

RECHERCHE DE RÉPONSE DANS UNE BASE DE DE DOCUMENTS À L'AIDE DE RÉSEAUX RÉCURRENTS

CES Data Science 2019

Soutenance de mon projet personnel

Nabil Boukraa

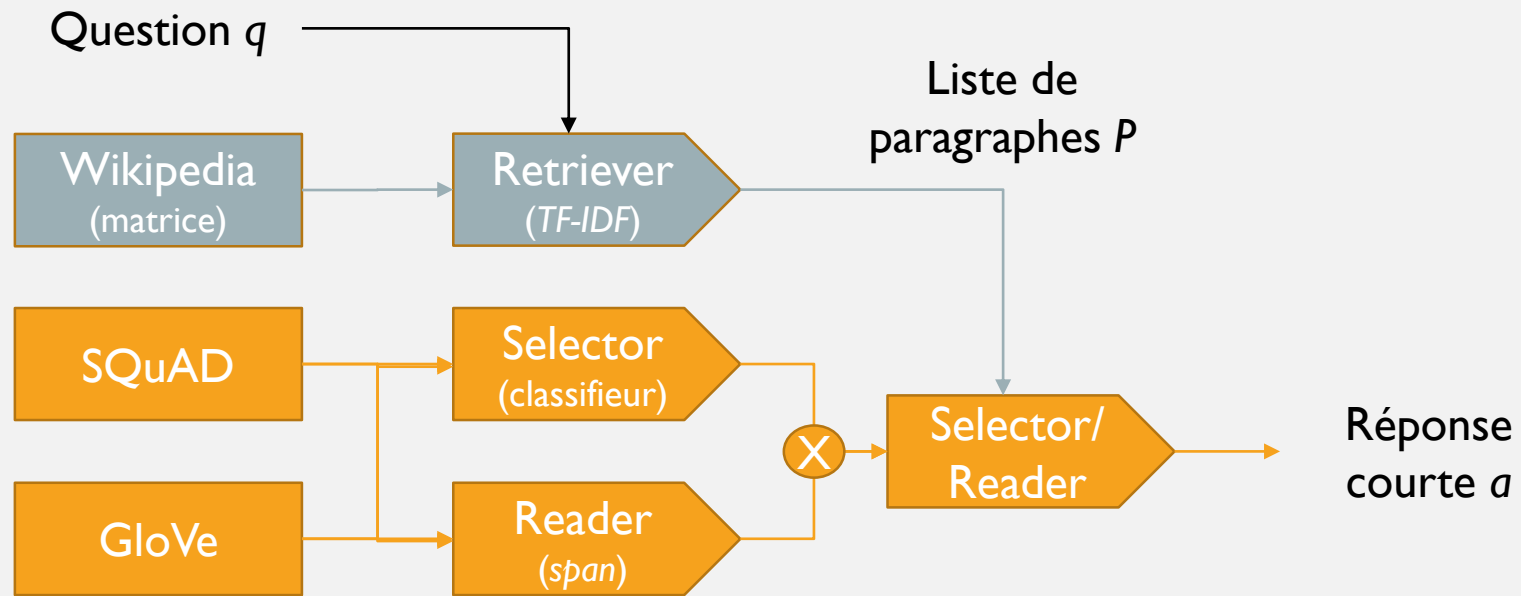
OBJECTIFS

- Introduction au problème de la réponse automatique :
 - **DrQA:** “*Reading Wikipedia to Answer Open-Domain Questions*”, Chen et al. (2017)
 - **OpenQA:** “*Denoising Distantly Supervised Open-Domain Question Answering*”, Lin et al. (2018)
- Implémentations disponibles sous python / pytorch

PRÉSENTATION DES DONNÉES DISPONIBLES

- Une collection d'articles Wikipedia:
 - Unique source de connaissance pour répondre aux questions
 - Pas de base de connaissance
- Un jeu de questions/réponses/paragraphes:
 - “SQuAD: 100,000+ Questions for Machine Comprehension of Text”, Rajpurkar et al. (2016)
- Une représentation vectorielle des mots:
 - “GloVe : Global Vectors for Word Representation”, Pennington et al. (2014)
 - La représentation vectorielle des mots est telle que leur produit scalaire est égal au logarithme de la probabilité que ces mots apparaissent ensemble.

