SEGMENTATION PAROLE/ NON-PAROLE

20/11/2014

Méthodes classiques et neuronales (Grégory GELLY)

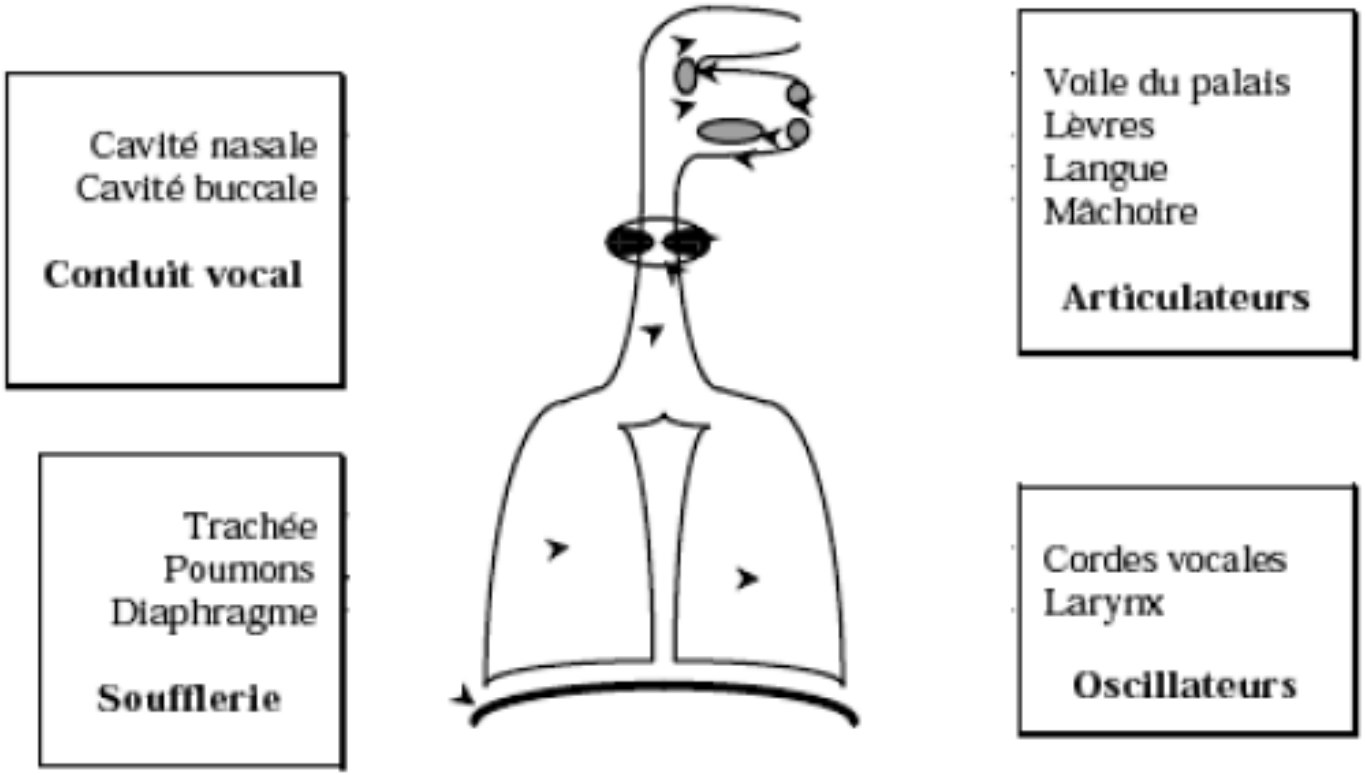
Ordre du jour

- Segmentation de la parole
  - Méthodes classiques
  - Réseaux de neurones
- Apprentissage/Optimisation
- Expérimentations
  - Segmentation
    - Corpus & métrique
    - Résultats
  - Transcription
    - Corpus & métrique
    - Résultats
- Bonnes pratiques
- Conclusions & Perspectives

Segmentation de la parole

Généralités

Organes impliqués dans la production de la parole



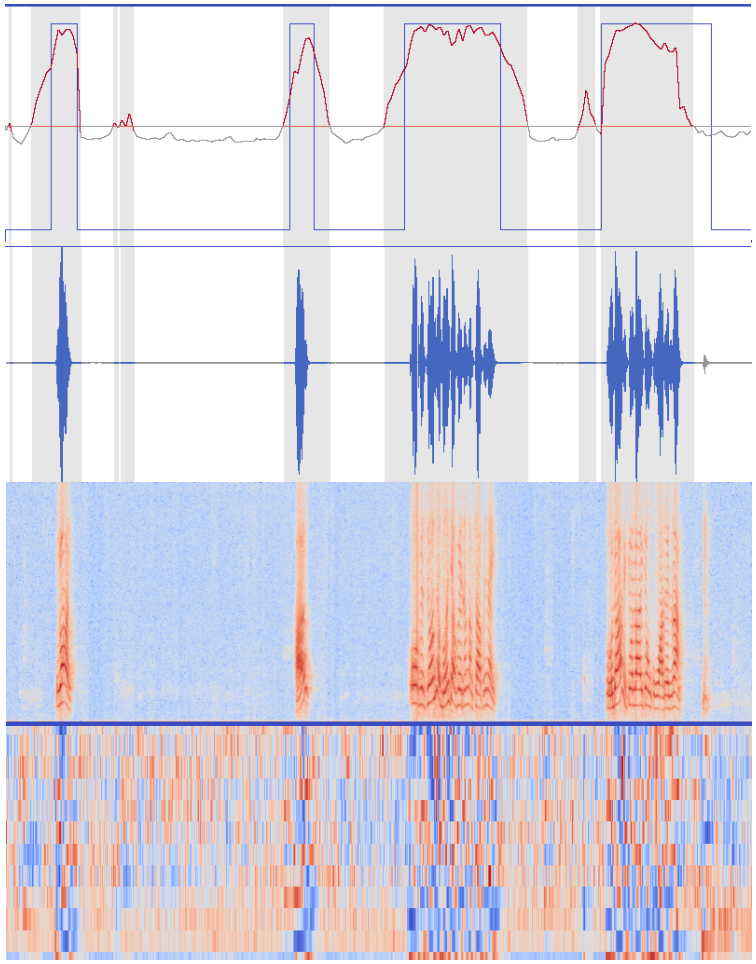
Représentations d'un signal audio

Segmentation

Forme d'onde

Spectre

MFCC

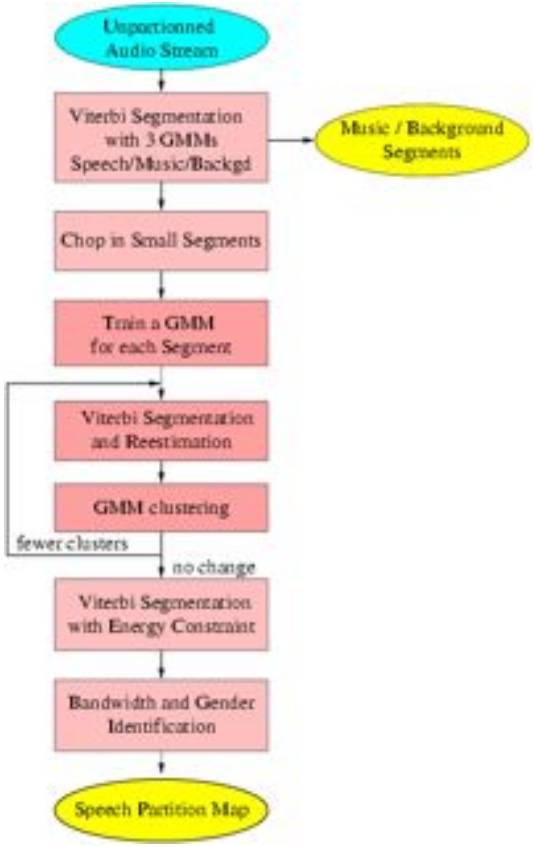


Segmentation de la parole

Méthodes classiques

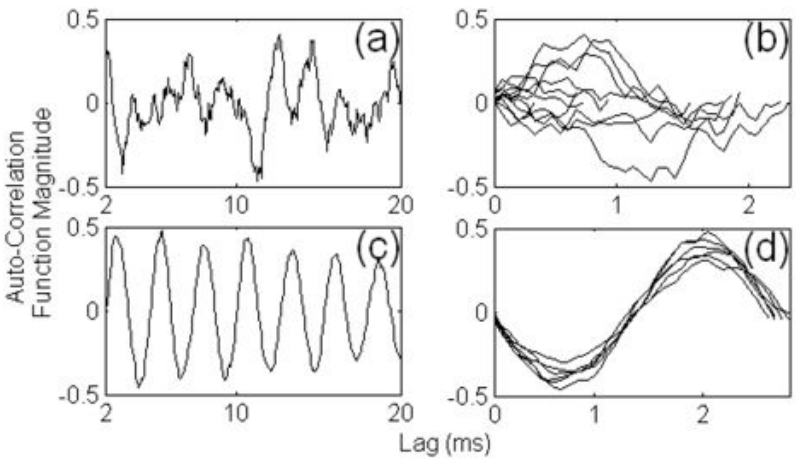
Méthodes classiques

- Outil de référence du LIMSI : GMM
  - Principe :
    - 1 : rejet de la non-parole (musique, tonalités, bruits…)
    - 2 : clustering de ce qui reste par maximum de vraisemblance pour construire des segments de parole cohérents
    - 3 : rejet des segments incertains
  - Bibliographie :
    - « Partitioning and transcription of broadcast news data », 1998, J.L. Gauvain, L. Lamel, G. Adda



