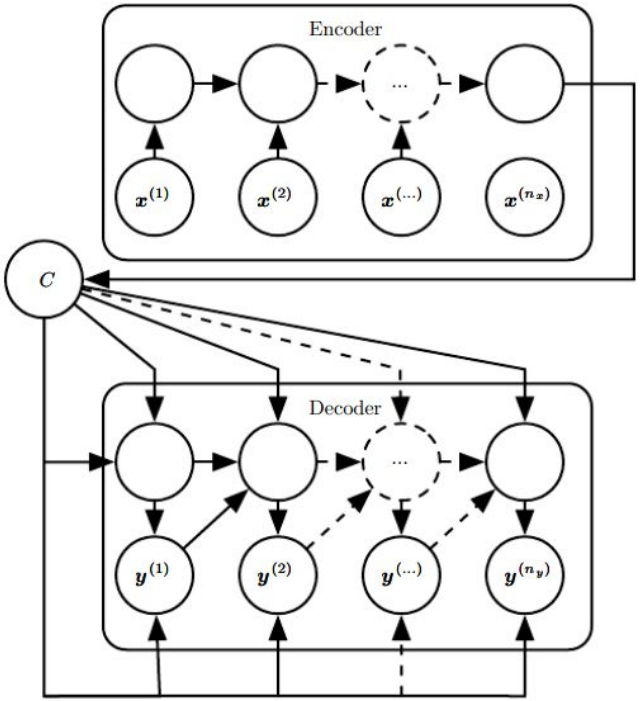
$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{k=1}^{T_x} \exp(e_{i,k})}$$

$$C_i = \sum_{k=1}^{T_x} \alpha_{i,k} h_k$$

$$\tilde{h}_i = f(W' \tilde{h}_{i-1} + W_c C_i + b_{eh})$$

9. Machine Translation

9.1 Encoder Decoder



Encoder-Decoder model for sequence-to-sequence

encoder

decoder

learning

$$\prod_{t=1} p(\boldsymbol{y}_t \mid \boldsymbol{c}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{t-1})$$

[ ] Deep Learning, Ian Goodfellow, Yoshua Bengio and Aaron Courville, 2016.

9. Machine Translation

9.1 Encoder Decoder

encoder

$$
\begin{aligned}
\boldsymbol{h}_t &= f(\boldsymbol{W}[e(\boldsymbol{x}_t), \boldsymbol{h}_{t-1}] + \boldsymbol{b}_e) \\
\vdots &= \quad \vdots \\
\boldsymbol{h}_{n_x} &= f(\boldsymbol{W}[e(\boldsymbol{x}_{n_x}), \boldsymbol{h}_{n_x-1}] + \boldsymbol{b}_e) \\
\boldsymbol{c} &= g(\boldsymbol{V}\boldsymbol{h}_{n_x})
\end{aligned}
$$

decoder

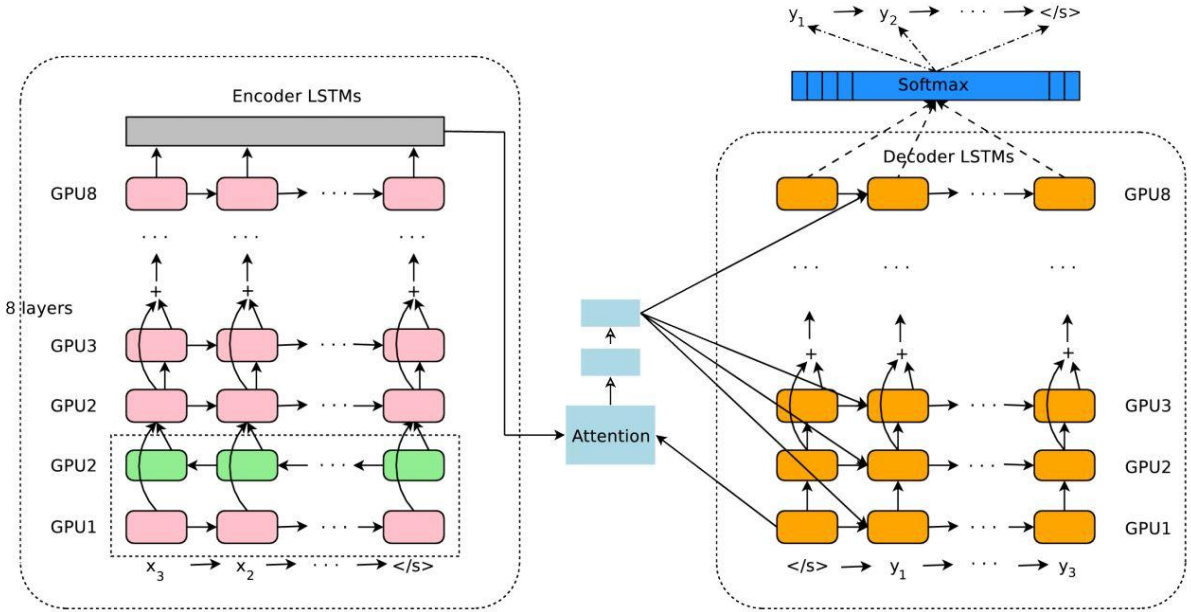
$$
\begin{aligned}
\boldsymbol{h}'_0 &= \tanh(\boldsymbol{V}'\boldsymbol{c}) \\
\boldsymbol{h}'_{t'} &= f(\boldsymbol{W}'\boldsymbol{h}'_{t'-1} + \boldsymbol{U}e(\boldsymbol{y}_{t'-1}) + \boldsymbol{C}\boldsymbol{c} + \boldsymbol{b}_{dh}) \\
\boldsymbol{s}_{t'} &= o(\boldsymbol{O}_h\boldsymbol{h}'_{t'} + \boldsymbol{O}_y e(\boldsymbol{y}_{t'-1}) + \boldsymbol{O}_c\boldsymbol{c} + \boldsymbol{b}_{do})
\end{aligned}
$$

[ ] Deep Learning, Ian Goodfellow, Yoshua Bengio and Aaron Courville, 2016.

