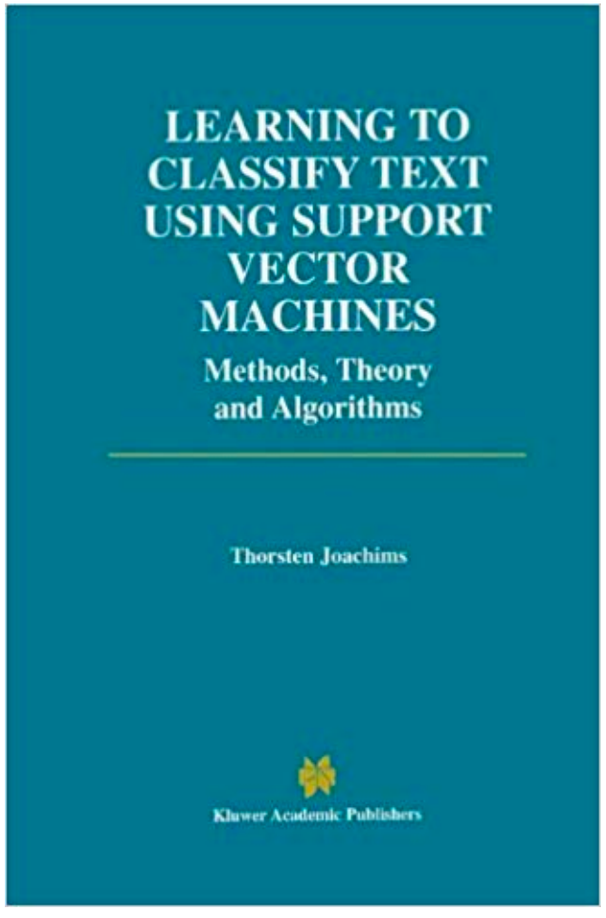
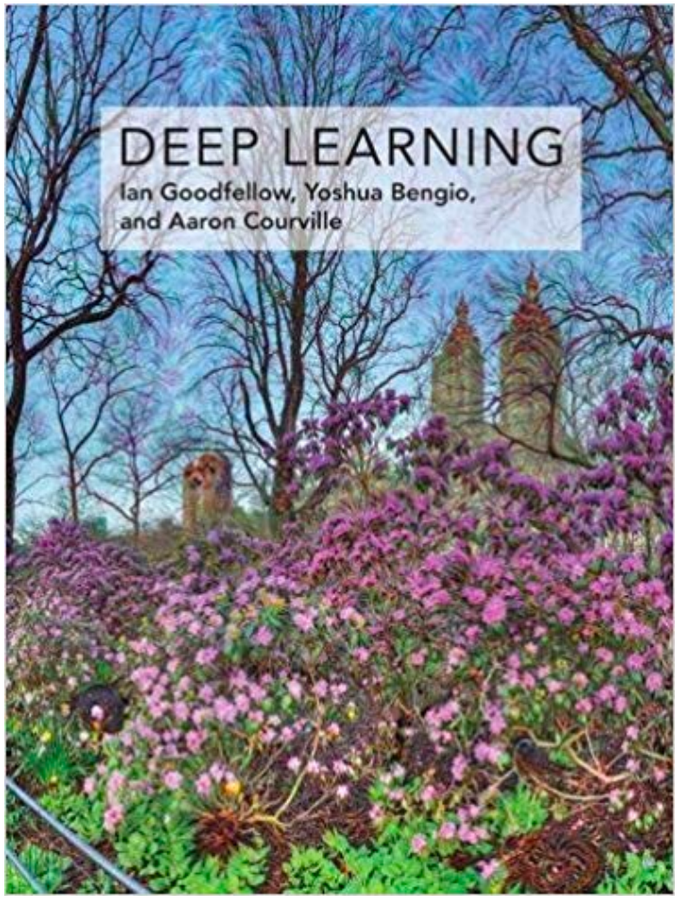
MODÈLES DE LANGUE



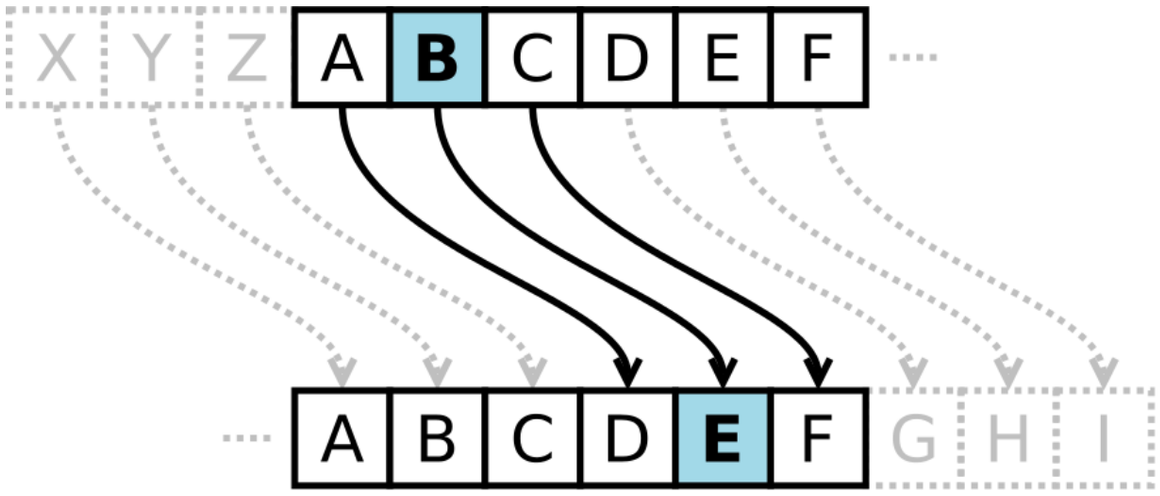


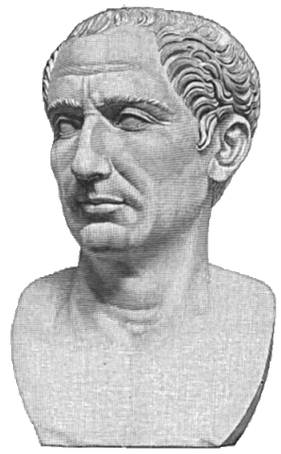
Emanuela Boros
boros@teklia.com

Christopher Kermorvant
kermorvant@teklia.com

Chiffrement par substitution

ZLNLSHGLD O'HQFBFORSHGLH OLEUH





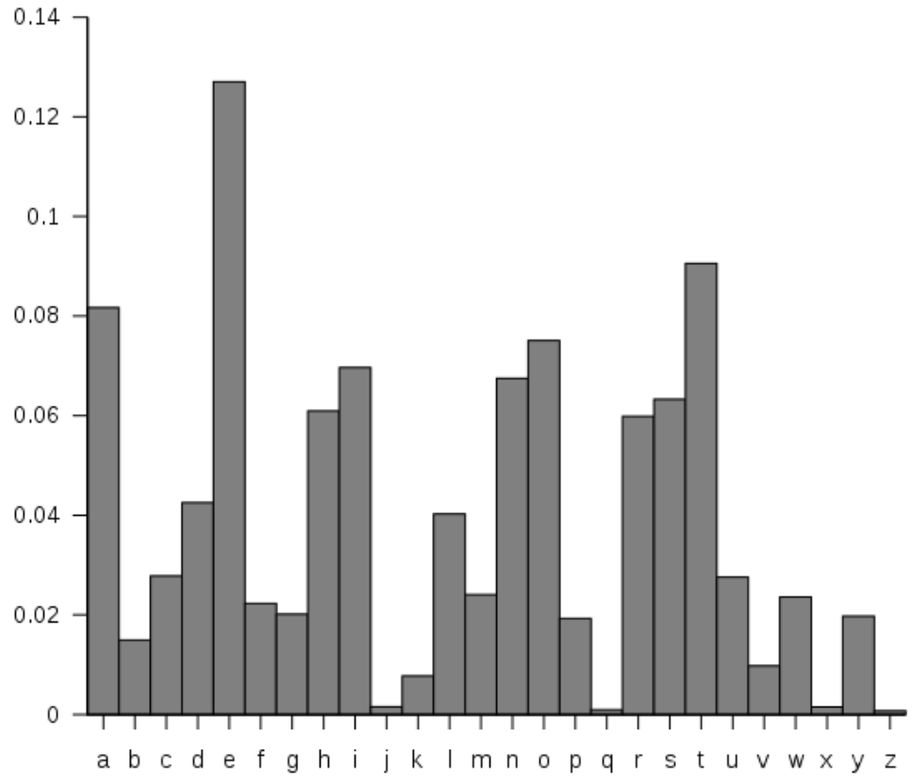
Code de César

WIKIPEDIA L'ENCYCLOPEDIE LIBRE

Cryptographie et modèles de langue

Décryptage par modélisation de la fréquence des lettres

LIVITCSWPIYVEWHEVSRIQMXLEYVEOIEWH
RXEXIPFEMVEWHKVSTYLXZIXLIKIIXPIJVSZE
YPERRGERIMWQLMGLMXQERIWGPSRIHMX
QEREKIETXMJTPRGEVEKEITREWHEXXLEXX
MZITWAWSQWXSWEXTVEPMRXRSJGSTVRI
EYVIEXCVMUIMWERGMIWXMJMGCSMWXS
JOMIQXLIVIQIVIXQSVSTWHKPEGARCSXRW



Hereupon Legrand arose, with a grave and stately air, and brought me the beetle from a glass case in which it was enclosed. It was a beautiful scarabaeus, and, at that time, unknown to naturalists—of course a great prize in a scientific point

+ fréquence des bigrams

+ fréquence des trigrams

Modèles de langue

Modélisent des séquences de mots ou caractères :

$$P(w_1, \ldots, w_T)$$

Permettent d'évaluer la probabilité d'une phrase :

- mot plus probable :

*P(*the, cat, sat, on,the, **mat**) > *P* (the, cat, sat , on, the, **yesterday**)

- ordre plus probable

*P(*the, cat, sat, on,the, mat) > *P*(the sat, cat the on, mat)

Modèles de langue par chaine de Markov

Hypothèse Markovienne : la probabilité du prochain symbole ne dépend que des *n* précédents :

$$P(w_1, \ldots, w_m) = \prod_{i=1}^{m} P(w_i \mid w_1, \ldots, w_{i-1})$$

$$\approx \prod_{i=1}^{m} P(w_i \mid w_{i-(n-1)}, \ldots, w_{i-1})$$

Modèles de langue par chaines de Markov

Zerogram : $\frac{1}{|V|}$, |V| est la taille du vocabulaire

Unigram : probabilité des symboles

Bigram : probabilité des symboles étant donné le symbole précédent :

$$p(w_2|w_1) = \frac{\text{count}(w_1, w_2)}{\text{count}(w_1)}$$

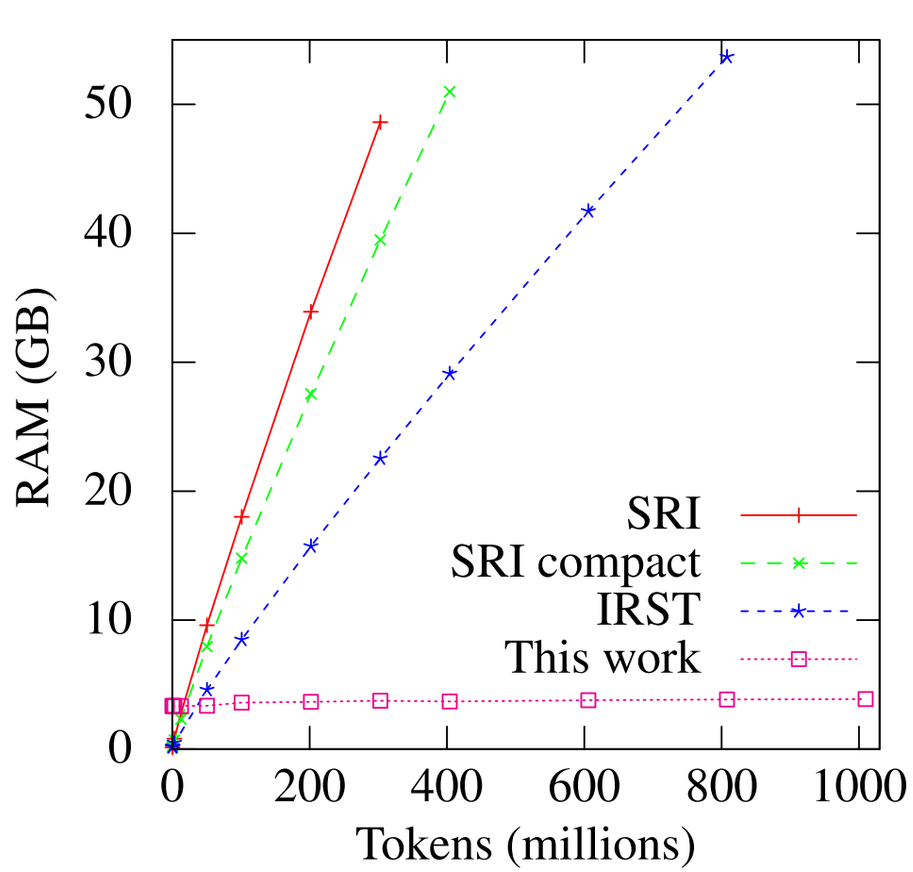
Trigram : probabilité des symboles étant donné les 2 symboles précédents :