Méthodes classiques

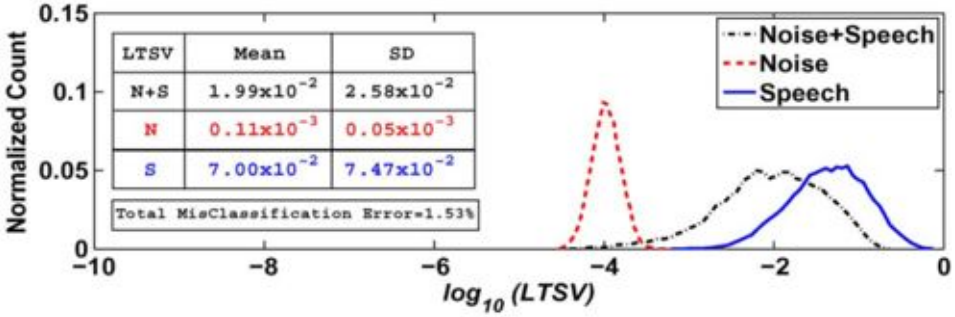
- Auto-corrélation du signal temporel
  - Principe :
    - capture du signal périodique des cordes vocales dans les fréquences basses de la parole



  - Bibliographie :
    - WALE (Windowed Autocorrelation Lag Energy) : « *Voicing Features for Robust Speech Detection* », 2005, T. Kristjansson
    - MaxPeak, CrossCor : « *Noise Robust Voice Activity Detection Using Features Extracted From the Time-Domain Autocorrelation Function* », 2011, H. Ghaemmaghami

Méthodes classiques

- Analyse de l'enveloppe spectrale
  - Principe :
    - exploite la variabilité du spectre de parole sur différentes bandes fréquences



  - Bibliographie :
    - LTSV : « *Robust Voice Activity Detection Using Long-Term Signal Variability* », P. K. Ghosh
    - LTSD : « *Efficient Voice Activity Detection Algorithms Using Long-Term Speech Information* », J. Ramirez

10

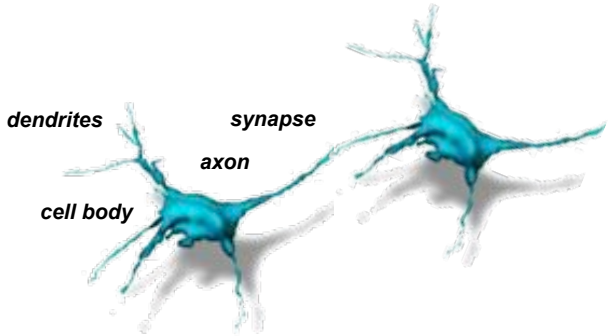
Segmentation de la parole

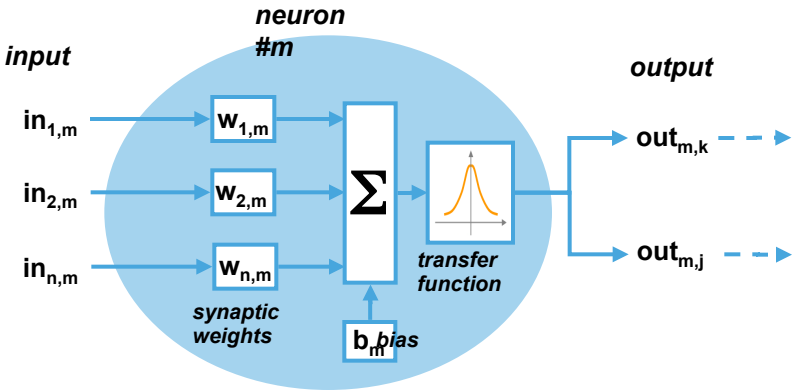
Réseaux de neurones

20/11/2014

Réseaux de neurones

- Les premières études datent des années 1960
  - Inspiré de la structure neuronale du cerveau :

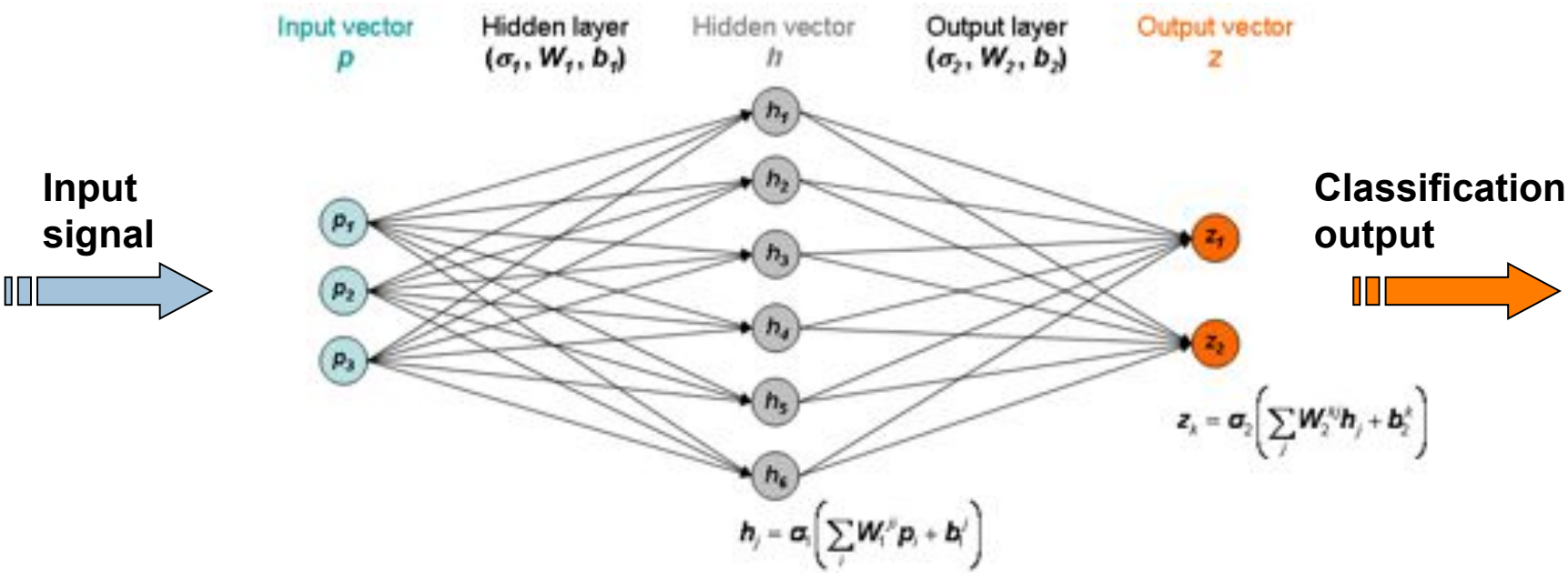




  - Cette technique a été utilisée avec succès dans de multiples et diverses applications : traitement de la parole, traitement d'image, aéronautique, finance, robotique…

Réseaux de neurones

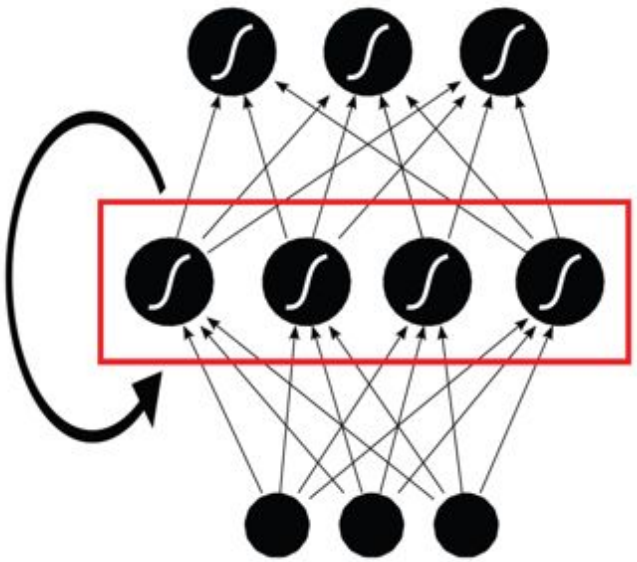
- Architecture classique :

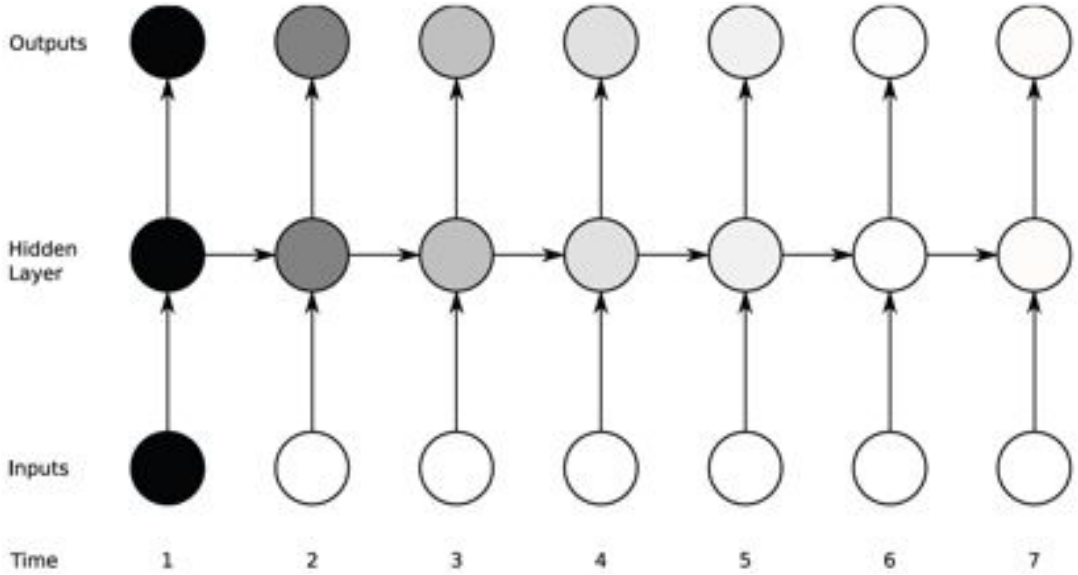


  - Ce type de réseaux n'exploite pas les événements passés ou futurs.