9. Machine Translation

9.2 Google Neural Machine Translation

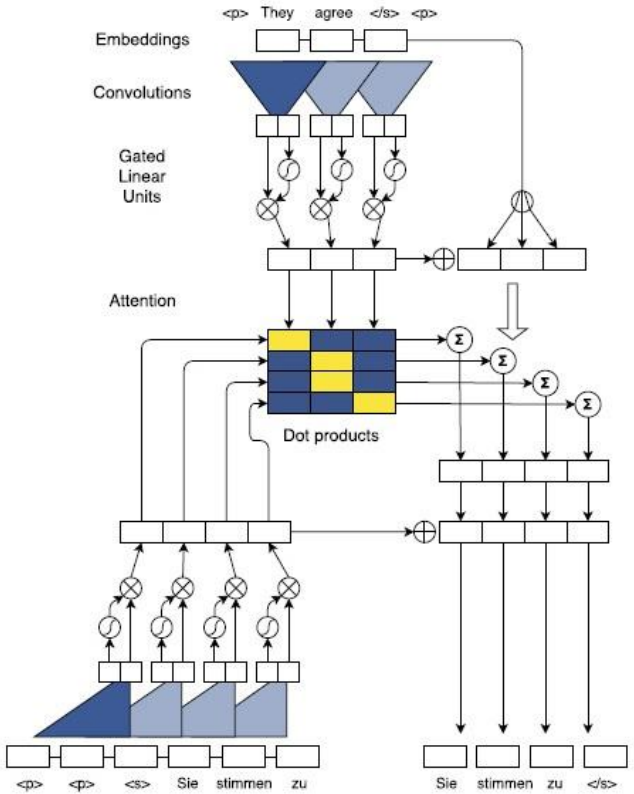
Architecture based Google Neural Machine Translation: Encoder (stack RNN with first layer BiRNN + residual connections) + Attention Mechanics + Decoder(stack RNN + residual connections) + ***Special Technique of Transfer Learning***.



[ ] Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, Y. Wu, M. Schuster, Z. Chen, Q.V. Lec, M. Norouzi, 2016

9. Machine Translation

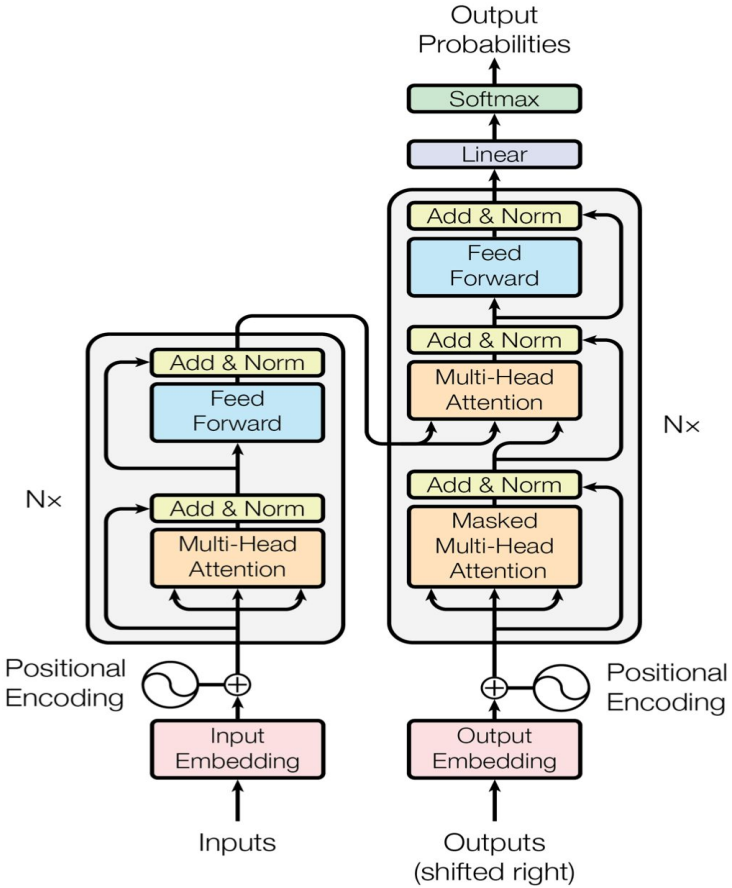
9.3 Conv2Seq



[ ] Convolutional Sequence to Sequence Learning, Jonas Gehring, 2017.