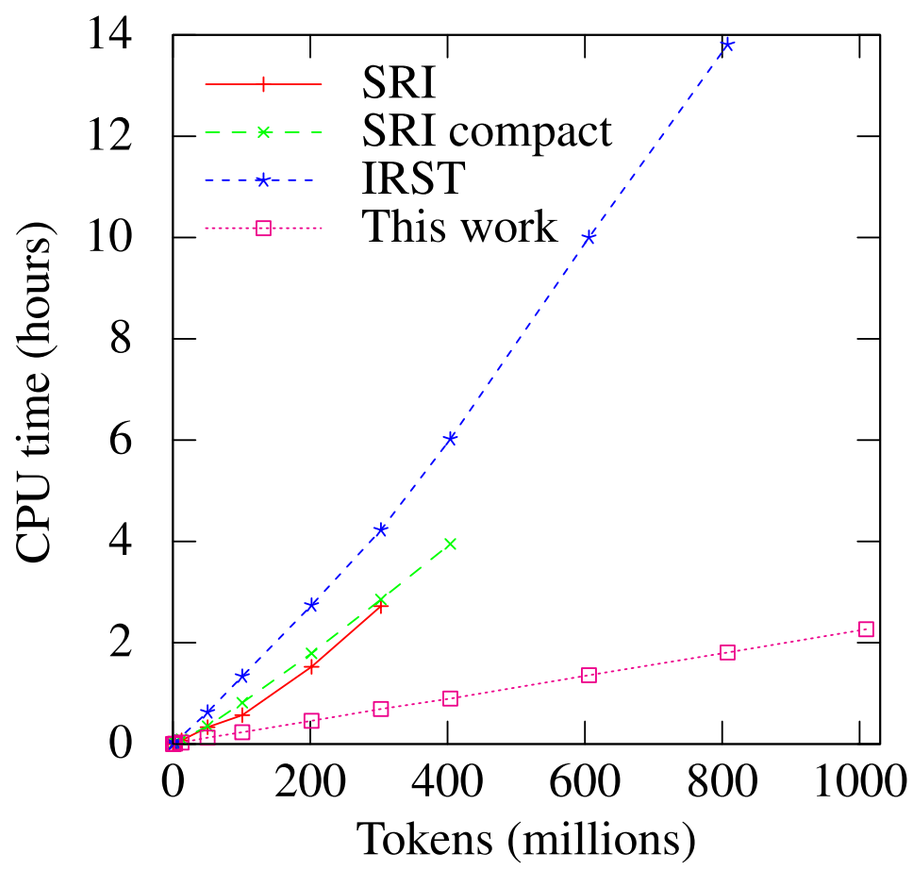
$$p(w_3|w_1, w_2) = \frac{\text{count}(w_1, w_2, w_3)}{\text{count}(w_1, w_2)}$$

Modèles de langue par chaines de Markov

Entrainement par comptage sur de très grands corpus de texte électronique :

Consommation mémoire et CPU importante





The ClueWeb09 Dataset : 1,040,809,705 de pages web

Heafield et al., Scalable Modified Kneser-Ney Language Model Estimation, 2013

Modèles de langue

Mesures de qualité

Cross-Entropie

$$H(w_1, ..., w_N) = -\frac{1}{N} \log_2 P(w_1, ..., w_N)$$

$$P(w_1, \ldots, w_m) = \prod_{i=1}^{m} P(w_i \mid w_1, \ldots, w_{i-1})$$

= combien de bit faut-il pour encoder la séquence de mot avec le modèle

Perplexité : $2^{H(w_1,\ldots,w_N)}$

= combien de mots sont possibles après un contexte donné

Modèles de langue

Lissage (smoothing)

$$P(w_1, ..., w_N) = \prod_{I=1}^{N} P(w_i | w_{i-(n-1)}, ..., w_{i-1})$$

Si une seule des probabilités est nulle, tous la séquence est nulle :

P(je bois un tesgüino avec mes amis) = 0

Il faut s'assurer que $P(w_i | \text{contexte})$ ne soit jamais nulle, quel que soit le contexte et w_i

Modèles de langue

Lissage (smoothing) : une solution possible

Interpolation :

$$\begin{aligned} p_I(w_n|w_{n-2}, w_{n-1}) &= \lambda_3 p(w_n|w_{n-2}, w_{n-1}) + \\ &\quad \lambda_2 p(w_n|w_{n-1}) + \\ &\quad \lambda_1 p(w_n). \end{aligned}$$

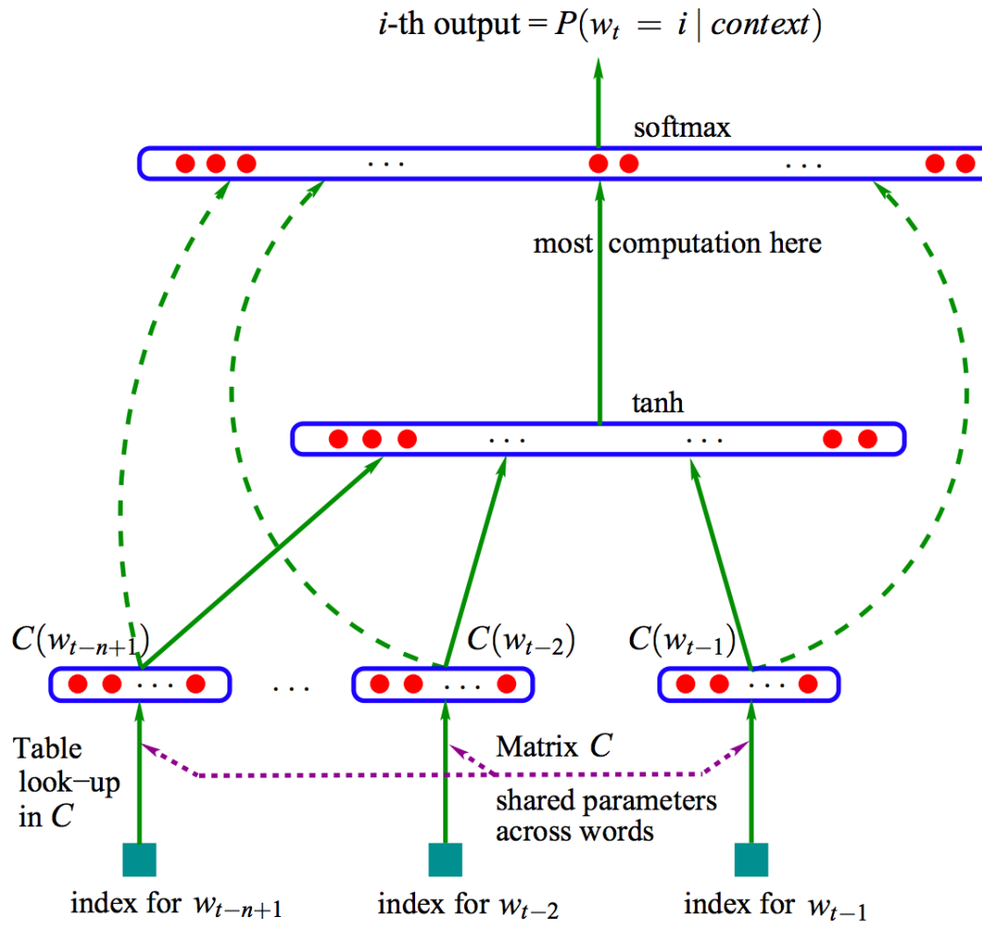
$$\lambda_3 + \lambda_2 + \lambda_1 = 1.$$

De multiples méthodes ont été proposées :

An empirical study of smoothing techniques for language modeling. Stanley Chen and Joshua Goodman, 1998.

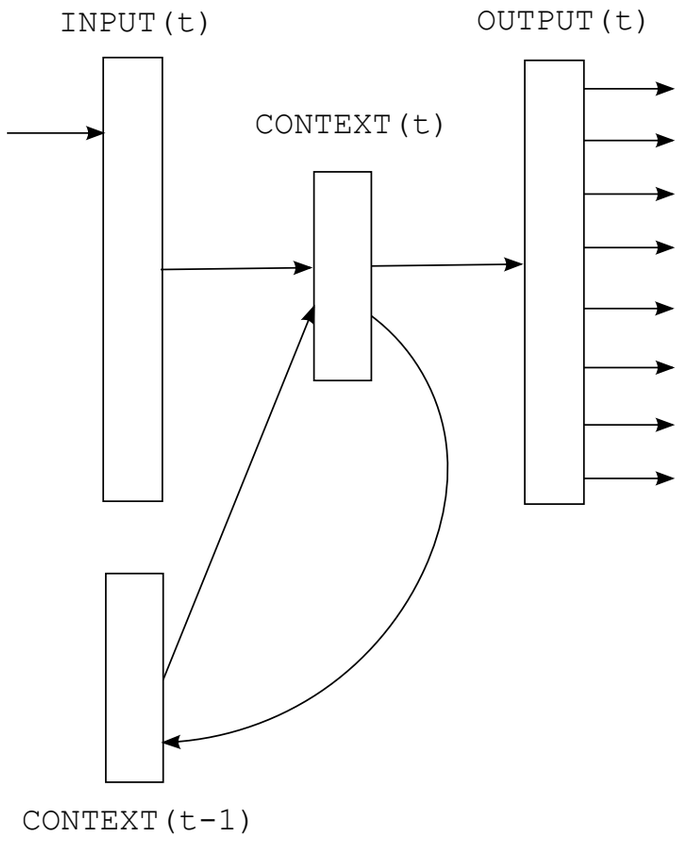
Modèles de langue neuronaux

Les précurseurs :