ARCHITECTURE



$$\Pr(a | q, P) = \sum_{p \in P} \Pr(a | q, p) \cdot \Pr(p | q, P)$$

$$L(\theta) = - \sum_{(q, a, P) \in \Gamma} \ln(\Pr(a | q, P)) - \alpha R(P)$$

PHASE I - RETRIEVER

- La pertinence d'un article j est évaluée à l'aide de l'équation:

$$w_{i,j} = tf_{i,j} \times \ln \left(\frac{N}{df_i} \right)$$

$tf_{i,j}$ = nombre d'occurrences du terme i dans l'article j

df_i = nombre de documents contenant le terme i

N = nombre total de documents dans le corpus

PHASE II – SELECTOR/READER

- Décomposition du problème:

$$\Pr(a \mid q, P) = \sum_{p \in P} \Pr(a \mid q, p) \times \Pr(p \mid q, P)$$

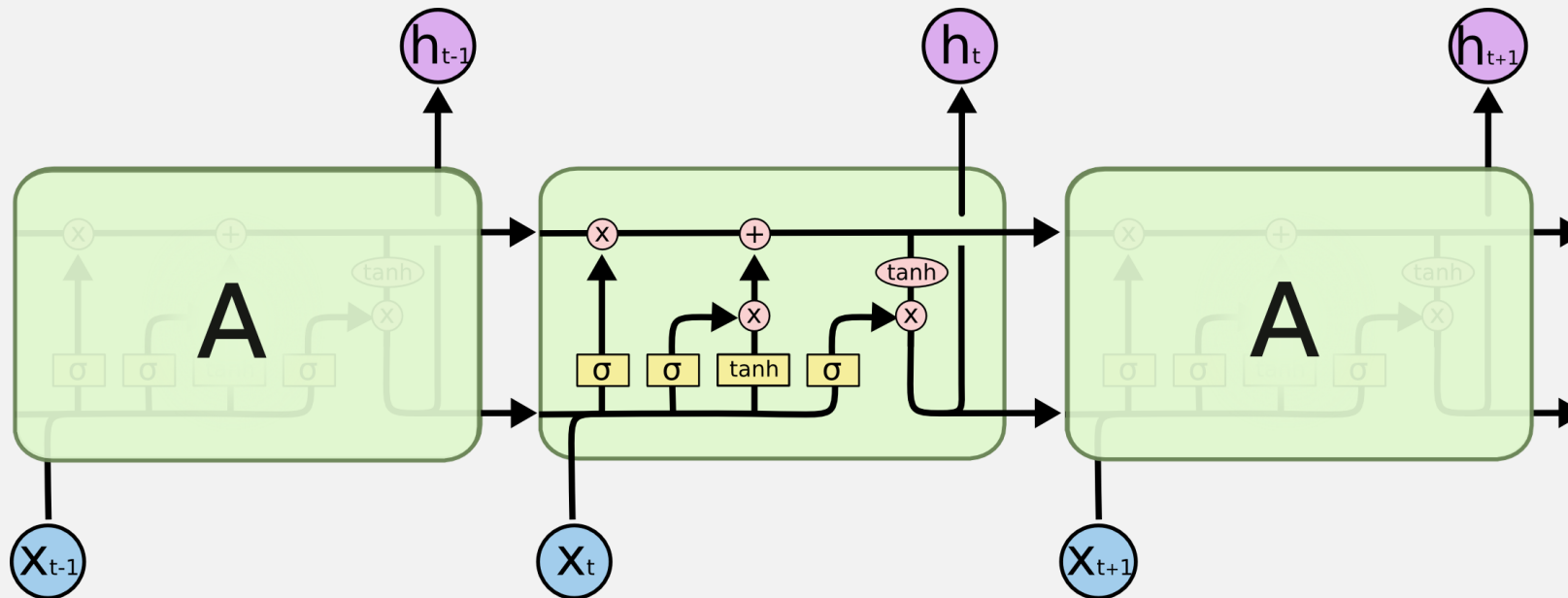
- *Document Selector:*

$$\Pr(p_i \mid q, P) = \text{softmax}_{1 \leq i \leq M} \left(\max_{1 \leq j \leq N} (p_{i,j} \mathbf{W}^{SELECT} q) \right)$$