Réseaux de neurones récurrents (RNN)

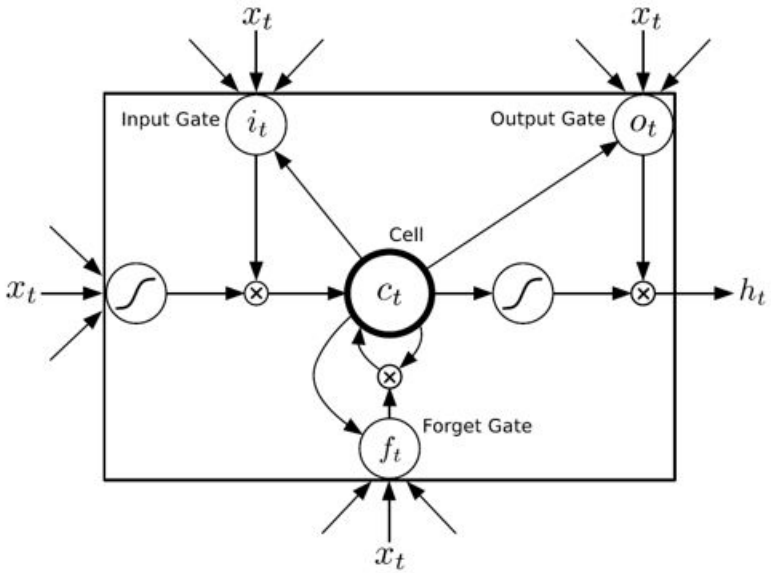
- Les RNN permettent de gérer les séquences mais décroissance exponentielle de la mémoire

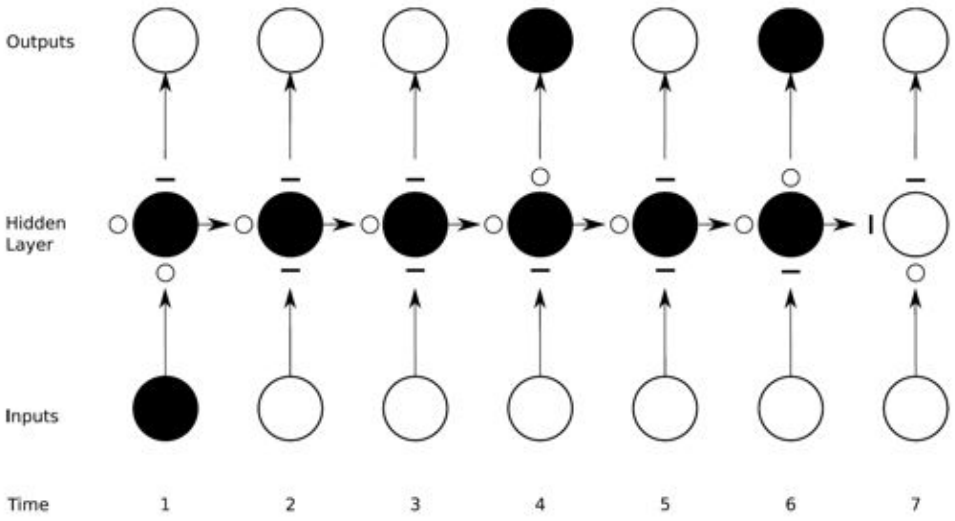




Réseaux de neurones LSTM

- Cellule à mémoire court-terme persistante :
  - Modification de la cellule de base (neurone) :

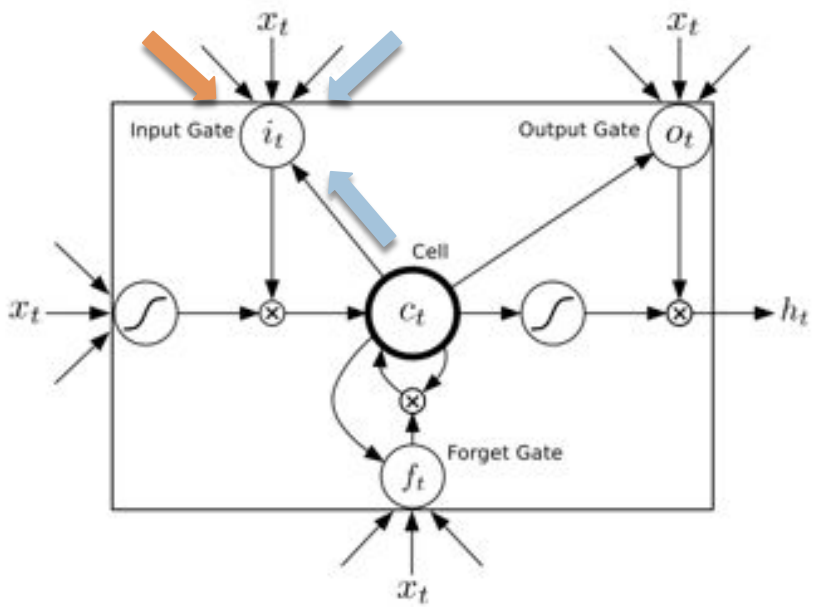




  - Cette technique a été utilisée avec succès dans plusieurs applications traitant des séquences : prédiction de structure de protéines, OCR.

Réseaux de neurones LSTM

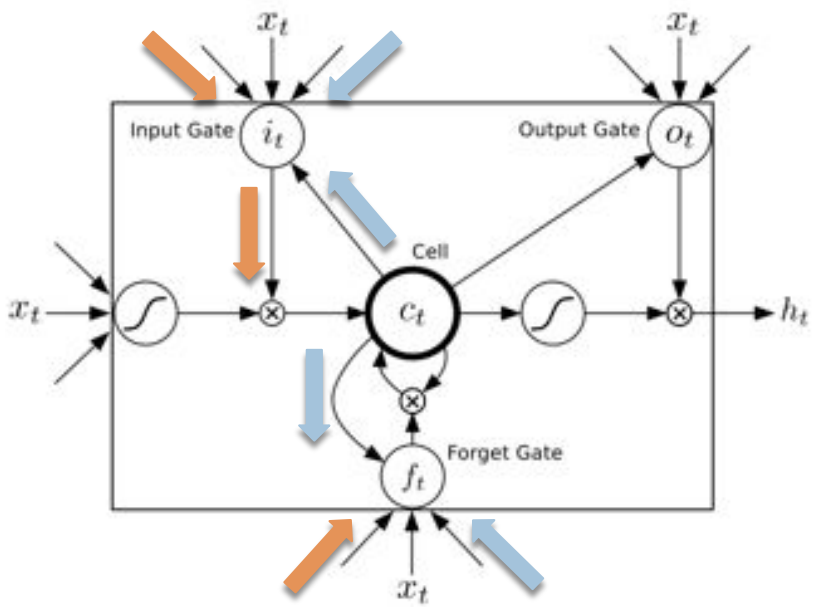
- Cellule à mémoire court-terme persistante :
  - Passe avant :



$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right)$$

Réseaux de neurones LSTM

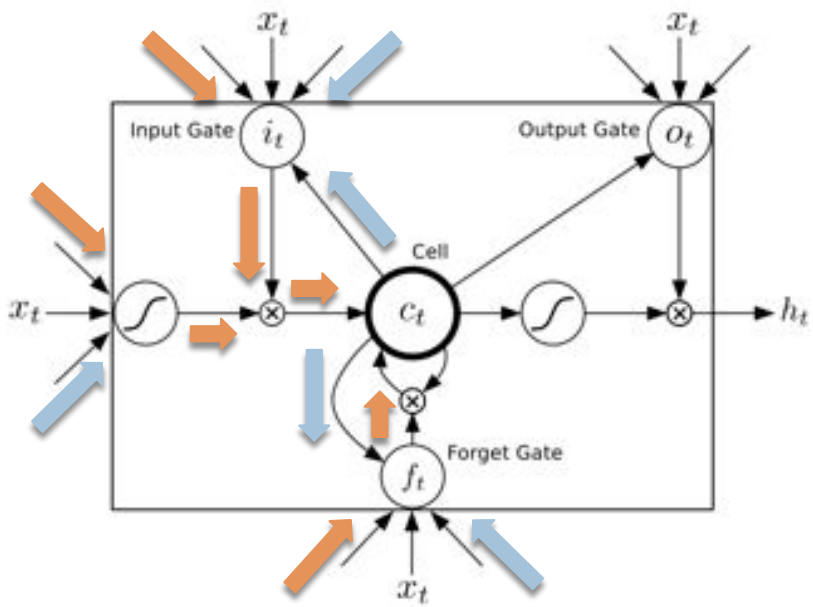
- Cellule à mémoire court-terme persistante :
  - Passe avant :