Bengio et al, a Neural probabilistic language model, 2003

Limite : le contexte est fixe

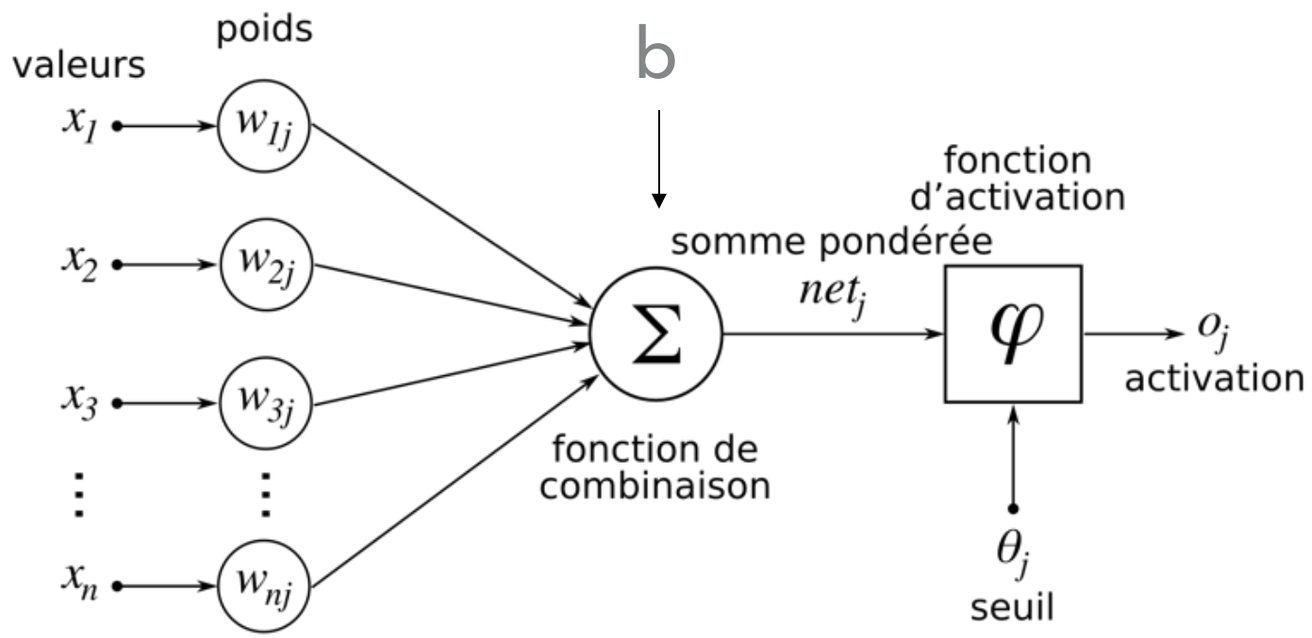


Mikolov et al, Recurrent neural network based language model, 2010

Réseaux de neurones

La régression logistique peut être vue comme un réseau de neurones très simple :

$$p(x) = \frac{1}{1 + e^{-(b+Wx)}}$$

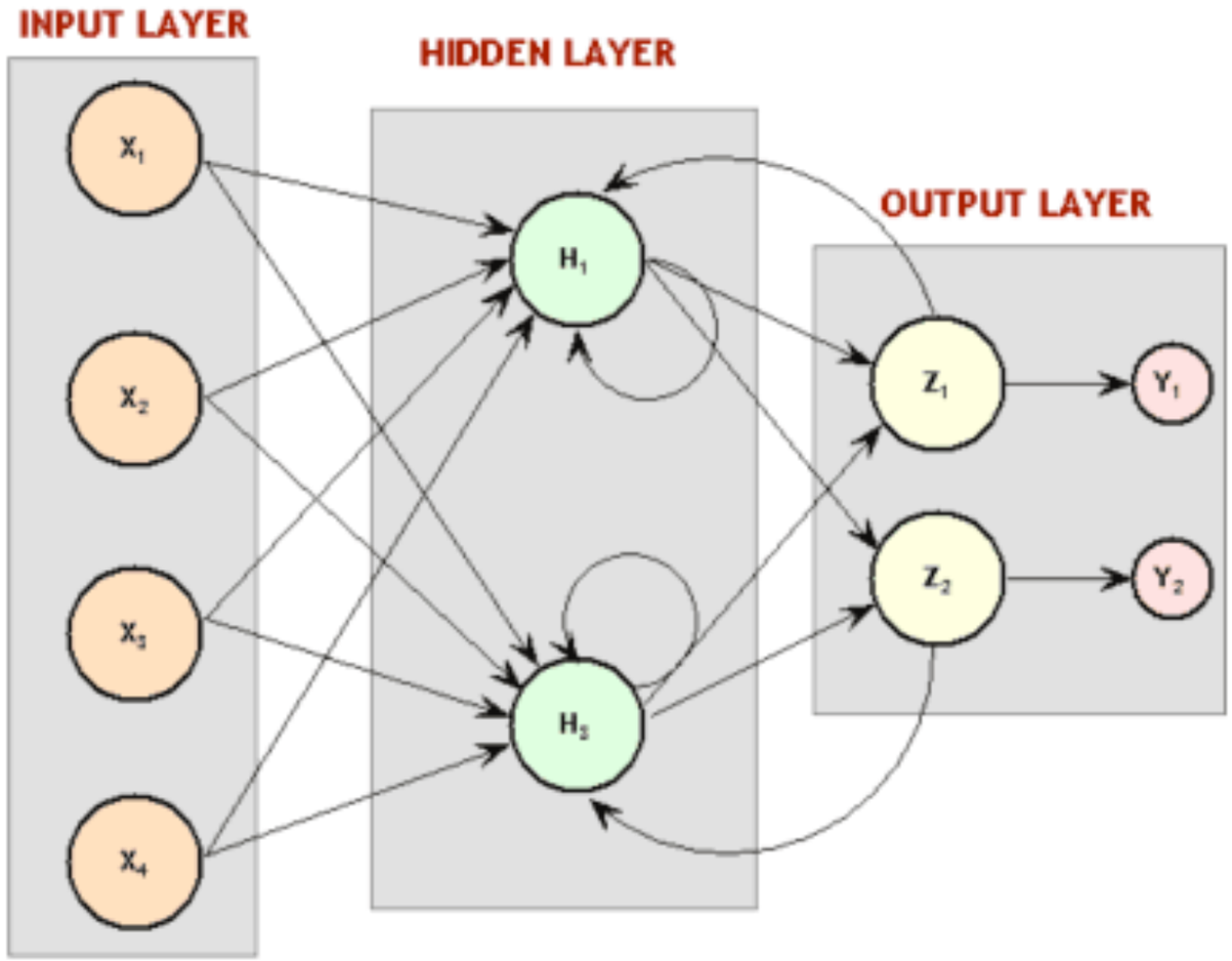


Réseaux de neurones récurrents

- Famille de réseaux de neurones permettant le traitement de séquences
- Les réseaux récurrents permettent de faire des prédictions pour chaque élément de la séquence d'entrée (*time step*)
- Les paramètres sont partagés entre les éléments à chaque *time step*

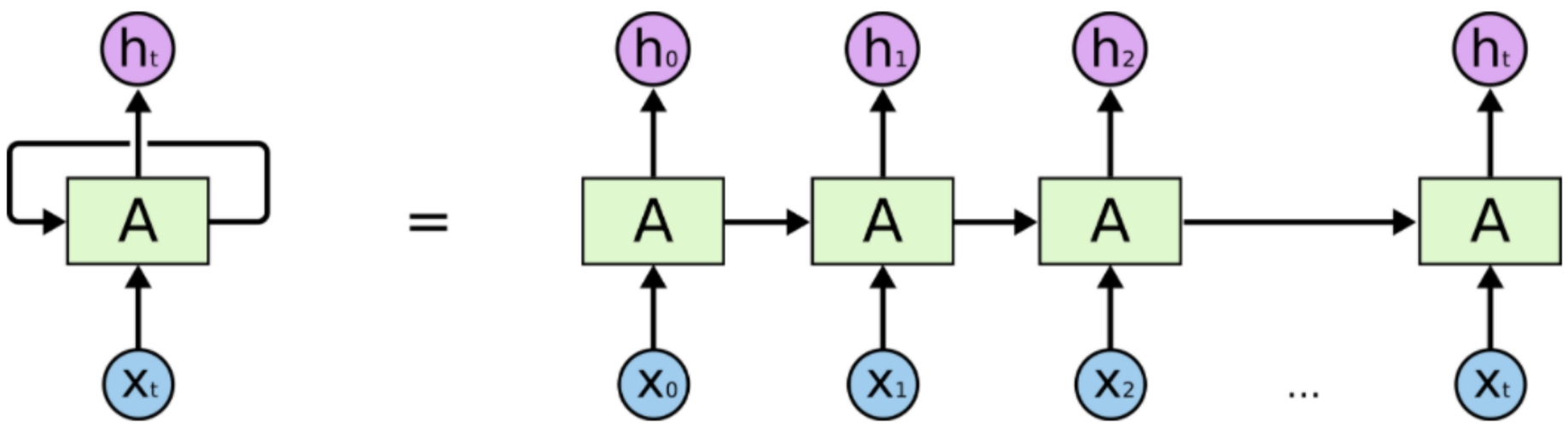
Réseaux de neurones récurrents

Les réseaux récurrents incluent des connexions entres les neurones des couches cachées et/ou entre les neurones de sortie et les neurones des couches cachées

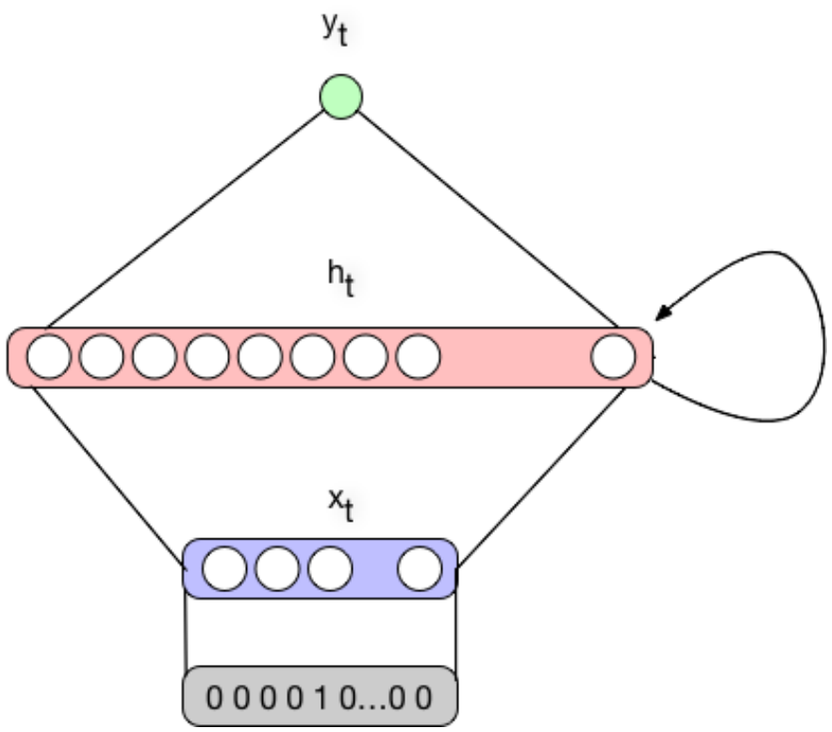


Réseaux de neurones récurrents

On peut « dérouler » un neurone récurrent selon le temps : à chaque pas de temps i , le même neurone est utilisé pour faire une prédiction en prenant en compte l'entrée courante X_i et la sa sortie au pas précédent i-1