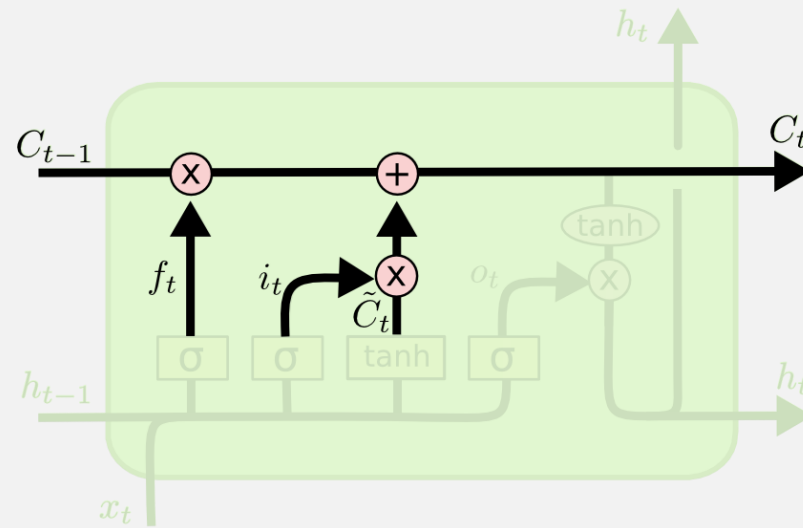
- *Document Reader:*

$$\begin{cases} \Pr(a \mid q, p) = \max_{1 \leq a_s, a_e \leq N} (P_s(a_s \mid q, p) \times P_e(a_e \mid q, p)) \\ P_s(j) = \text{softmax}_{1 \leq j \leq N} (p_j \mathbf{W}_s^{READ} q) \\ P_e(j) = \text{softmax}_{1 \leq j \leq N} (p_j \mathbf{W}_e^{READ} q) \end{cases}$$