$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right)$$
$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right)$$

Réseaux de neurones LSTM

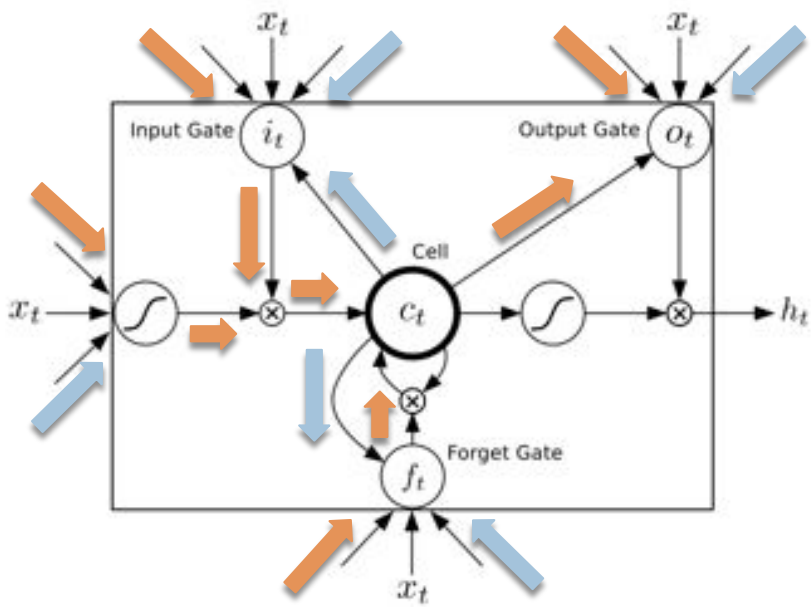
- Cellule à mémoire court-terme persistante :
  - Passe avant :



$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right)$$
$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right)$$
$$c_t = f_t c_{t-1} + i_t \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right)$$

Réseaux de neurones LSTM

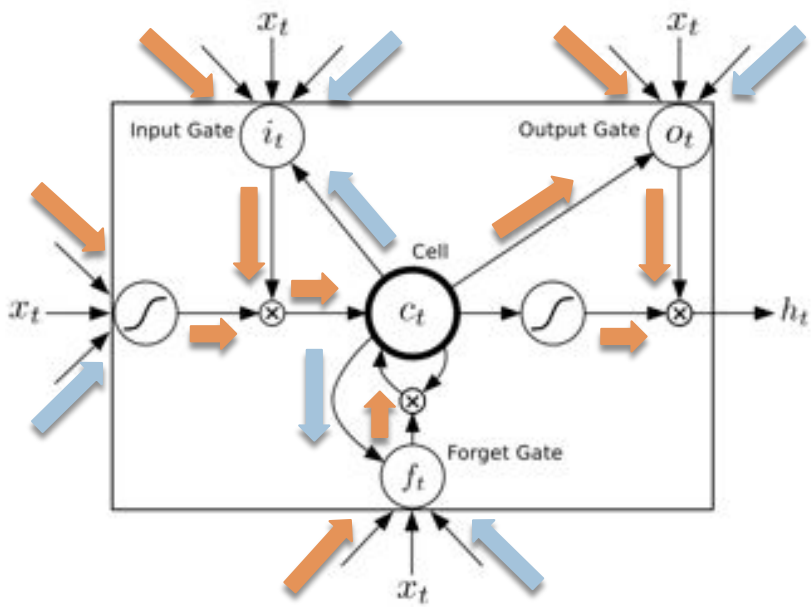
- Cellule à mémoire court-terme persistante :
  - Passe avant :



$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right)$$
$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right)$$
$$c_t = f_t c_{t-1} + i_t \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right)$$
$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o\right)$$

Réseaux de neurones LSTM

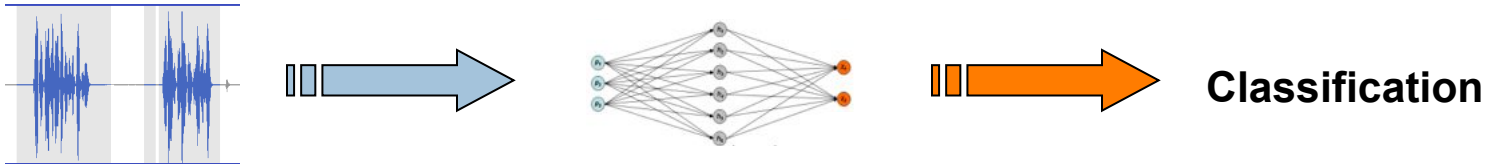
- Cellule à mémoire court-terme persistante :
  - Passe avant :



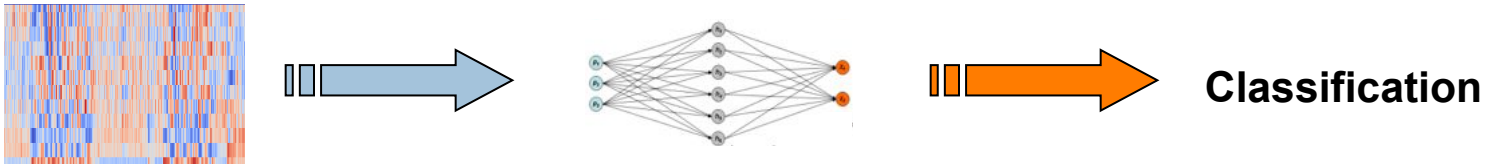
$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right)$$
$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right)$$
$$c_t = f_t c_{t-1} + i_t \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right)$$
$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o\right)$$
$$h_t = o_t \tanh(c_t)$$

Réseaux de neurones

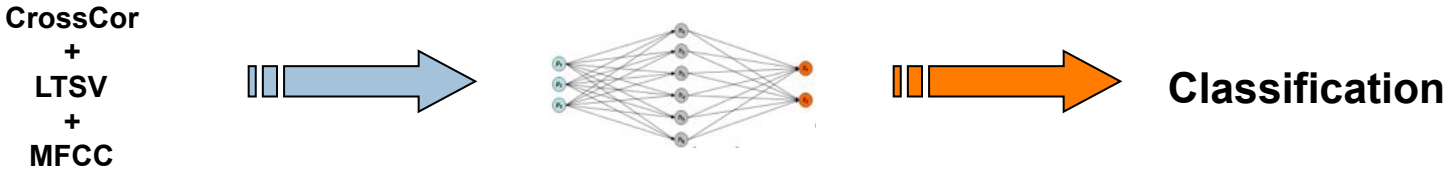
- Comment utiliser les réseaux de neurones :
  - Piste 1 : alternative directe des méthodes classiques temporelles



  - Piste 2 : alternative directe des méthodes classiques spectrales



  - Piste 3 : surcouche de décision sur un intervalle plus grand



Apprentissage/Optimisation

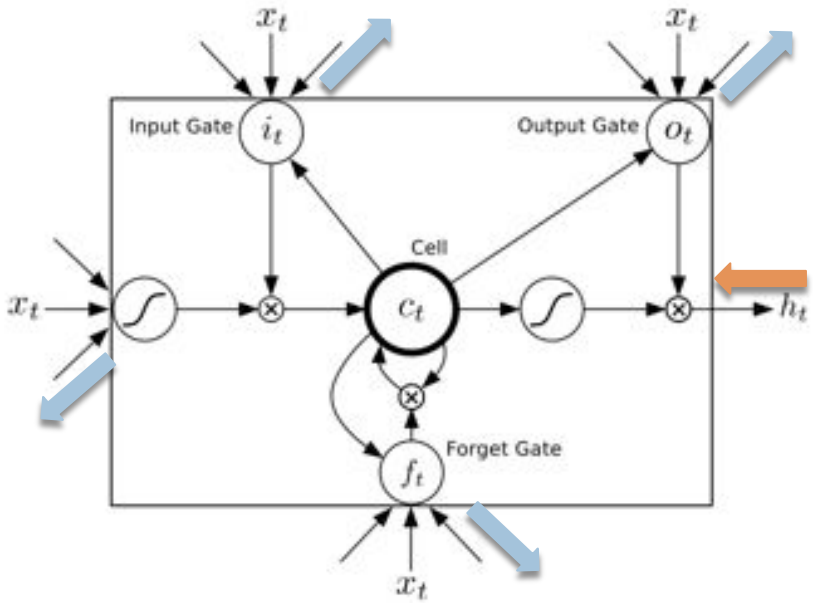
Comment choisir les paramètres

- Rétro-Propagation
  - Réseaux de neurones classiques : rétro-propagation des erreurs et optimisation par descente de gradient ;
  - Adaptation pour RNN : Back-Propagation Through Time (BPTT) ;
  - Adaptation pour LSTM : « Supervised Sequence Labelling with Recurrent Neural Networks » Alex Graves, (implémentation C++ en open source) ;
- Ces techniques ne permettent pas d'entraîner des architectures complexes ou hybrides.