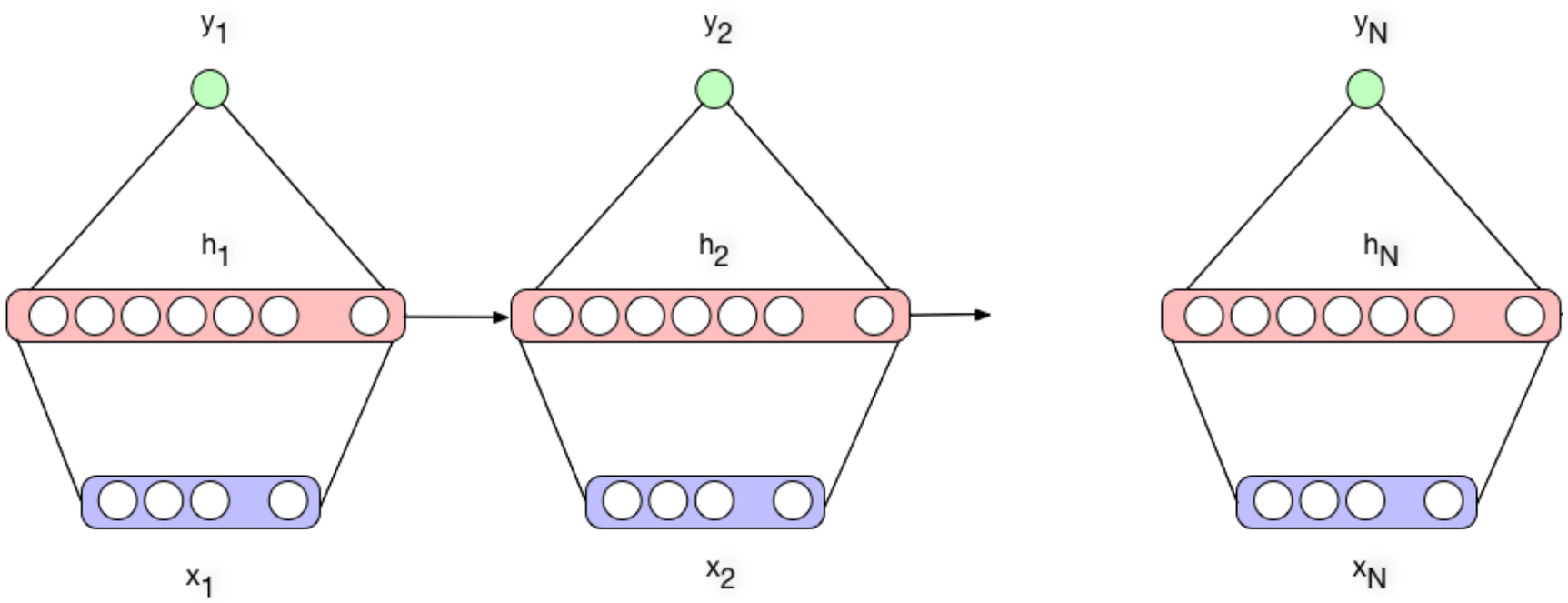
Réseaux de neurones récurrents



$$x_1, \ldots, x_{t-1}, x_t, x_{t+1}, \ldots, x_T$$

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

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



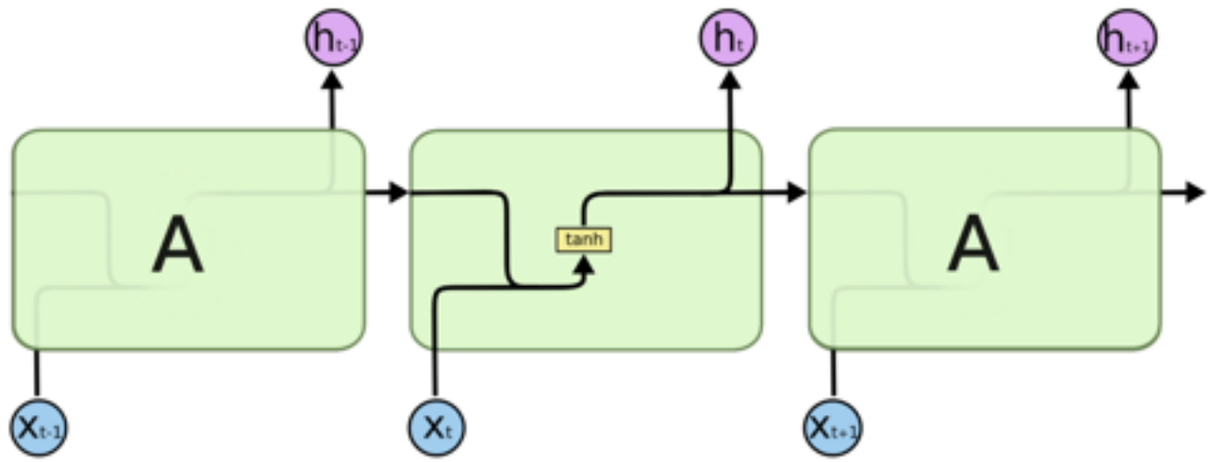
Réseaux de neurones récurrents

Ces réseaux ont longtemps été difficiles à entrainer : plus le nombre de pas de temps augmente, plus il est difficile d'entrainer les neurones éloignées du pas de temps courant. C'est le problème du *vanishing gradient* (Bengio1994)

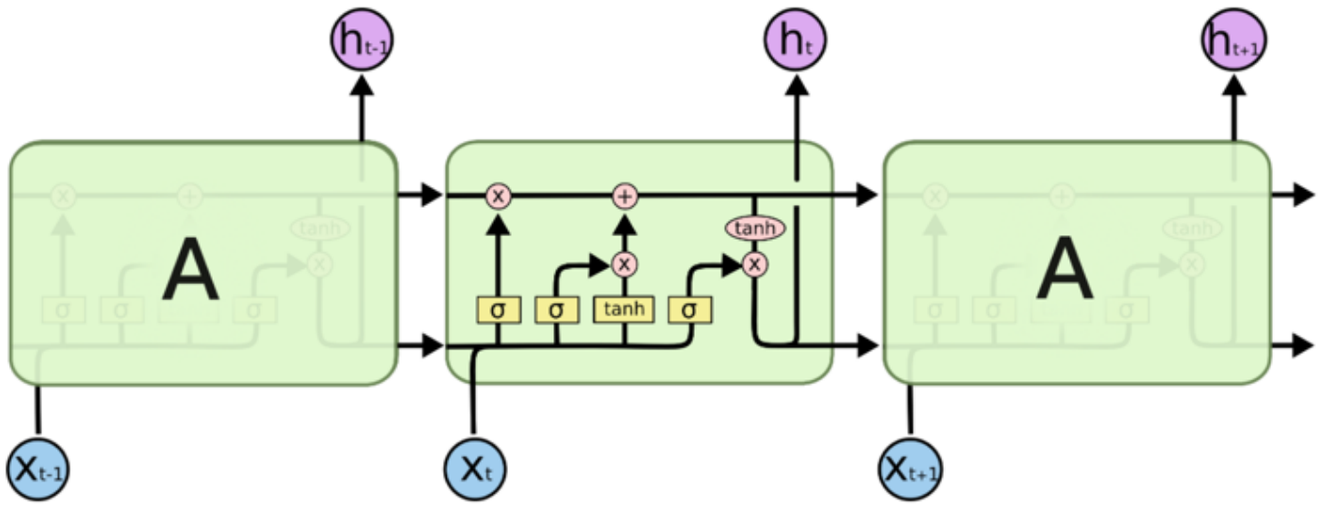
Un autre problème de ces réseaux est l'explosion numérique du gradient : dépassement des capacité numériques, saturation des neurones.

Réseaux de neurones récurrents

Pour résoudre le problème du *vanishing gradient*, un mécanisme de mémoires et de portes a été introduit dans la cellule récurrente : le modèle LSTM (*long short-term memory*) [Hochreiter1997]