RESEAU RÉCURRENT DE TYPE *LSTM*

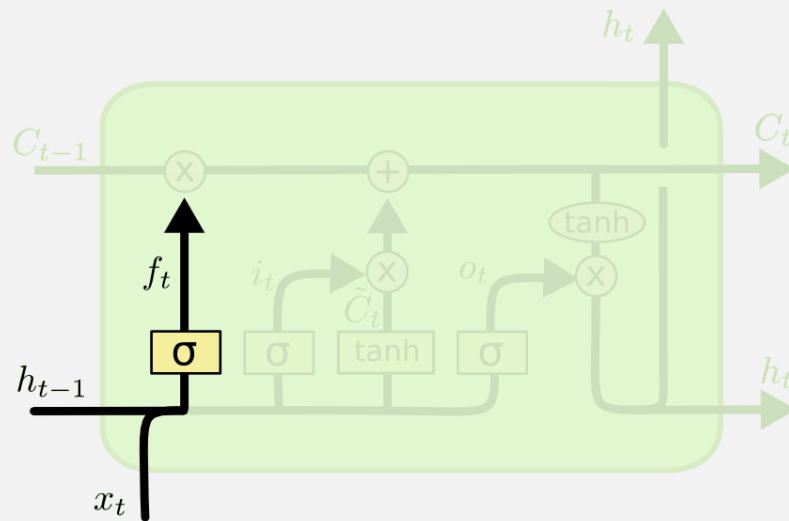


ETATS DE LA CELLULE



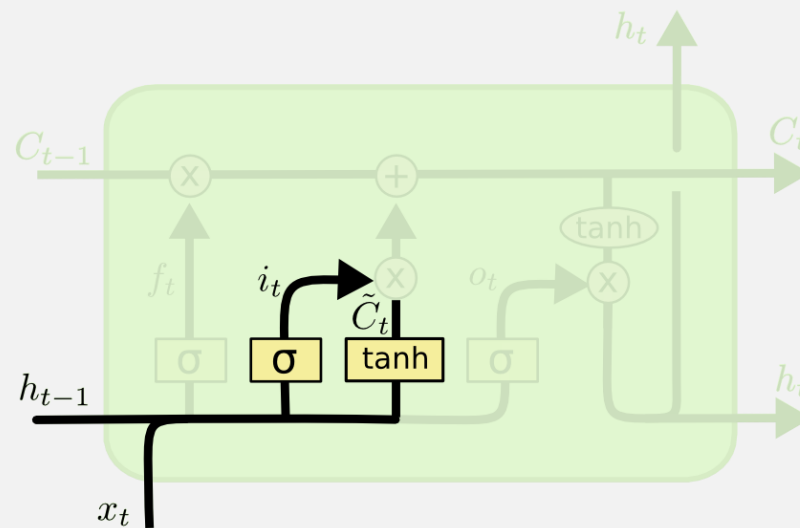
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

FORGET GATE



$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f)$$

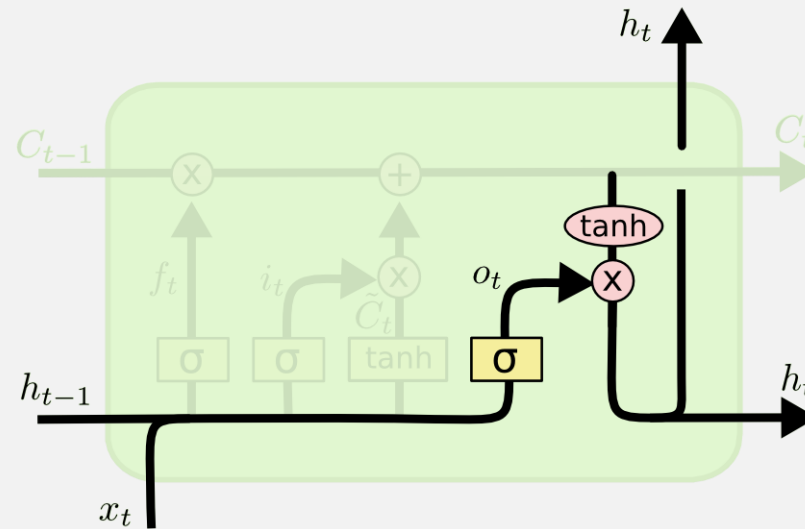
INPUT GATE



$$i_t = \sigma (W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

OUTPUT GATE



$$o_t = \sigma (W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh (C_t)$$

L'AUTO-ATTENTION

$$\begin{cases} \hat{q}_j = \sum_{1 \leq k \leq L} \alpha_{j,k} q_k \\ \alpha_j = \text{softmax}_{1 \leq k \leq N} \left(\mathbf{w}^{ATTN} q_k \right) \end{cases}$$

où \mathbf{w}^{ATTN} est un vecteur de pondérations à apprendre.

TEST DE CULTURE GÉNÉRALE



TEST DE CULTURE GÉNÉRALE

1. On which island did Napoleon Bonaparte end his life?
2. What is the longest river in mainland France?
3. Who made the June 18, 1940 appeal, broadcast from the BBC in London?
4. About which case did Emile Zola publish the famous *J'accuse* open letter in 1898?
5. Since January 1, 2016, how many regions does France have (including overseas)?
6. Which singer, who has become the hip-hop's first billionaire this year, is Beyoncé's husband?
7. Which nation won the 2019 rugby world cup?
8. Which novel, written by Saint-Exupéry, is the second most translated book in the world after the Bible?
9. Which Miss France won the Miss Universe title in January 2017 in the Philippines?
10. Who is the only French actor to have ever won an Oscar for best actor?