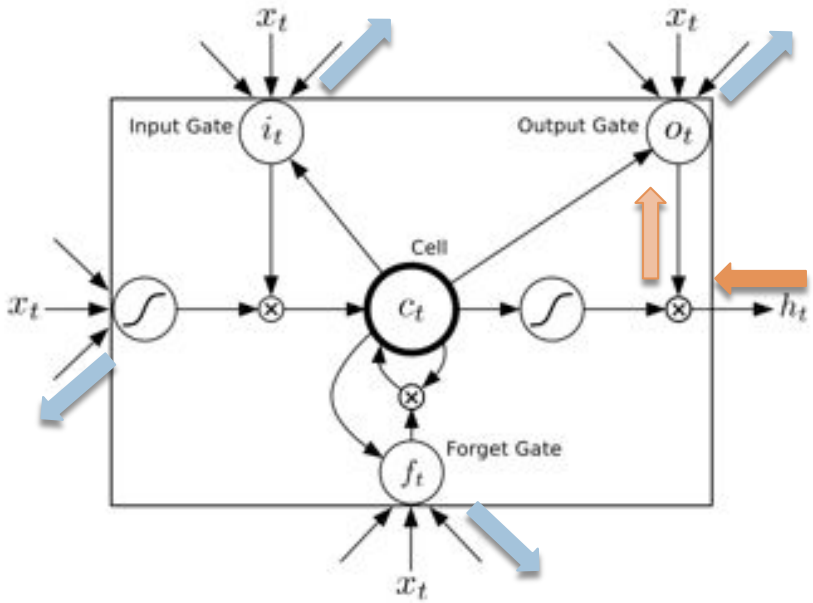
Rétro-propagation LSTM

Cell Outputs

$$\epsilon_c^t = \sum_{k=1}^{K} w_{ck}\delta_k^t + \sum_{h=1}^{H} w_{ch}\delta_h^{t+1}$$



Rétro-propagation LSTM



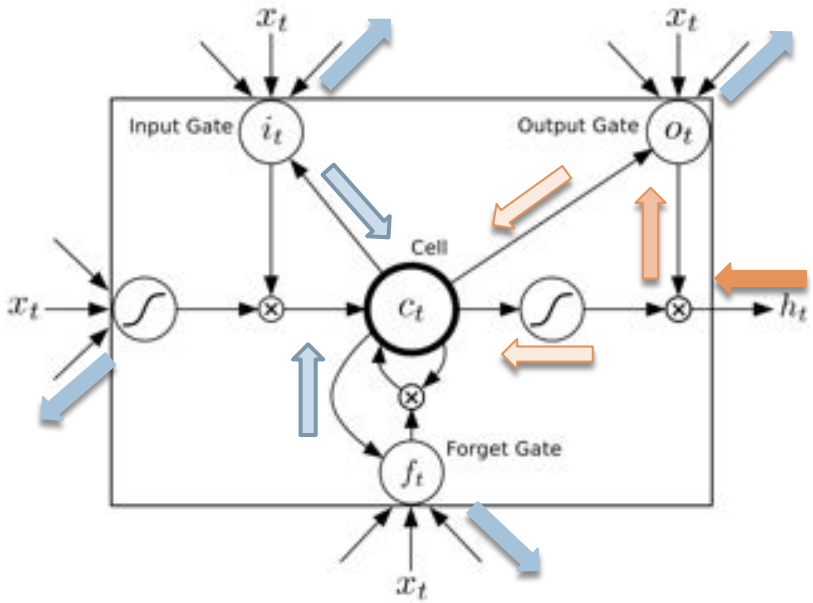
Cell Outputs

$$\epsilon_c^t = \sum_{k=1}^{K} w_{ck}\delta_k^t + \sum_{h=1}^{H} w_{ch}\delta_h^{t+1}$$

Output Gates

$$\delta_\omega^t = f'(a_\omega^t)\sum_{c=1}^{C} h(s_c^t)\epsilon_c^t$$

Rétro-propagation LSTM



Cell Outputs

$$\epsilon_c^t = \sum_{k=1}^{K} w_{ck}\delta_k^t + \sum_{h=1}^{H} w_{ch}\delta_h^{t+1}$$

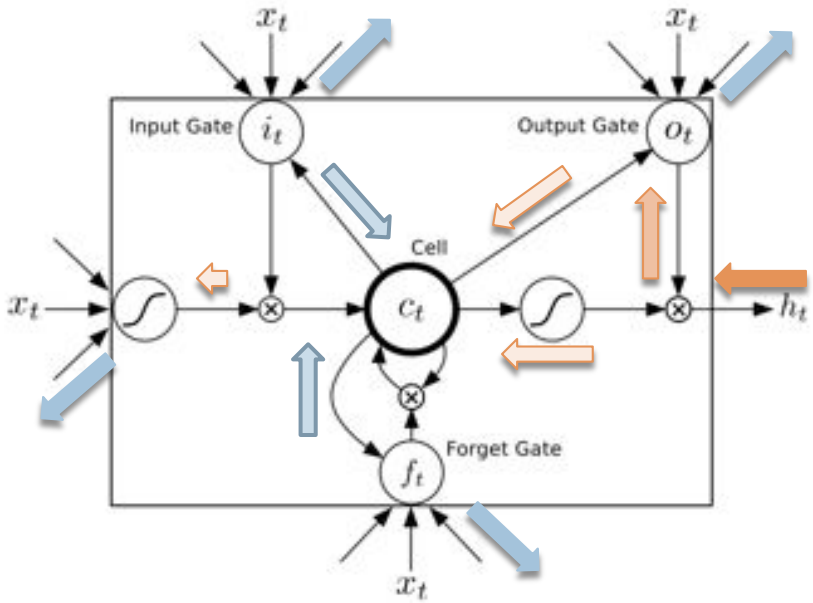
Output Gates

$$\delta_\omega^t = f'(a_\omega^t)\sum_{c=1}^{C} h(s_c^t)\epsilon_c^t$$

States

$$\epsilon_s^t = b_\omega^t h'(s_c^t)\epsilon_c^t + b_\phi^{t+1}\epsilon_s^{t+1} + w_{c\iota}\delta_\iota^{t+1} + w_{c\phi}\delta_\phi^{t+1} + w_{c\omega}\delta_\omega^t$$

Rétro-propagation LSTM



Cell Outputs

$$\epsilon_c^t = \sum_{k=1}^{K} w_{ck}\delta_k^t + \sum_{h=1}^{H} w_{ch}\delta_h^{t+1}$$

Output Gates

$$\delta_\omega^t = f'(a_\omega^t)\sum_{c=1}^{C} h(s_c^t)\epsilon_c^t$$

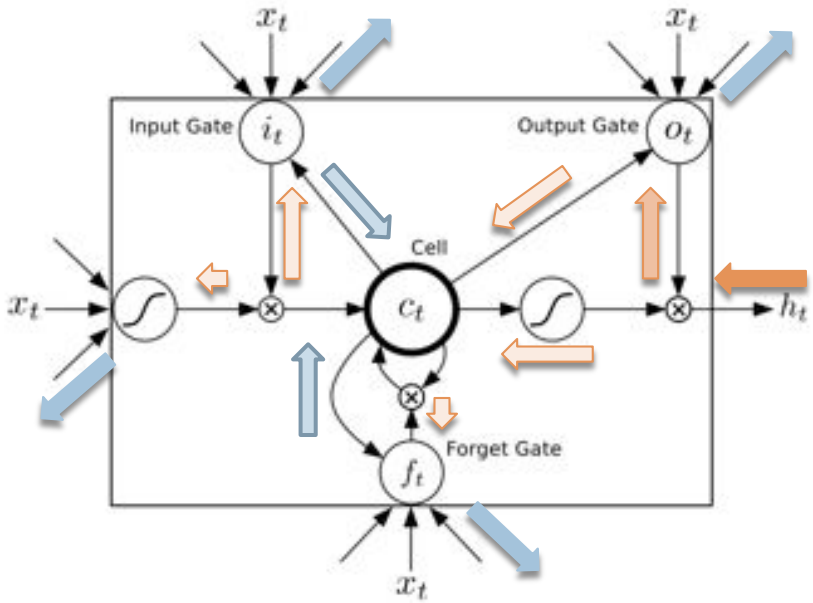
States

$$\epsilon_s^t = b_\omega^t h'(s_c^t)\epsilon_c^t + b_\phi^{t+1}\epsilon_s^{t+1} + w_{c\iota}\delta_\iota^{t+1} + w_{c\phi}\delta_\phi^{t+1} + w_{c\omega}\delta_\omega^t$$

Cells

$$\delta_c^t = b_\iota^t g'(a_c^t)\epsilon_s^t$$

Rétro-propagation LSTM



Cell Outputs

$$\epsilon_c^t = \sum_{k=1}^{K} w_{ck}\delta_k^t + \sum_{h=1}^{H} w_{ch}\delta_h^{t+1}$$

Output Gates

$$\delta_\omega^t = f'(a_\omega^t)\sum_{c=1}^{C} h(s_c^t)\epsilon_c^t$$

States

$$\epsilon_s^t = b_\omega^t h'(s_c^t)\epsilon_c^t + b_\phi^{t+1}\epsilon_s^{t+1} + w_{c\iota}\delta_\iota^{t+1} + w_{c\phi}\delta_\phi^{t+1} + w_{c\omega}\delta_\omega^t$$