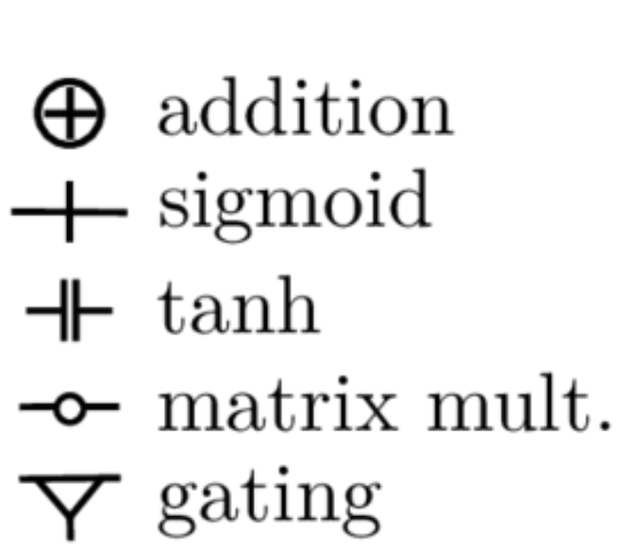


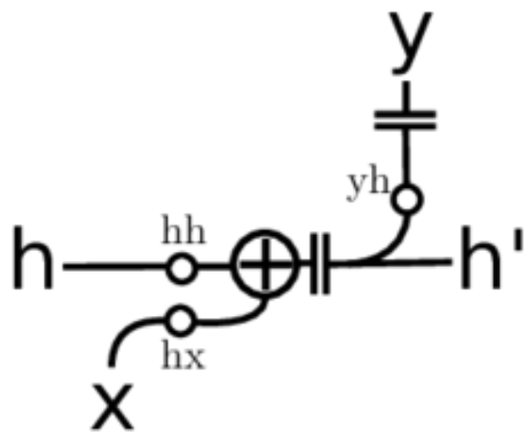
Réseau récurrent



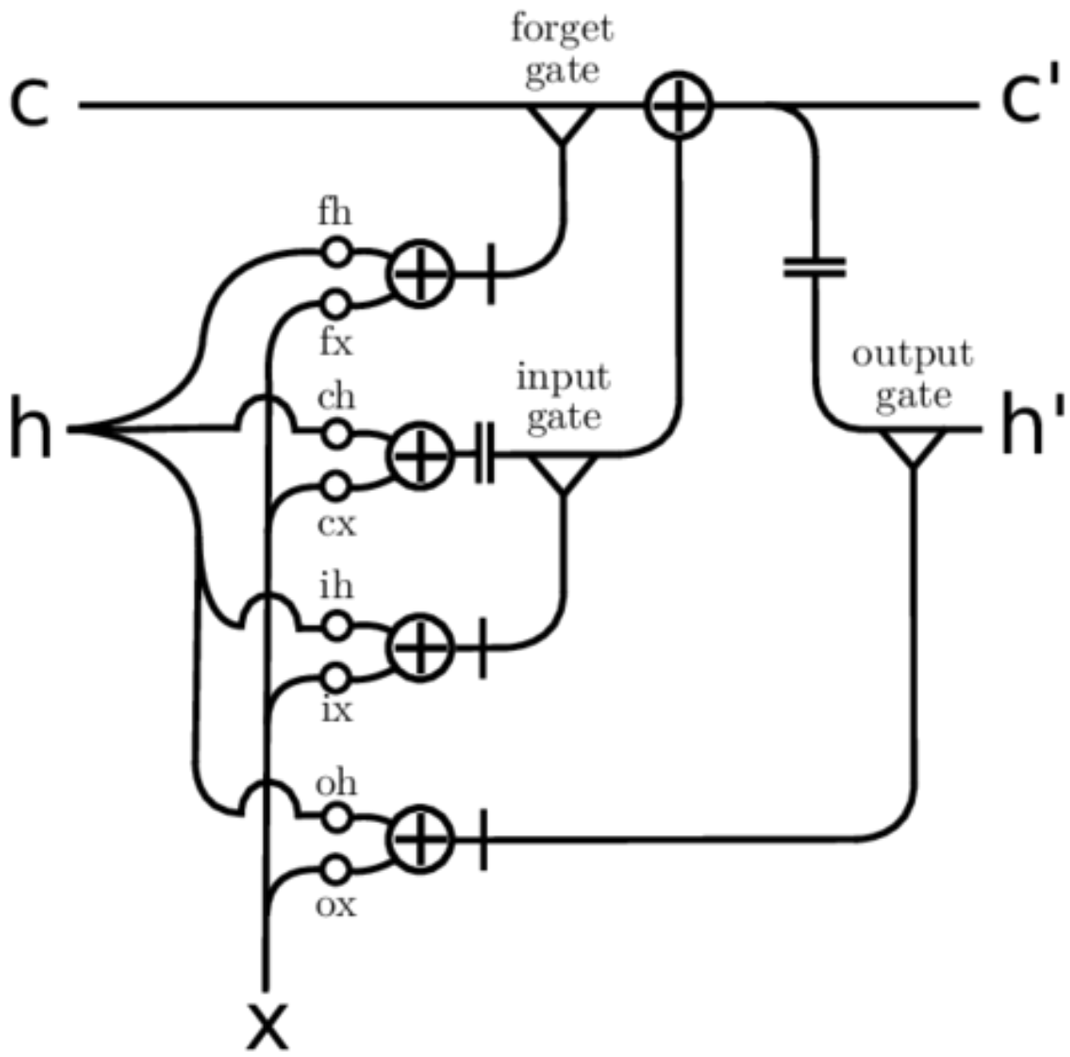
LSTM

Réseaux de neurones récurrents : LSTM





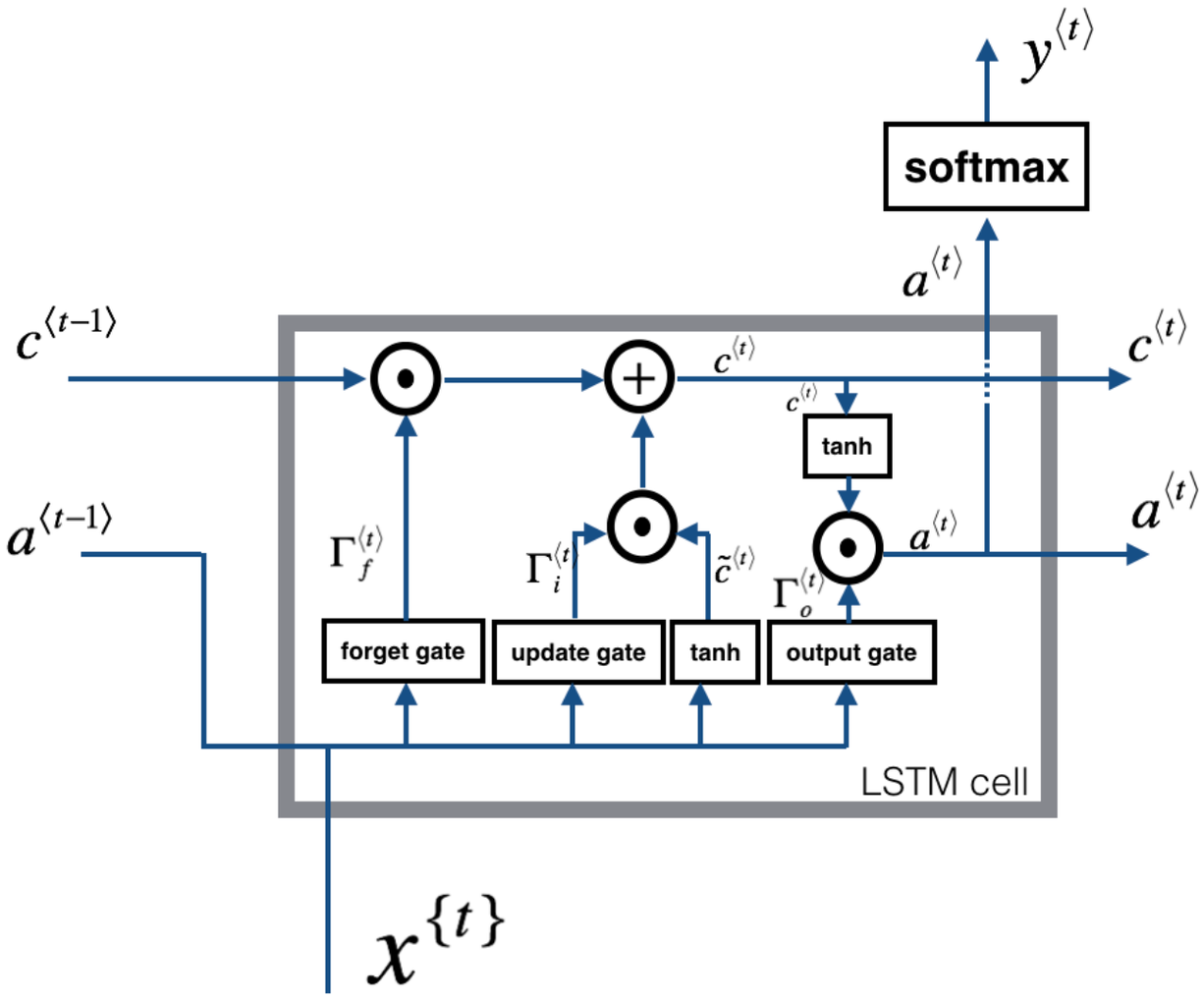
RNN



LSTM

Réseaux de neurones récurrents : LSTM

Détails des calculs :



$$\Gamma_f^{\langle t \rangle} = \sigma(W_f[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_f)$$

$$\Gamma_u^{\langle t \rangle} = \sigma(W_u[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_u)$$

$$\tilde{c}^{\langle t \rangle} = \tanh(W_C[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_C)$$

$$c^{\langle t \rangle} = \Gamma_f^{\langle t \rangle} \circ c^{\langle t-1 \rangle} + \Gamma_u^{\langle t \rangle} \circ \tilde{c}^{\langle t \rangle}$$

$$\Gamma_o^{\langle t \rangle} = \sigma(W_o[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_o)$$

$$a^{\langle t \rangle} = \Gamma_o^{\langle t \rangle} \circ \tanh(c^{\langle t \rangle})$$

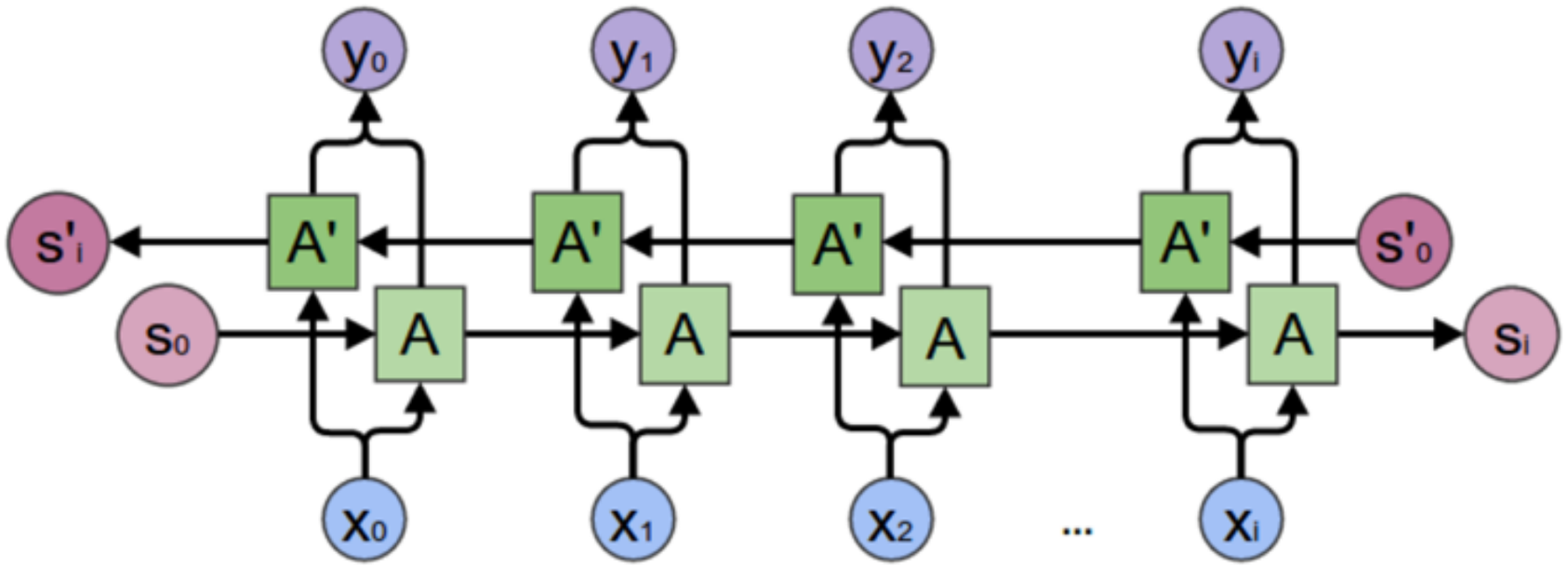
Réseaux de neurones récurrents : LSTM

Ce mécanisme, dont les paramètres sont appris, permet au réseau de choisir la longueur des dépendances utilisées pour faire les prédictions

Grace à ce mécanisme et à une méthode d'entrainement (CTC) [Graves2006], ces modèles ont pu être utilisés avec succès sur des applications réelles.

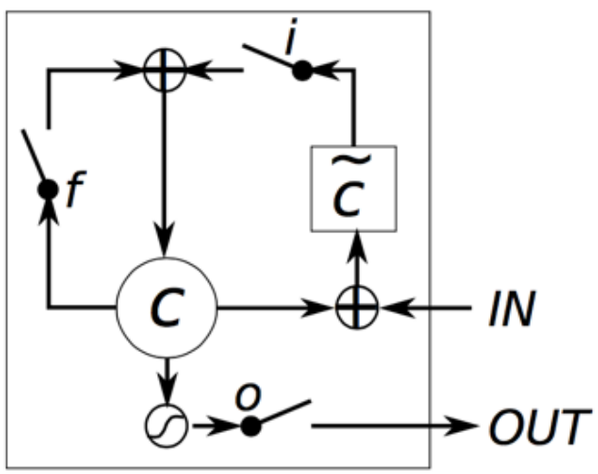
Réseaux de neurones récurrents : BiLSTM

Afin de modéliser les dépendances dans une séquence, les réseaux récurrent peuvent être utilisés dans les deux directions (voire 4 dans le cas d'une image) :

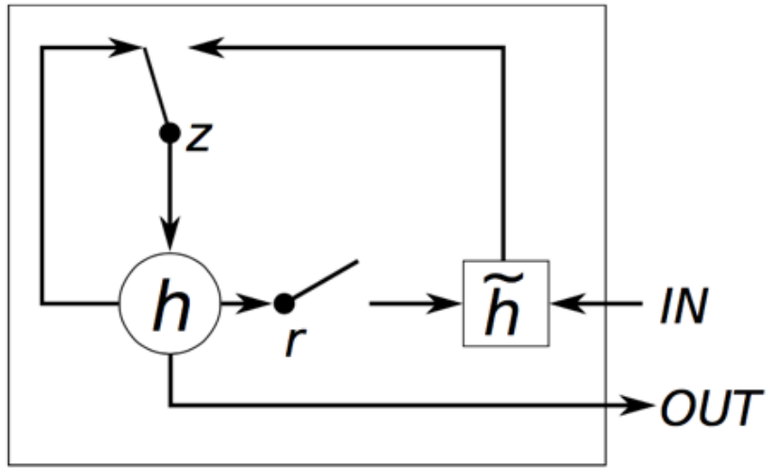


Réseaux de neurones récurrents : GRU

Une version simplifiée des LSTM a été proposée avec des performances similaires : GRU [Cho2014]