9. Machine Translation

9.4 Transformer

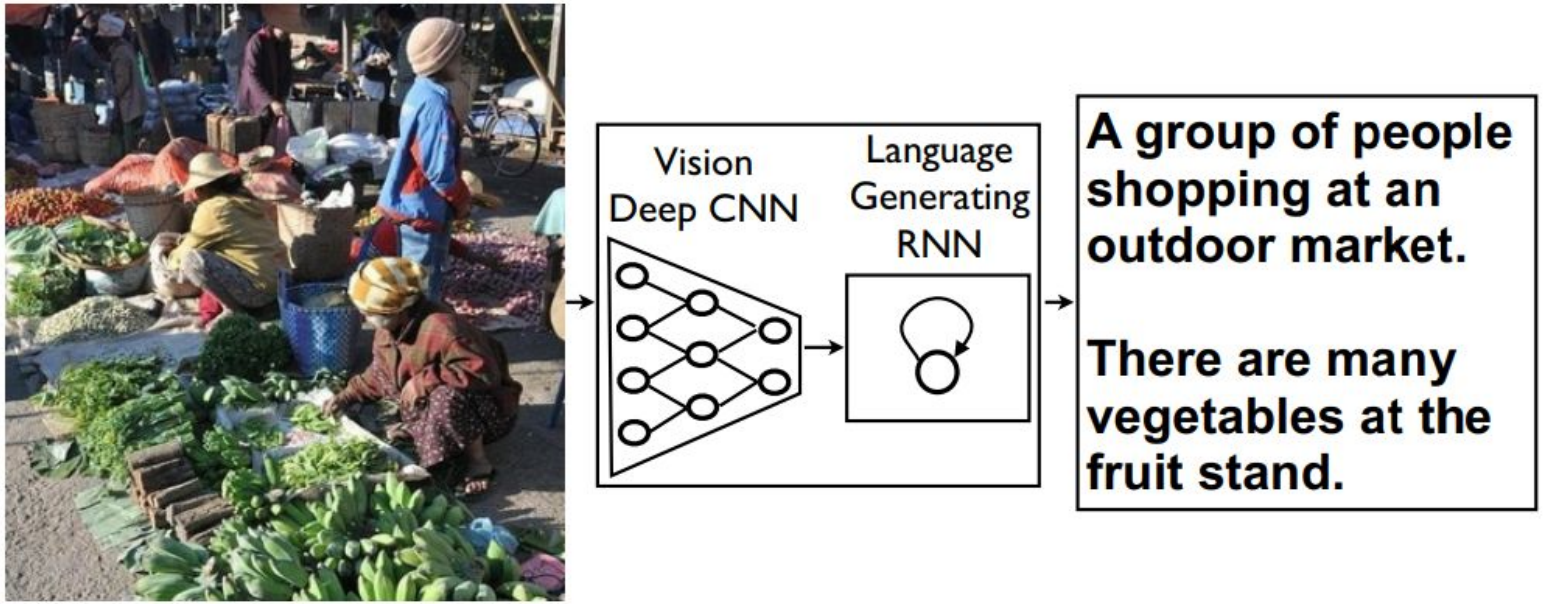


[ ] Attention is All you Need, Vaswani, 2017.

10. Image Caption

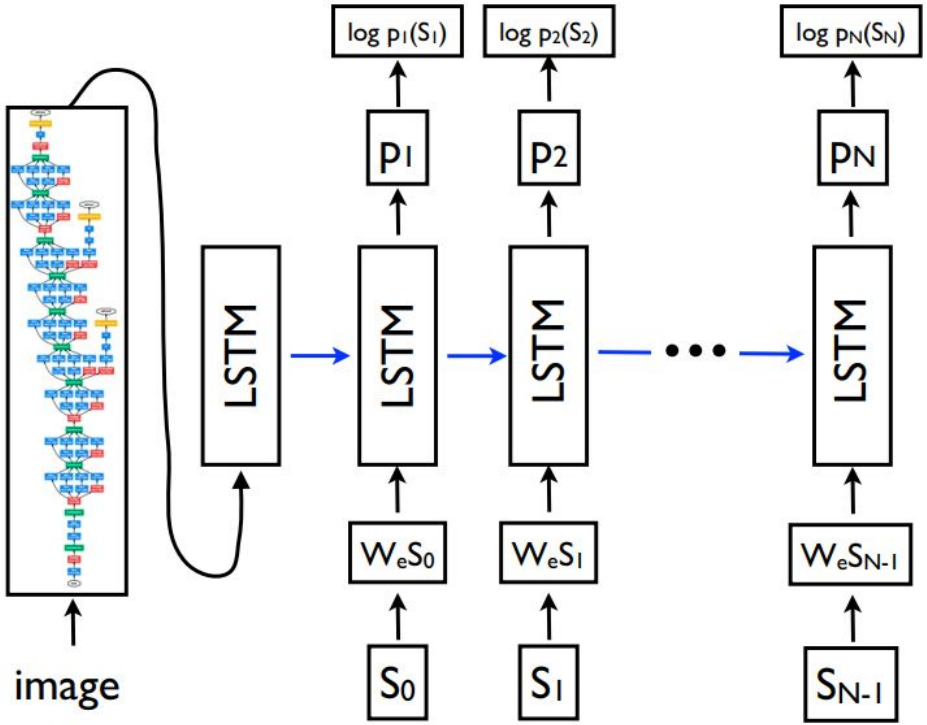
[ ] Show and Tell: A Neural Image Caption Generator, Vinyals, 2015