Cells

$$\delta_c^t = b_\iota^t g'(a_c^t)\epsilon_s^t$$

Forget Gates

$$\delta_\phi^t = f'(a_\phi^t)\sum_{c=1}^{C} s_c^{t-1}\epsilon_s^t$$

Input Gates

$$\delta_\iota^t = f'(a_\iota^t)\sum_{c=1}^{C} g(a_c^t)\epsilon_s^t$$

Comment choisir les paramètres

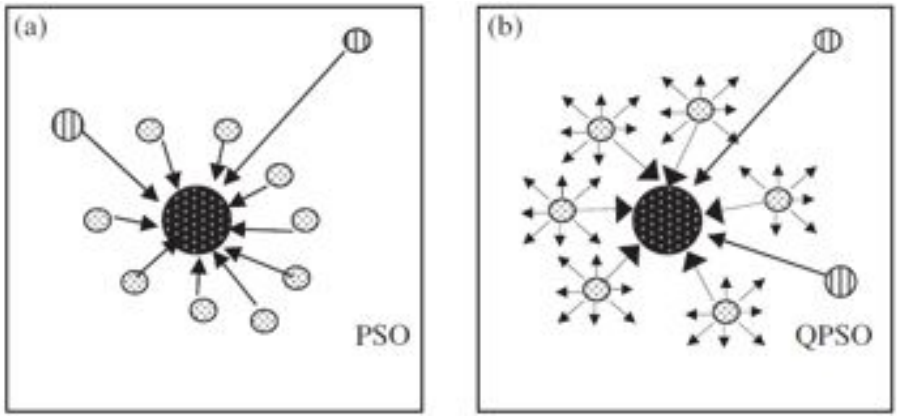
- Essaim Particulaire (Particle Swarm Optimisation)
  - Principe : métaheuristique d'optimisation se basant sur la collaboration des individus entre eux. Les particules se déplace dans l'espace suivant les équations :

$$V_{i,j}^{t+1} = wV_{i,j}^{t} + c_1 r_{i,j}^{t}\left(P_{ij}^{t} - x_{ij}^{t}\right) + c_2 R_{i,j}^{t}\left(G_j^{t} - X_{i,j}^{t}\right)$$
$$X_{i,j}^{t+1} = X_{i,j}^{t} + V_{i,j}^{t+1}$$

  - Bibliographie :
    - Kennedy and Eberhart (1995) "*Particle Swarm Optimization*".
    - Clerc and Kennedy (2002). "The particle swarm - explosion, stability, and convergence in a multidimensional complex space"
- pas besoin des dérivées partielles de la fonction à optimiser.

Comment choisir les paramètres

- QPSO : Variante quantique très efficace
  - Principe :
    - Améliore l'exploration de l'espace de recherche et évite les convergences précipitées.



  - Bibliographie :
    - QPSO : « Particle swarm optimization with particles having quantum behavior », Jun Sun ; Bin Feng ; Wenbo Xu

Comment choisir les paramètres

- QPSO : Variante quantique très efficace
  - Equations :
    - Attracteur local :

$$p_{i,j}^{t} = \varphi P_{i,j}^{t} + (1 - \varphi) G_{j}^{t}$$

    - Nouvelle position :

$$X_{i,j}^{t+1} = p_{i,j}^{t} \pm \alpha \left| C_{j}^{t} - X_{i,j}^{t} \right| \ln \left(1 / u_{i,j}^{t+1} \right)$$

Comment choisir les paramètres

- Quels paramètres :
  - paramètres de fenêtrage ;
  - paramètre de convolution ;
  - préemphase ;
  - paramètres de la fft ;
  - paramètres du banc de filtres ;
  - paramètres de lissage ;
  - paramètres spécifiques de chacun des algorithmes ;
  - poids des réseaux de neurones…

- Combien :
  - Crosscor : 35
  - LTSV : 45
  - MLP, BLSTM : 4000

Expérimentations

Segmentation

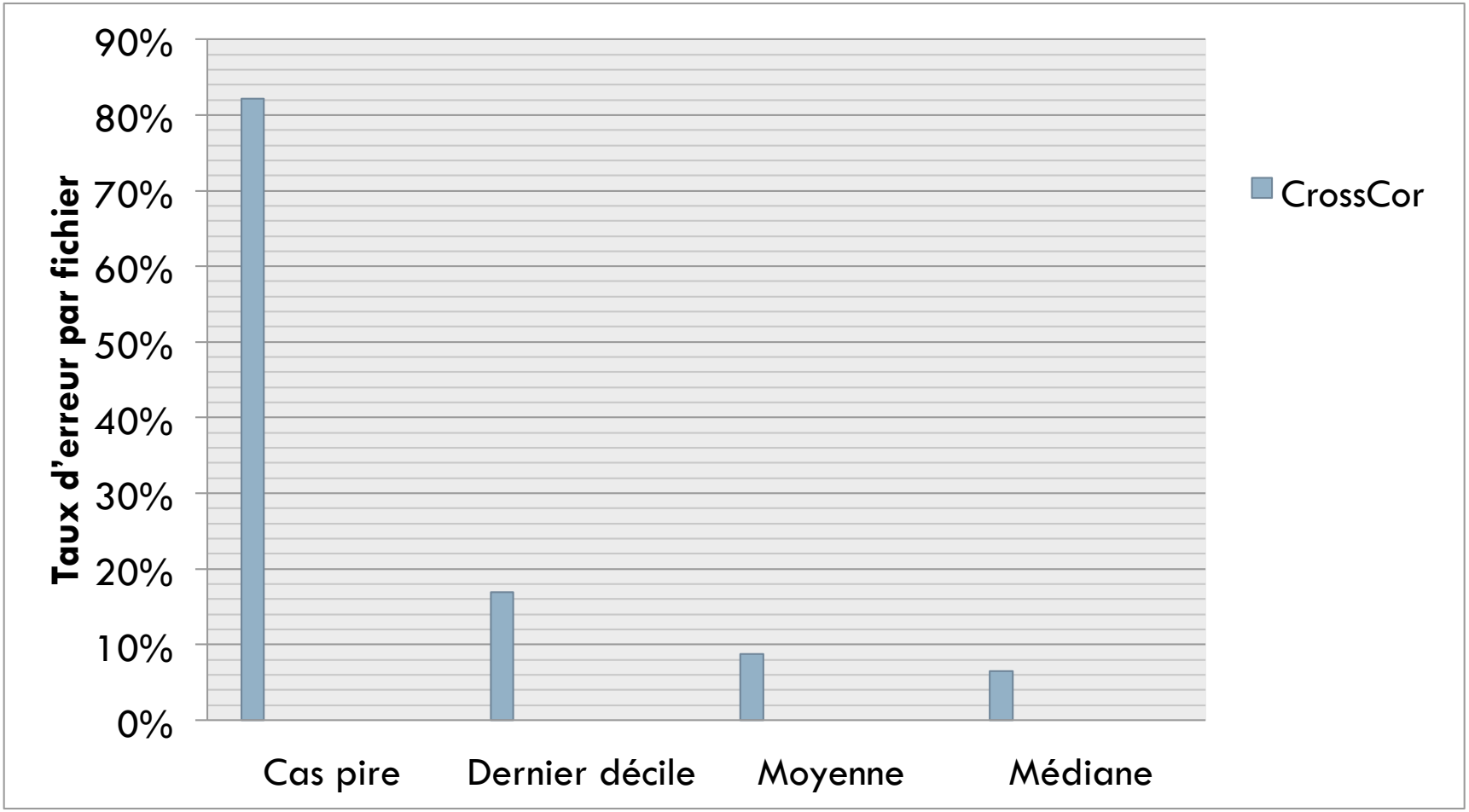
Corpus et métrique

- Corpus multilingue DGA segmenté par l'homme composé de :
  - Entrainement : 4300 fichiers (789 h de signal ; 384 h de parole)
  - Evaluation : 1026 fichiers (197 h de signal ; 96 h de parole)
  - Les difficultés principales de ce corpus sont :
    - parole spontanée ;
    - musique ;
    - erreurs et conventions d'annotation.