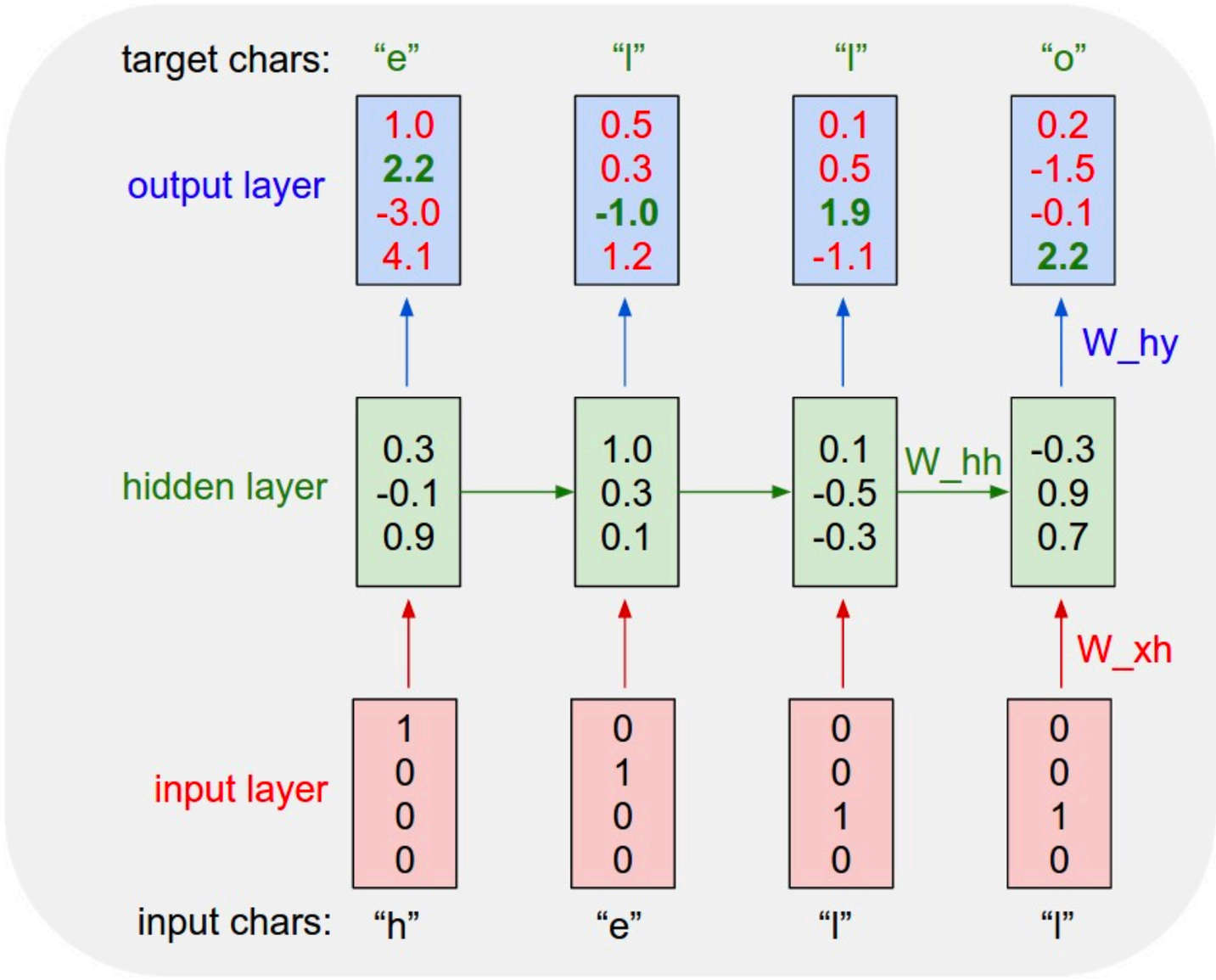
LSTM



GRU

Modèles de langue neuronaux

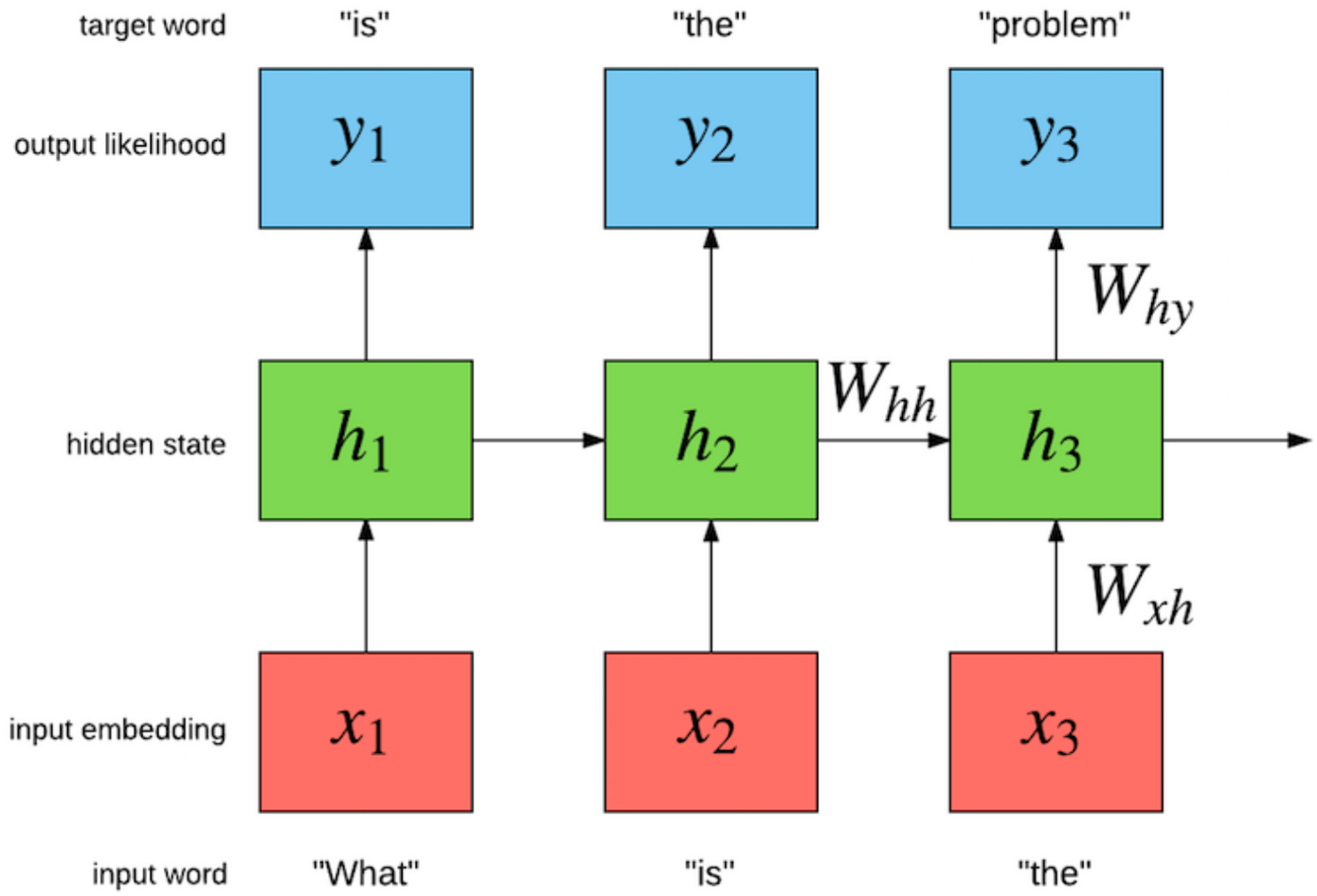
Modèle de caractères :



Karpathy, The Unreasonable Effectiveness of Recurrent Neural Networks, 2015

Modèles de langue neuronaux

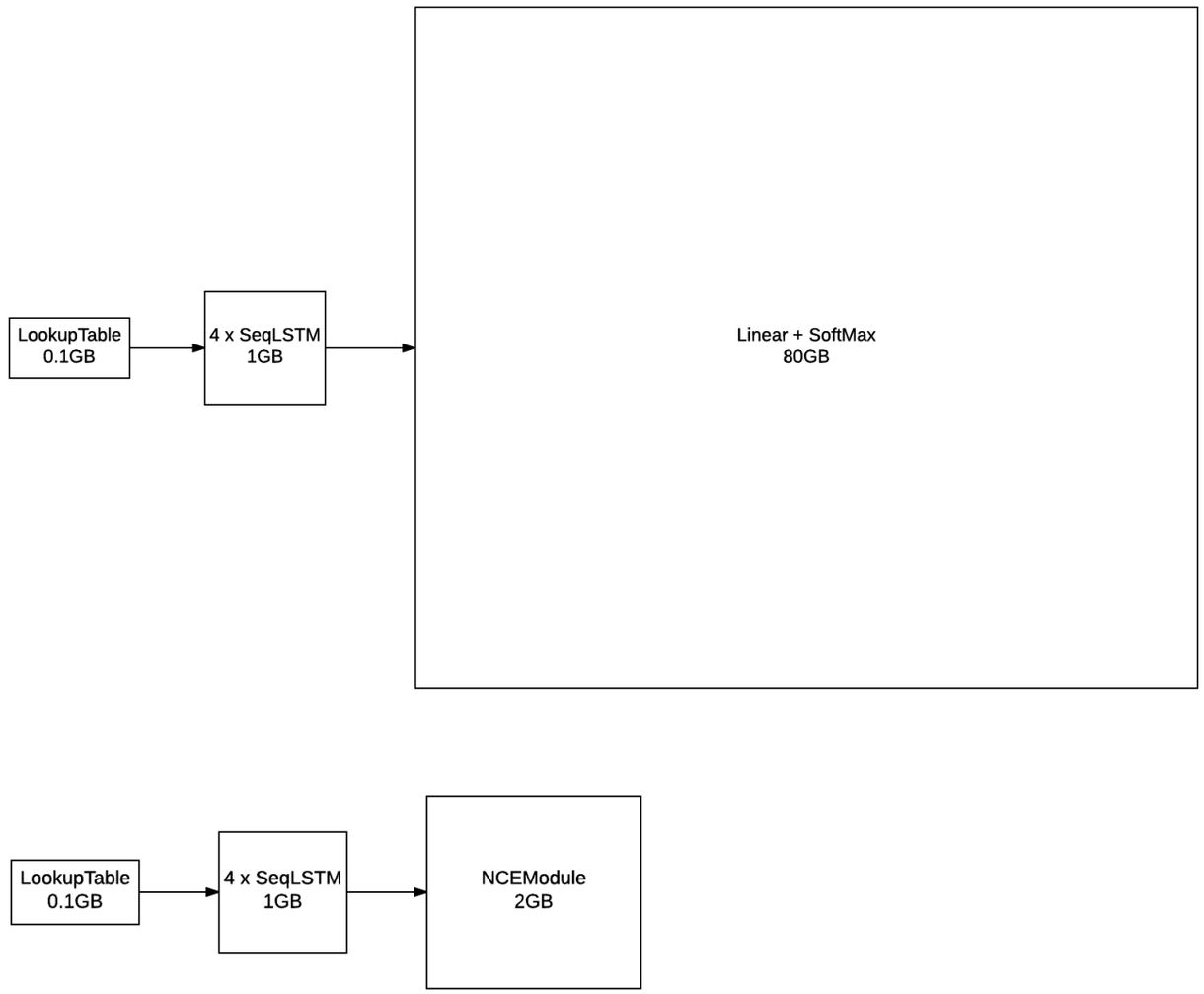
Modèle de mots:



Leonard, Language modeling a billion words, 2016.

Modèles de langue neuronaux

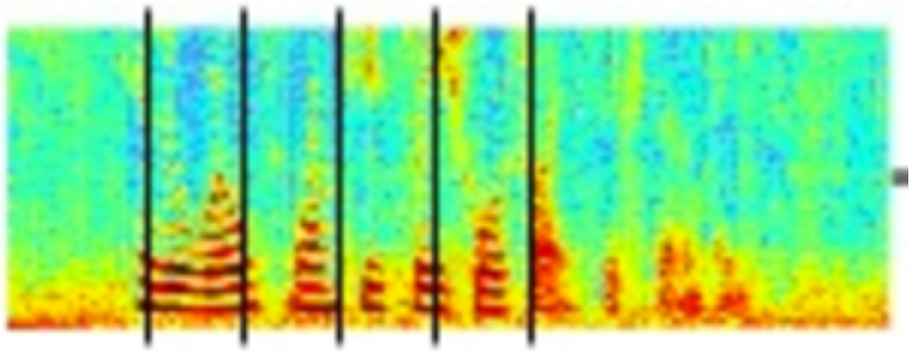
Modèle de mots:



Leonard, Language modeling a billion words, 2016.