10. Image Caption



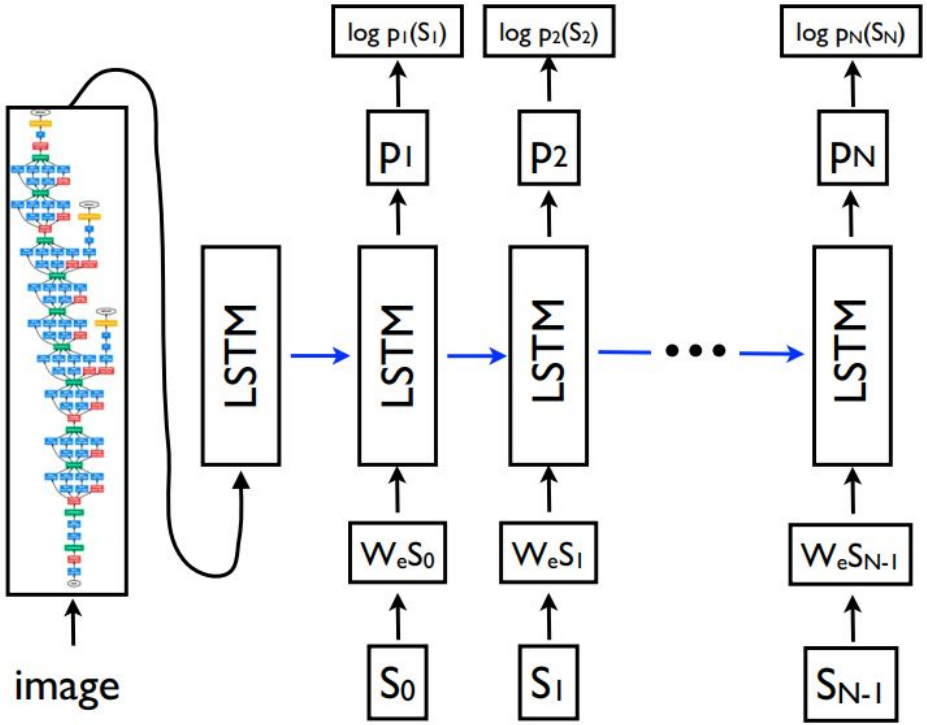
[ ] Show and Tell: A Neural Image Caption Generator, Vinyals, 2015

10. Image Caption



[ ] Show and Tell: A Neural Image Caption Generator, Vinyals, 2015

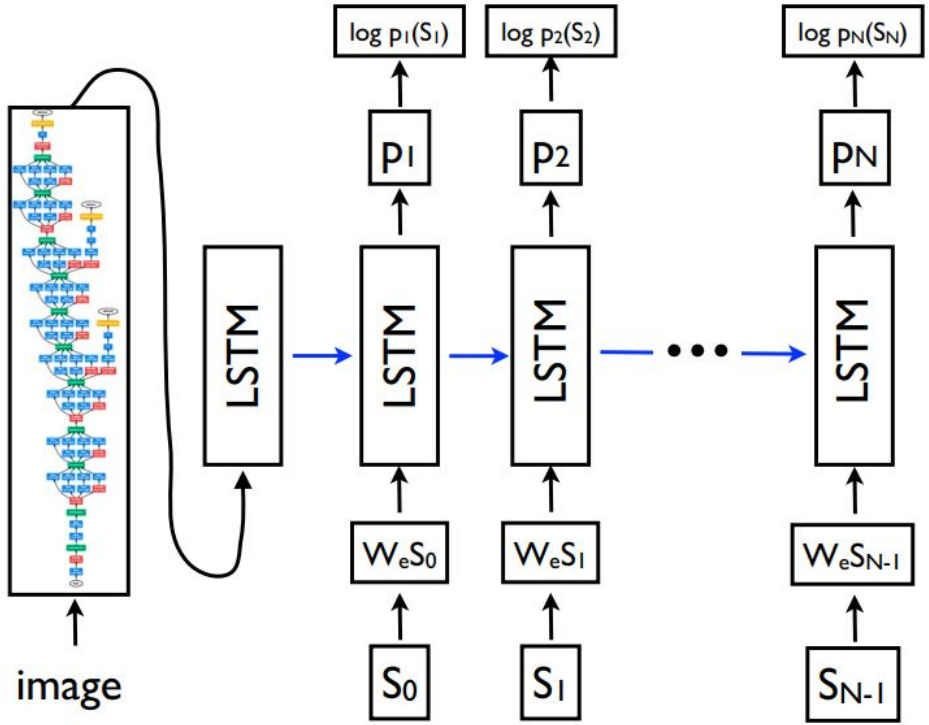
10. Image Caption



Trouble?