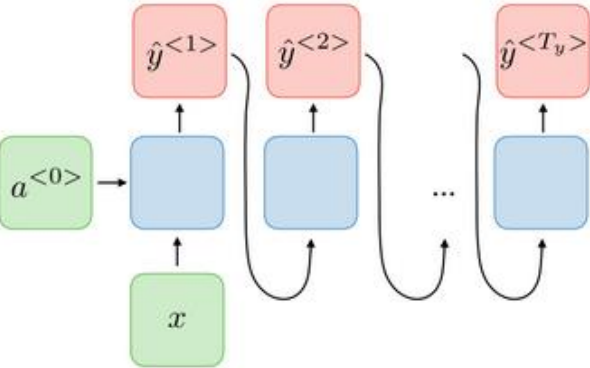
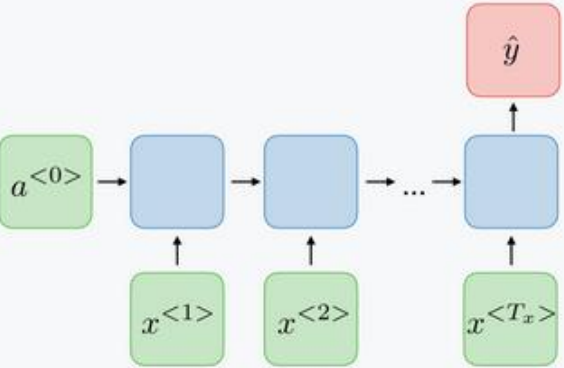
PROCHAINES ÉTAPES

- Enrichissement des données avec de nouvelles variables explicatives.
- Exploitation d'autres sources de connaissance.
- Utilisation d'encodages (*word embeddings*) construits à partir de la base Wikipedia: <https://github.com/ido/wiki2vec/>
- Utilisation de transformeurs à la place des réseaux de type LSTM, en me basant dans un premier temps sur la librairie *torch.nn.Transformer* de PyTorch: https://pytorch.org/tutorials/beginner/transformer_tutorial.html

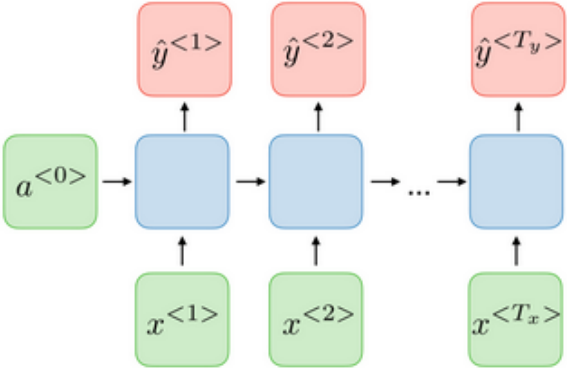
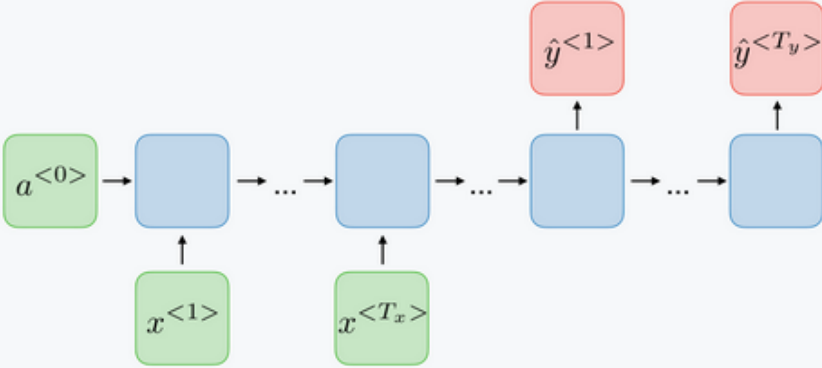
QUESTIONS?

APPENDICES

LES DIFFÉRENTS TYPES DE RESEAUX RÉCURRENTS (1/2)

<p>One-to-many $T_x = 1, T_y > 1$</p>		<p>Music generation</p>
<p>Many-to-one $T_x > 1, T_y = 1$</p>		<p>Sentiment classification</p>

LES DIFFÉRENTS TYPES DE RESEAUX RÉCURRENTS (2/2)

Many-to-many $T_x = T_y$		Name entity recognition
Many-to-many $T_x \neq T_y$		Machine translation