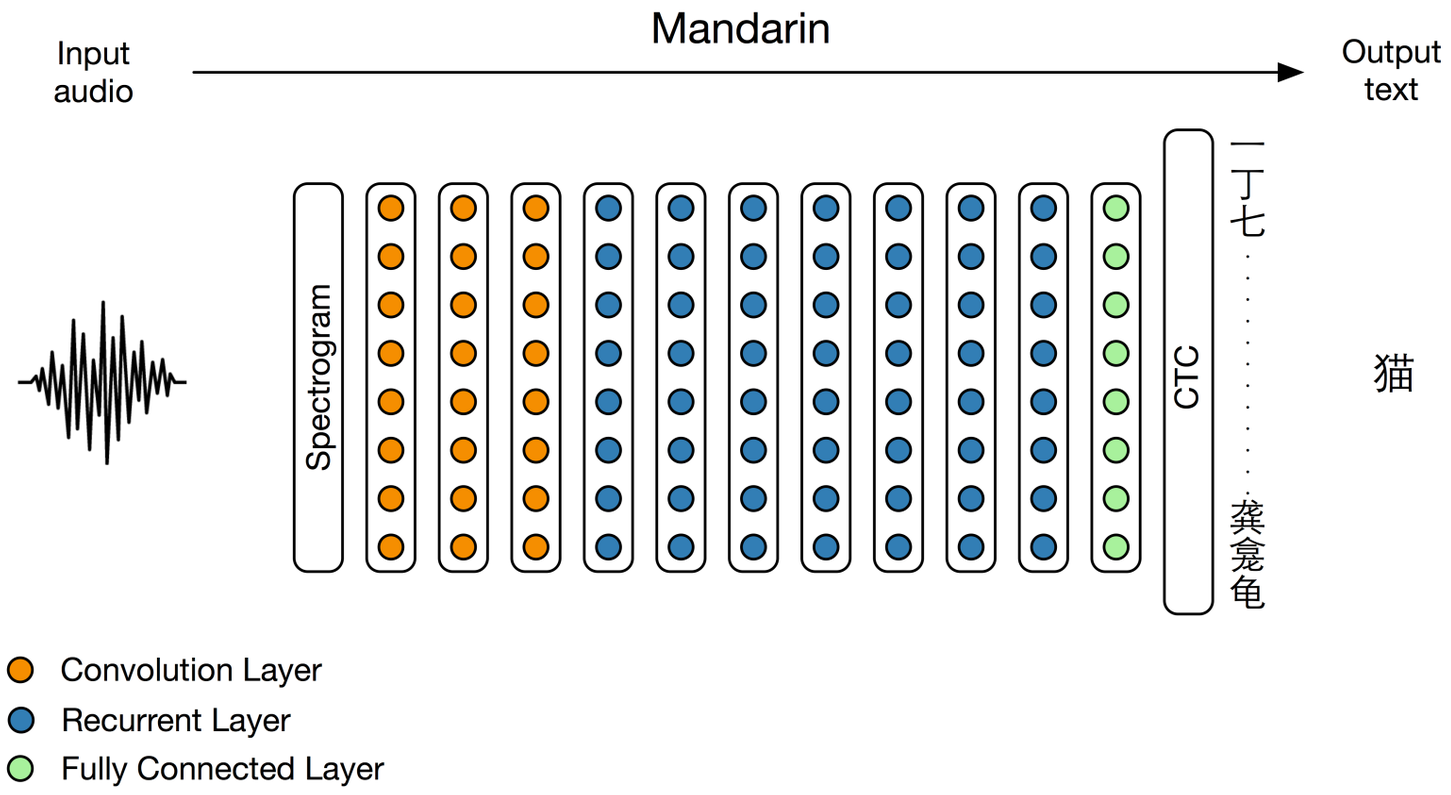
Modèles de langue : applications

Reconnaissance de la parole



Signal acoustique

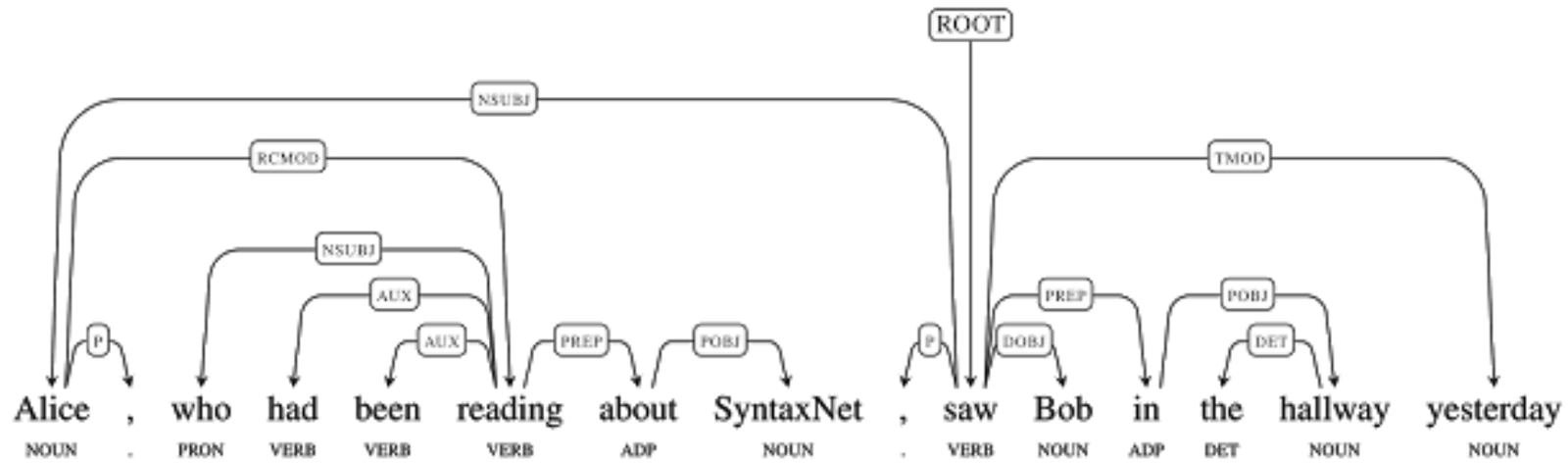
« How cold is it outside »

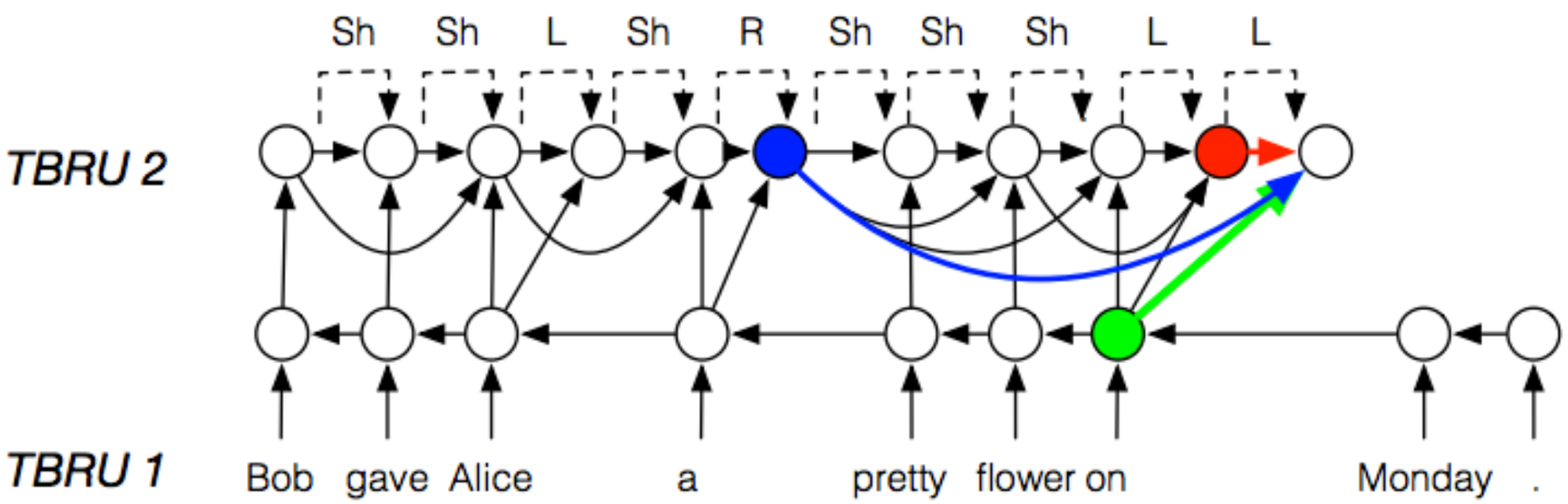


Modèles de langue : applications

Analyse syntaxique :

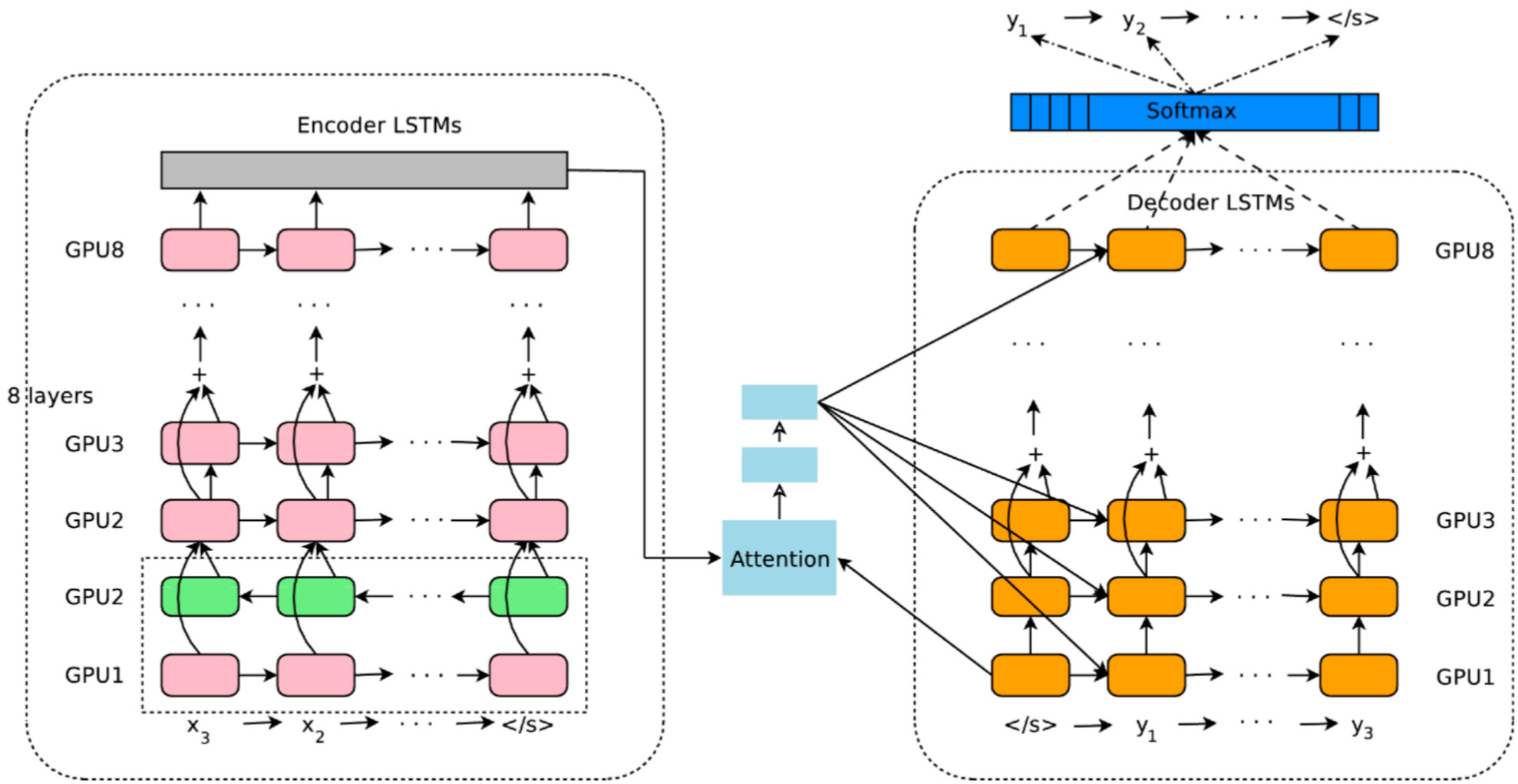
[SyntaxNet: Neural Models of Syntax.]