[ ] Show and Tell: A Neural Image Caption Generator, Vinyals, 2015

10. Image Caption



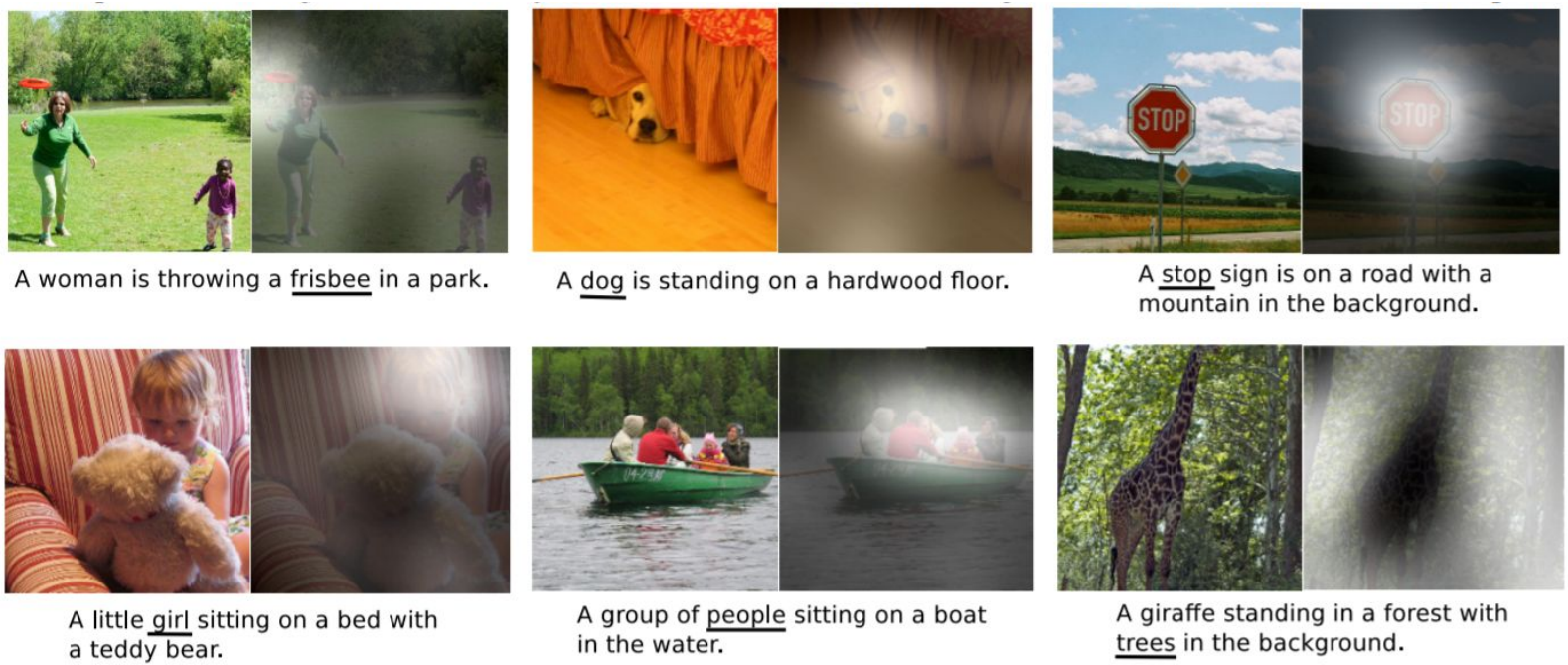
Trouble?

Problem: A limitation of the Encoder-Decoder architecture is that a single fixed-length representation is used to hold the extracted features.

[ ] Show and Tell: A Neural Image Caption Generator, Vinyals, 2015

10. Image Caption

With Attention



Examples of attending to the correct object (white indicates the attended regions, underlines indicated the corresponding word)

[ ] Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, Xu, 2015

11. Question Answering / Machine Comprehension / Chatbots

Nikola Tesla

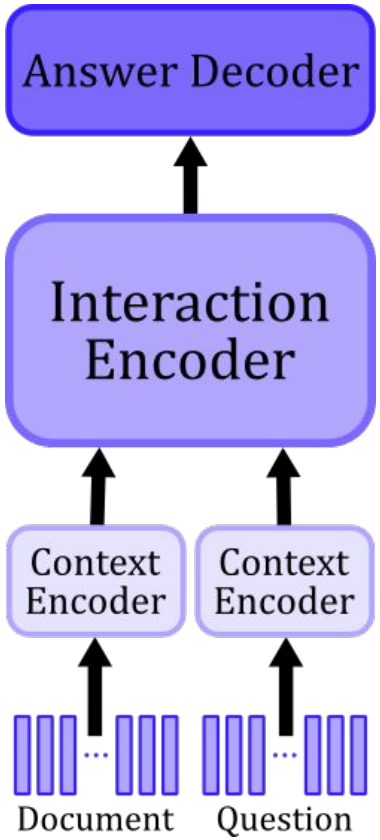
In 1870, Tesla moved to Karlovac, to **attend school at the Higher Real Gymnasium**, where he was profoundly influenced by a math teacher **Martin Sekulic**. The classes were held in **German**, as it was a school within the Austro-Hungarian Military Frontier. Tesla was able to perform integral calculus in his head, which prompted his teachers to believe that he was cheating. He finished a four-year term in three years, graduating in 1873.

In what language were the classes given?	**German**
Who was Tesla's main influence in Karlovac?	**Martin Sekulic**
Why did Tesla go to Karlovac?	**attend school at the Higher Real Gymnasium**

Pares de Question Answering extraidos de la base Stanford Question Answering Dataset (SQuAD).