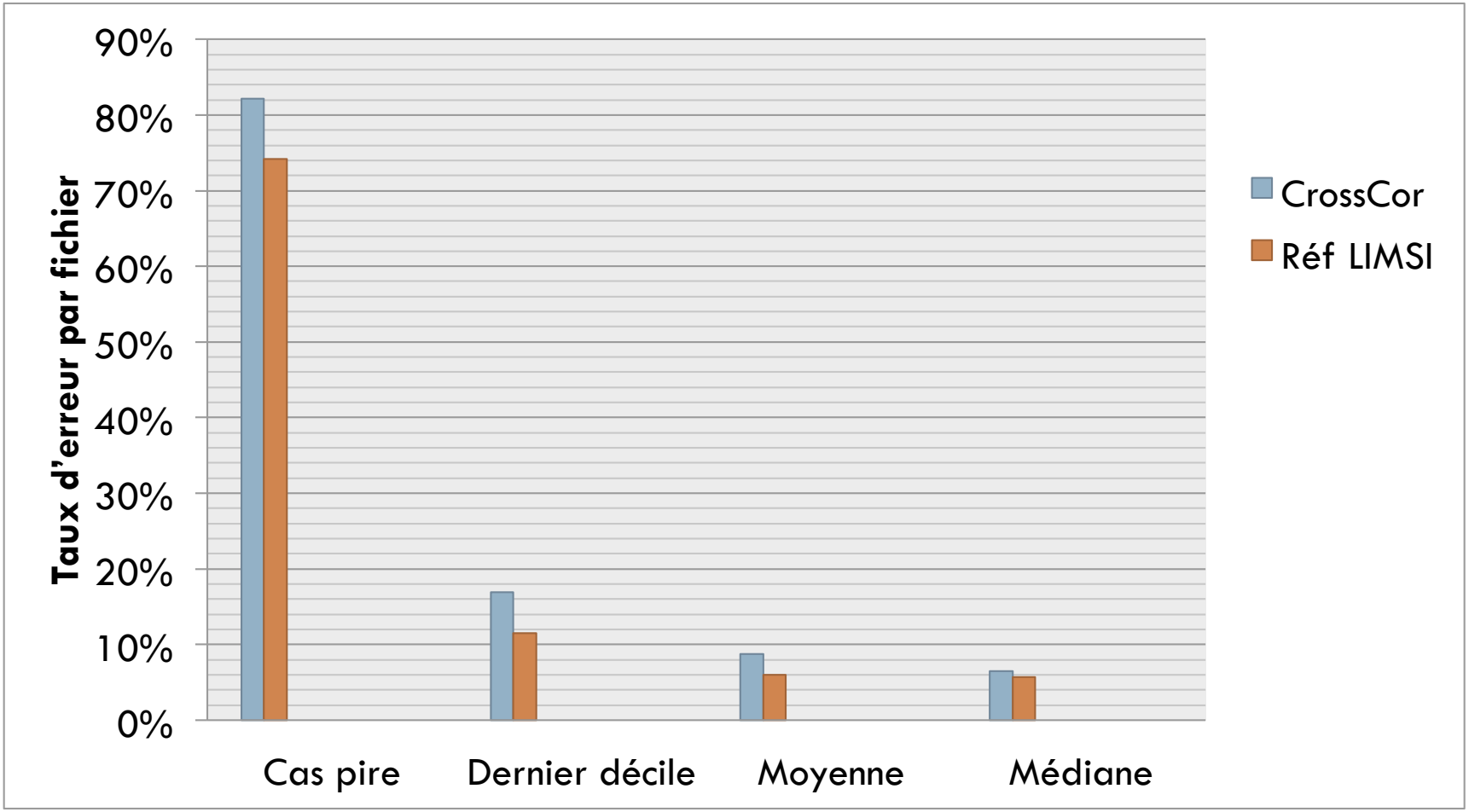
- Métrique :
  - QPSO : taux de mauvaises classifications
  - Rétro-propagation : $O = -\sum_{(x,z)\in S} z \ln y + (1-z)\ln(1-y)$

- Les méthodes testées sont systématiquement comparées à l'outil de segmentation de référence du LIMSI.

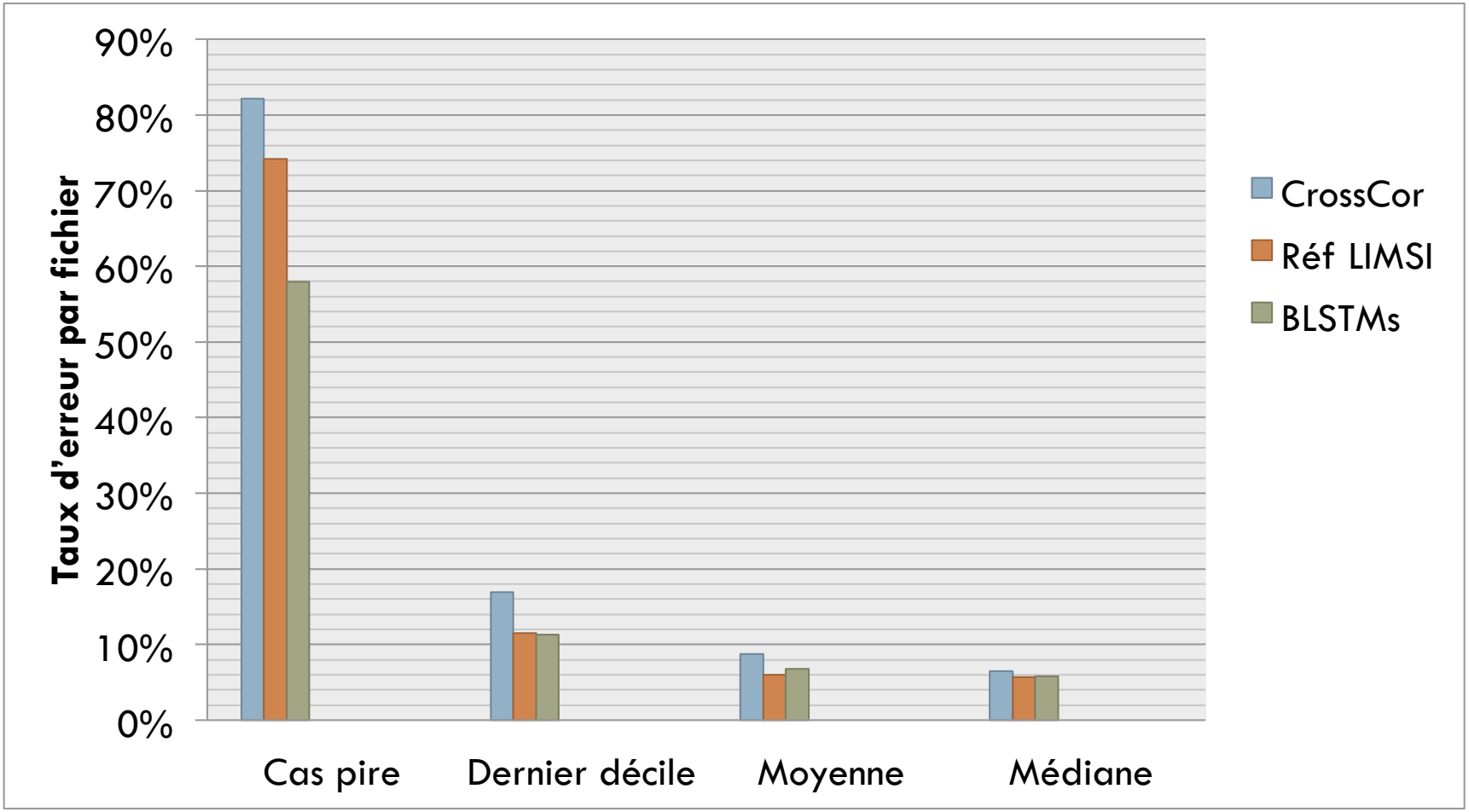
Résultats


90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Taux d'erreur par fichier
Cas pire
Dernier décile
Moyenne
Médiane
CrossCor

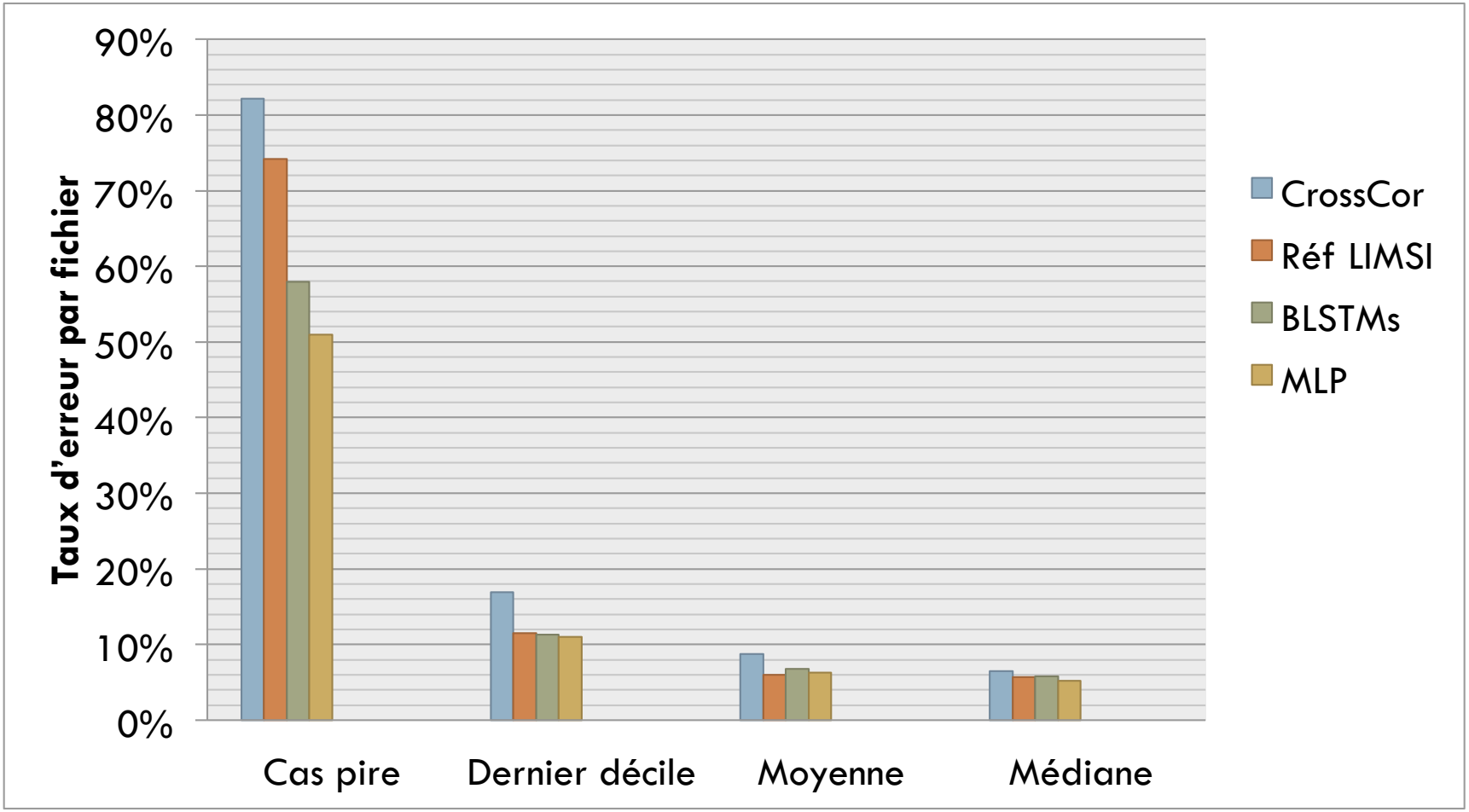
Résultats


90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Taux d'erreur par fichier
Cas pire
Dernier décile
Moyenne
Médiane
CrossCor
Réf LIMSI