Modèles de langue : applications

Traduction automatique



Wu et al., Google's Neural Machine Translation System: Bridging the Gapbetween Human and Machine Translation, 2016

Recherche d'information : question answering

Stanford Question Answering Dataset (SQuAD)

Question: Which British general was killed at Khartoum in 1885?
Answer: Gordon

Context: In February 1885 Gordon returned to the Sudan to evacuate Egyptian forces. Khartoum came under siege the next month and rebels broke into the city, killing Gordon and the other defenders. The British public reacted to his death by acclaiming 'Gordon of Khartoum', a saint. However, historians have suggested that Gordon...

SQuAD2.0 combines the 100,000 questions in SQuAD1.1 with over 50,000 new, unanswerable

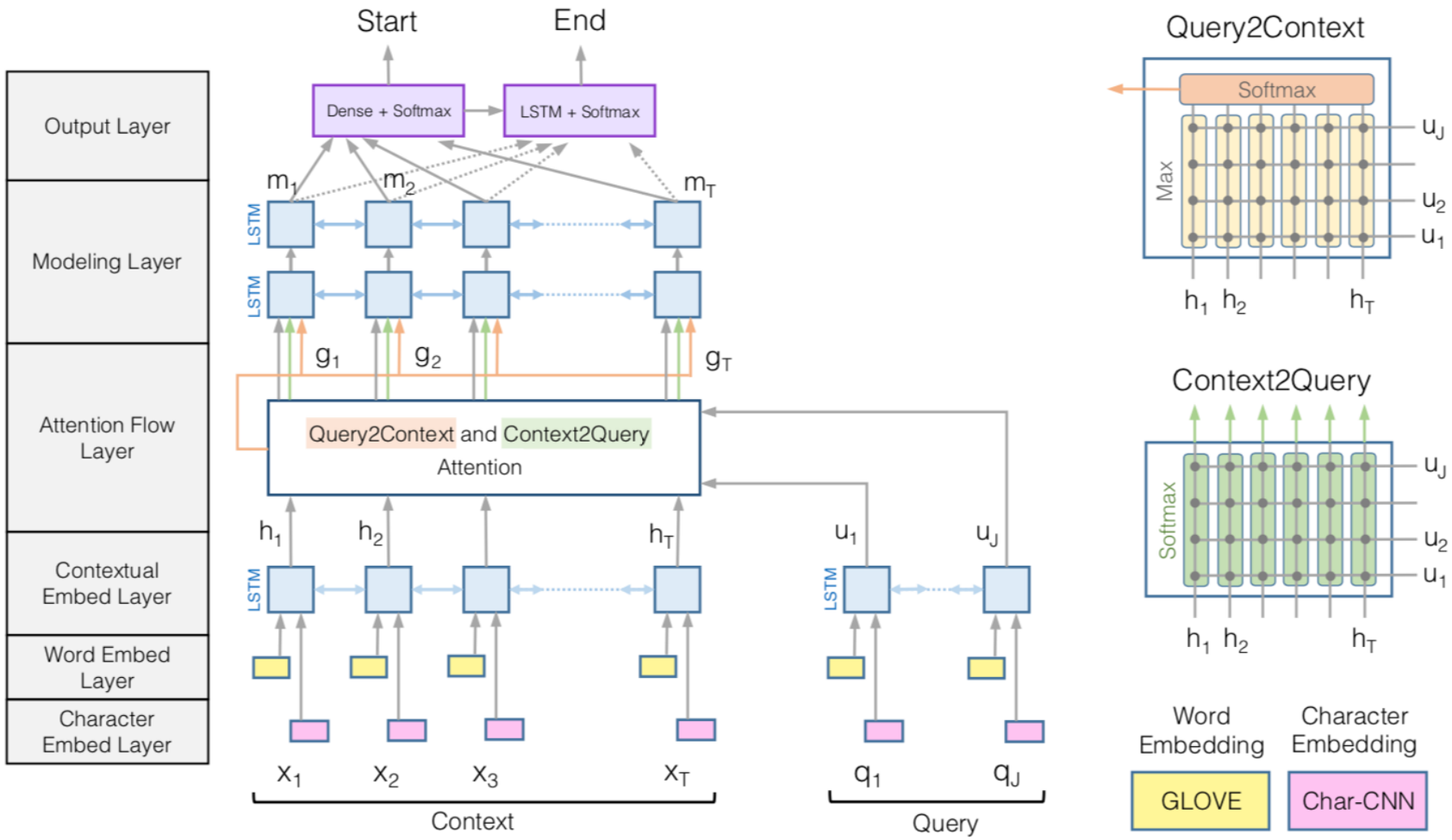
Leaderboard

SQuAD2.0 tests the ability of a system to not only answer reading comprehension questions, but also abstain when presented with a question that cannot be answered based on the provided paragraph. How will your system compare to humans on this task?

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Sep 13, 2018	nlnet (single model) *Microsoft Research Asia*	**74.238**	**77.022**
2 Oct 12, 2018	YARCS (ensemble) *IBM Research AI*	72.670	75.493
3 Oct 13, 2018	RNANetSimple (ensemble) *Anonymous*	72.602	75.089

Modèles de langue : applications

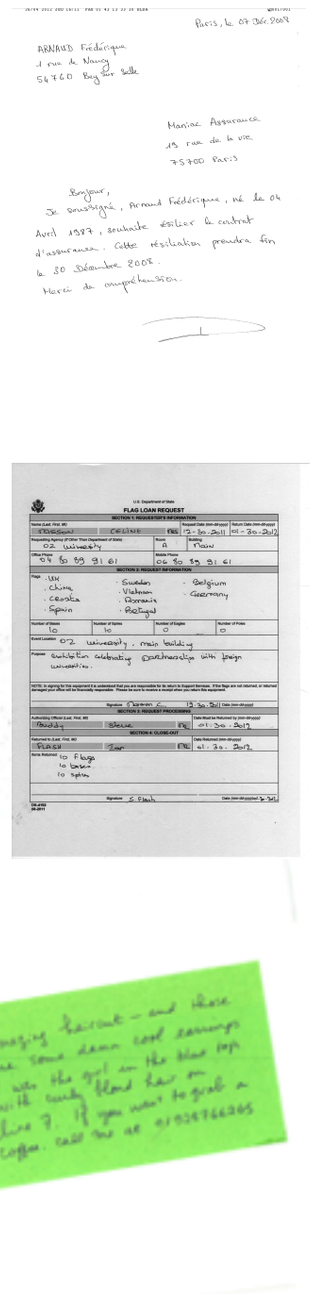
Recherche d'information : question answering

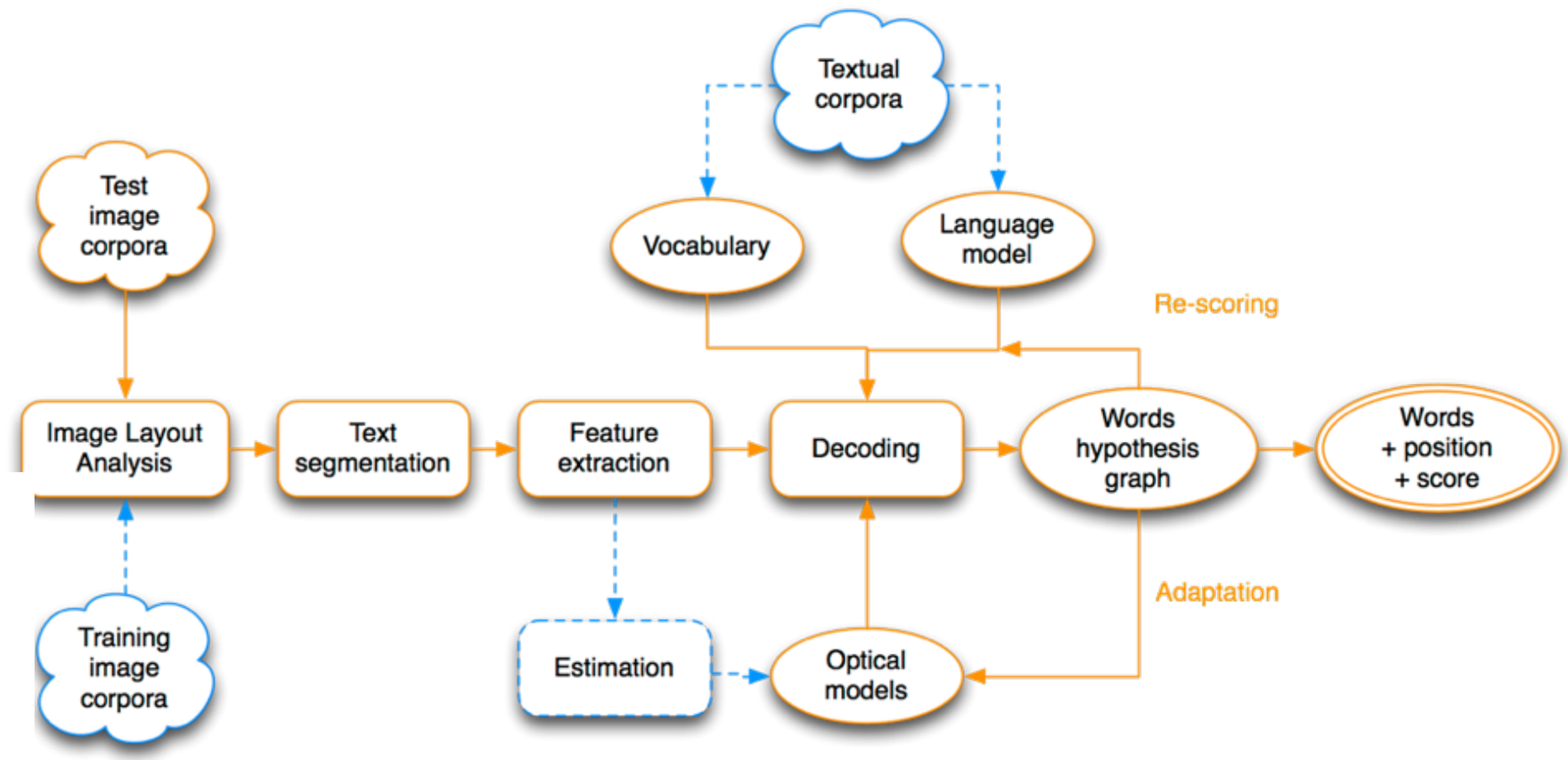


Wu et al., **Bidirectional Attention Flow for Machine Comprehension, ICLR 2017**

Modèles de langue : applications

Reconnaissance d'écriture





Modèles de langue : applications

Reconnaissance d'écriture