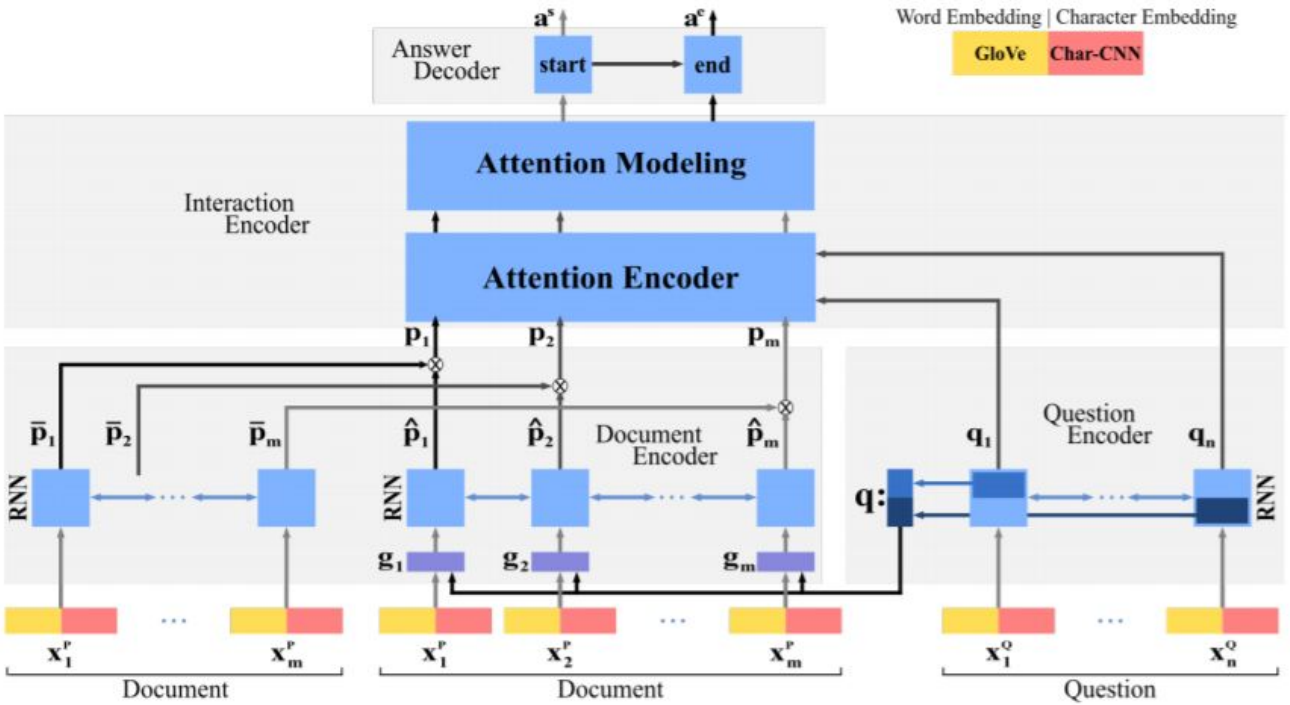
[ ] Fine-Grained Gating based on Question-Summary for Machine Comprehension, Mayhua, 2018.

11. Question Answering / Machine Comprehension / Chatbots



[ ] Fine-Grained Gating based on Question-Summary for Machine Comprehension, Mayhua, 2018.

11. Question Answering / Machine Comprehension / Chatbots



[ ] Fine-Grained Gating based on Question-Summary for Machine Comprehension, Mayhua, 2018.

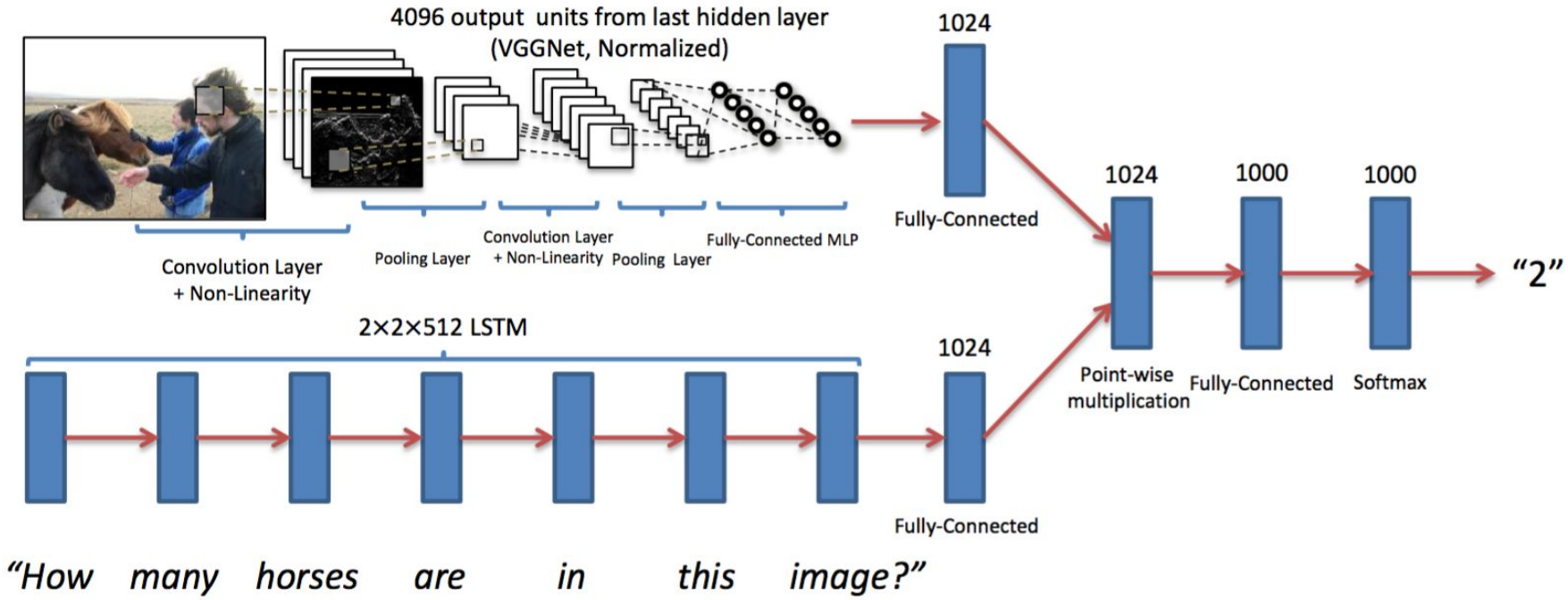
12. Visual Question Answering

What color are her eyes?
What is the mustache made of?

How many slices of pizza are there?
Is this a vegetarian pizza?

[ ] VQA: Visual Question Answering, Agrawal, Lu, Antol, 2017.

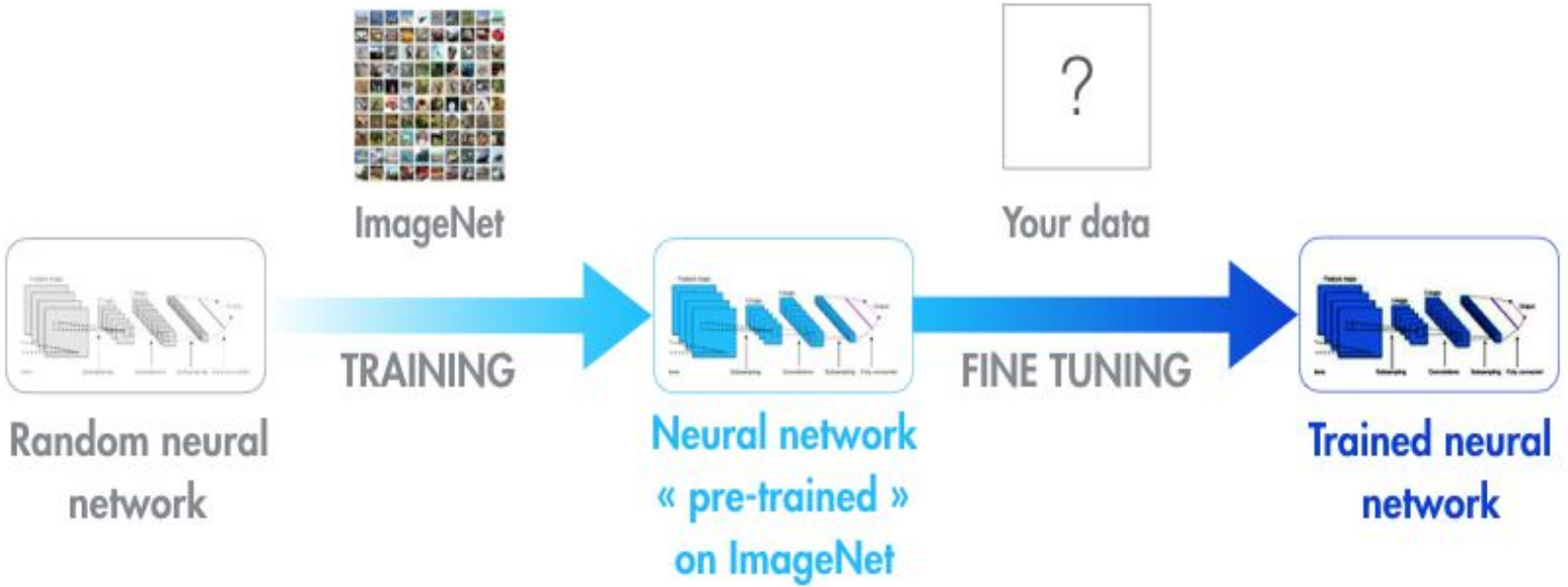
12. Visual Question Answering



[ ] VQA: Visual Question Answering, Agrawal, Lu, Antol, 2017.

13. Transfer learning

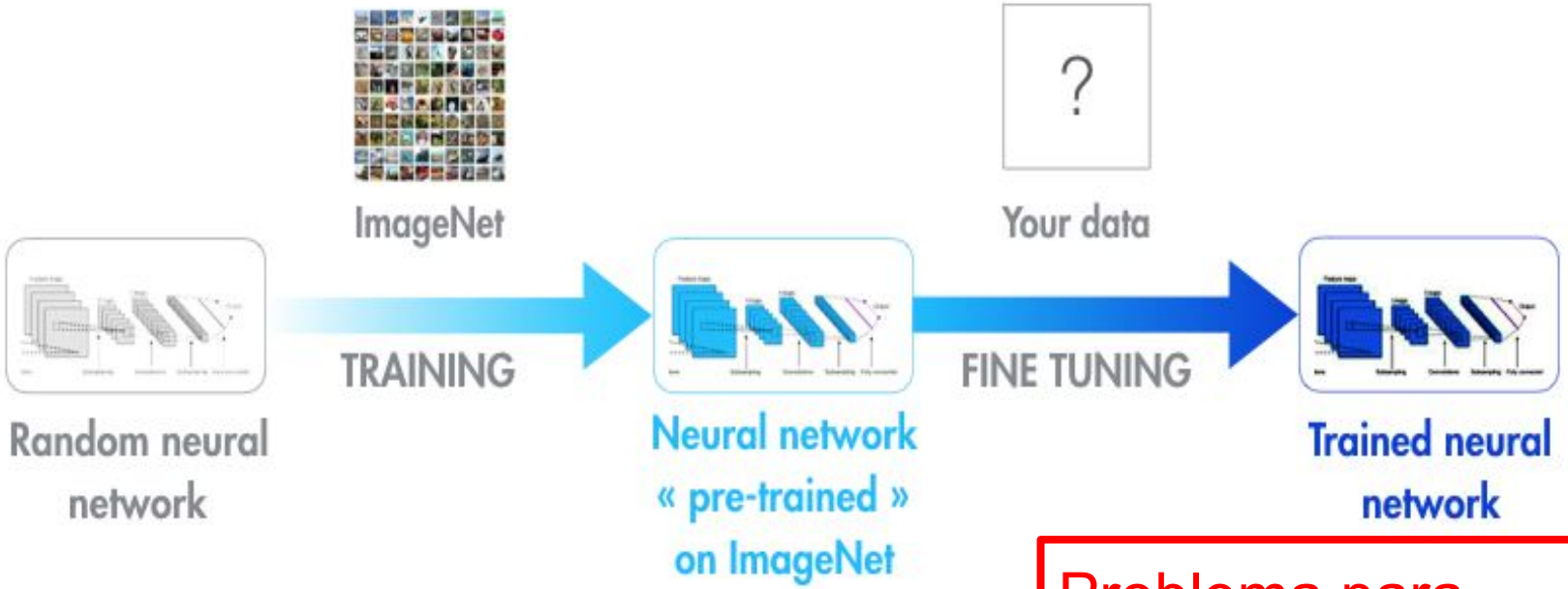
In Computer Vision



[ ] VQA: Visual Question Answering, Agrawal, Lu, Antol, 2017.

13. Transfer learning

In Computer Vision



Problema para NLP?

[ ] Medium, An Introduction to Transfer Learning in Machine Learning, Curry, 2018.

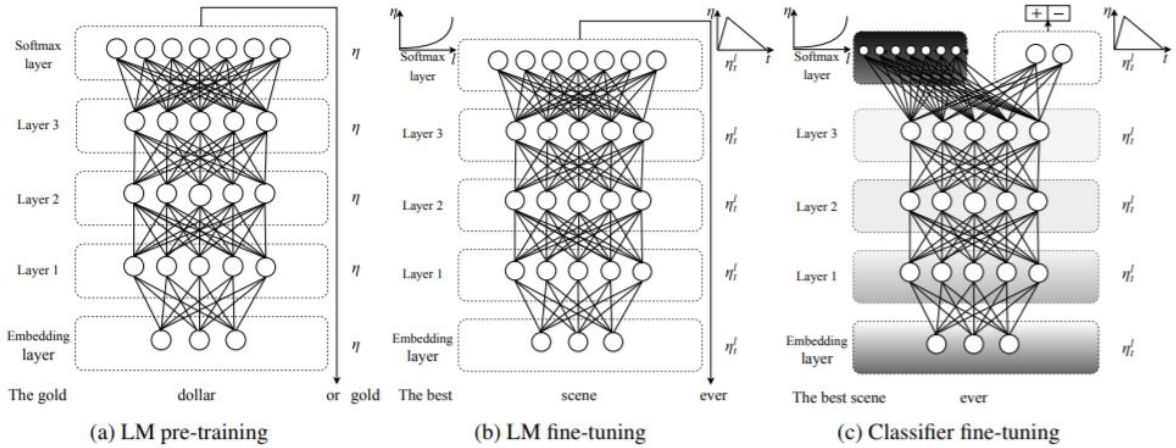
13. Transfer learning

In Natural Language Processing

- No funciona bien usar el enfoque clásico de transfer learning de visión computacional.
- Cual es una arquitectura-concepto ImagetNet para NLP.

13. Transfer learning

In Natural Language Processing



Howard, presenta un enfoque de *transfer learning* exitoso para problemas de clasificación, usando un *Language Modeling* como *pretrained network*. **Su propuesta se centra en una modificación inteligente y dinámica del *learning rate*, sobre ciertas capas durante el entrenamiento**.

[ ] Universal Language Model Fine-tuning for Text Classification, Howard, 2018.

References

- Natural Language Processing with Deep Learning, Christopher Manning and Richard Socher, cs224n,2017.
- Word Embeddings, Christopher Olah.
- Natural Language Processing, Yves Peirsman.
- Word Embeddings, Tomas Mikolov.
- RNN, cs224d, Richard Socher, 2016.
- CS224N/Ling284, Lecture 8, Stanford University, Manning , 2016.
- Learning Long-Term Dependencies with Gradient Descent is Difficult, Bengio, 1994.
- Deep Learning, Ian Goodfellow, Yoshua Bengio, Aaron Courville, 2016.
- Advanced in Optimizing Recurrent Networks, Ian Goodfellow, Yoshua Bengio, Aaron Courville, 2016.
- Lecture note 04 in Deep Learning for Natural Language Processing, Mohammadi, 2015.
- Recurrent Neural Network Regularization, Wojciech Zarembam, Ilya Sutskever and Oriol Vinyals, 2014.
- Learning phrase representations using RNN encoder-decoder for statistical machine translation, Cho, 2014.
- Lecture note 04 in Deep Learning for Natural Language Processing, Mohammadi, 2015.

References

- Neural Machine Translation and Sequence-to-sequence Models: A Tutorial, Graham Neubig, 2017.
- A Neural Probabilistic Language Model, Yoshua Bengio, Réjean Ducharme, Pascal Vincent, Christian Jauvin, 2003.
- Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, Socher, 2013.
- Neural Machine Translation for Low Resource Languages, Bragagnini, 2018.
- Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, Y. Wu, M. huster, Z. Chen, Q.V. Lec, M. Norouzi, 2016
- Convolutional Sequence to Sequence Learning, Jonas Gehring, 2017.
- Attention is All you Need, Vaswani, 2017.
- Show and Tell: A Neural Image Caption Generator, Vinyals, 2015
- Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, Xu, 2015
- Fine-Grained Gating based on Question-Summary for Machine Comprehension, Mayhua, 2018.
- VQA: Visual Question Answering, Agrawal, Lu, Antol, 2017.
- Medium, An Introduction to Transfer Learning in Machine Learning, Curry, 2018.
- Universal Language Model Fine-tuning for Text Classification, Howard, 2018.
- http://www.iwml.iitbhu.ac.in/presentations/iWML-talk-samarth.pdf

I SPDL

Deep Learning for Natural Language Processing

César Bragagnini

cesarbrma91@gmail.com
@MarchBragagnini

Universidad Católica San Pablo

ML/DL meetup AQP