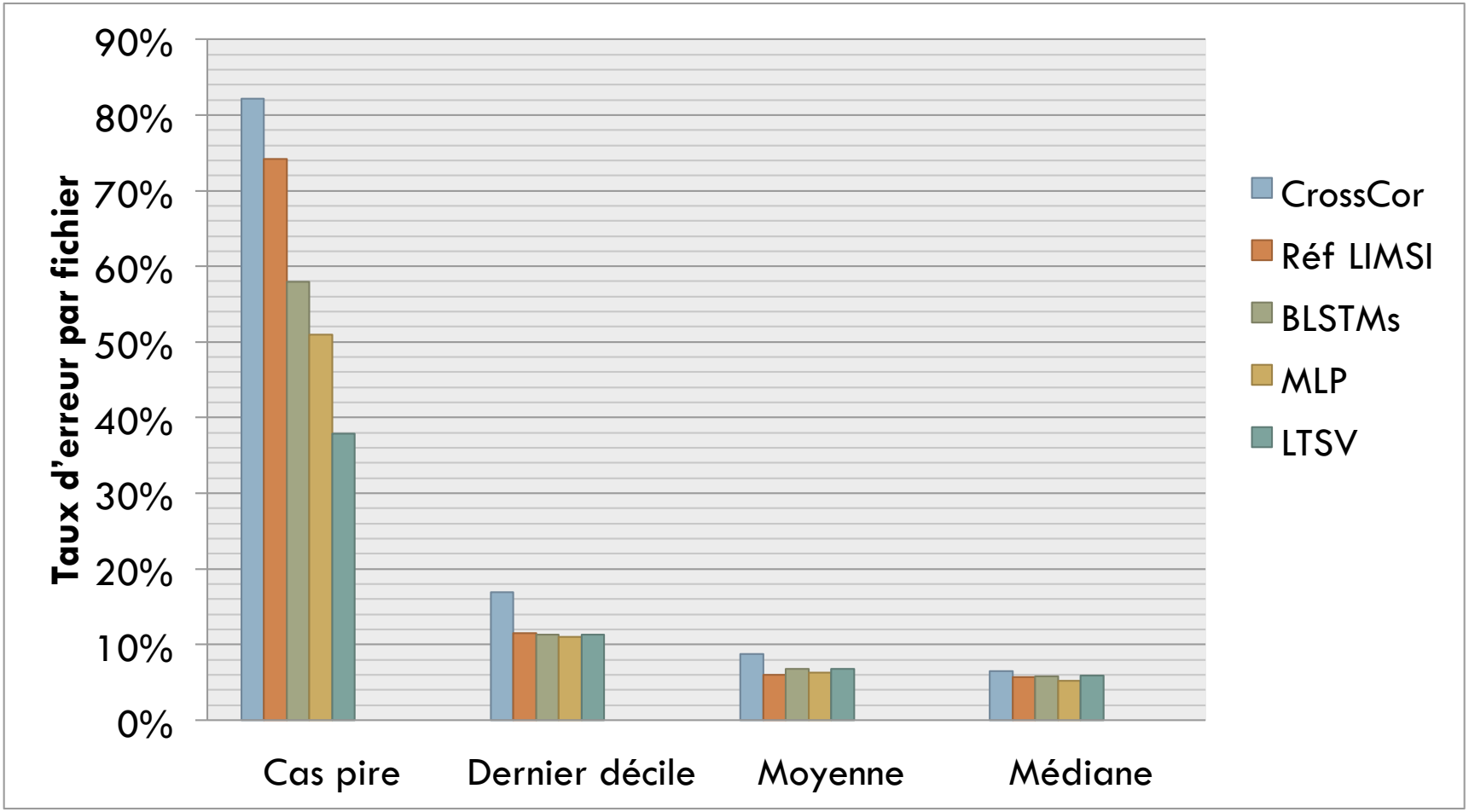
Résultats


90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Taux d'erreur par fichier
Cas pire
Dernier décile
Moyenne
Médiane
CrossCor
Réf LIMSI
BLSTMs

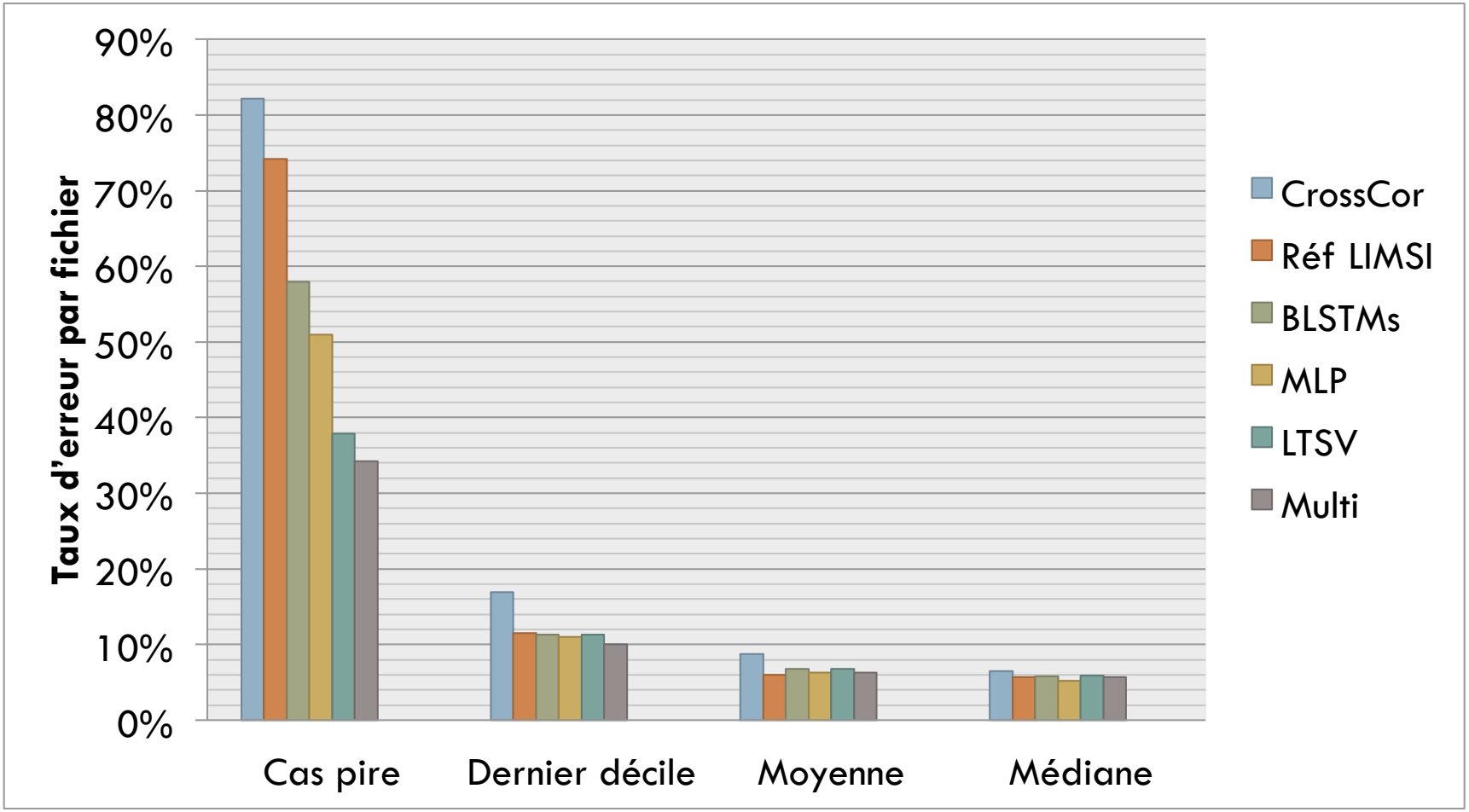
Résultats


90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Taux d'erreur par fichier
Cas pire
Dernier décile
Moyenne
Médiane
CrossCor
Réf LIMSI
BLSTMs
MLP

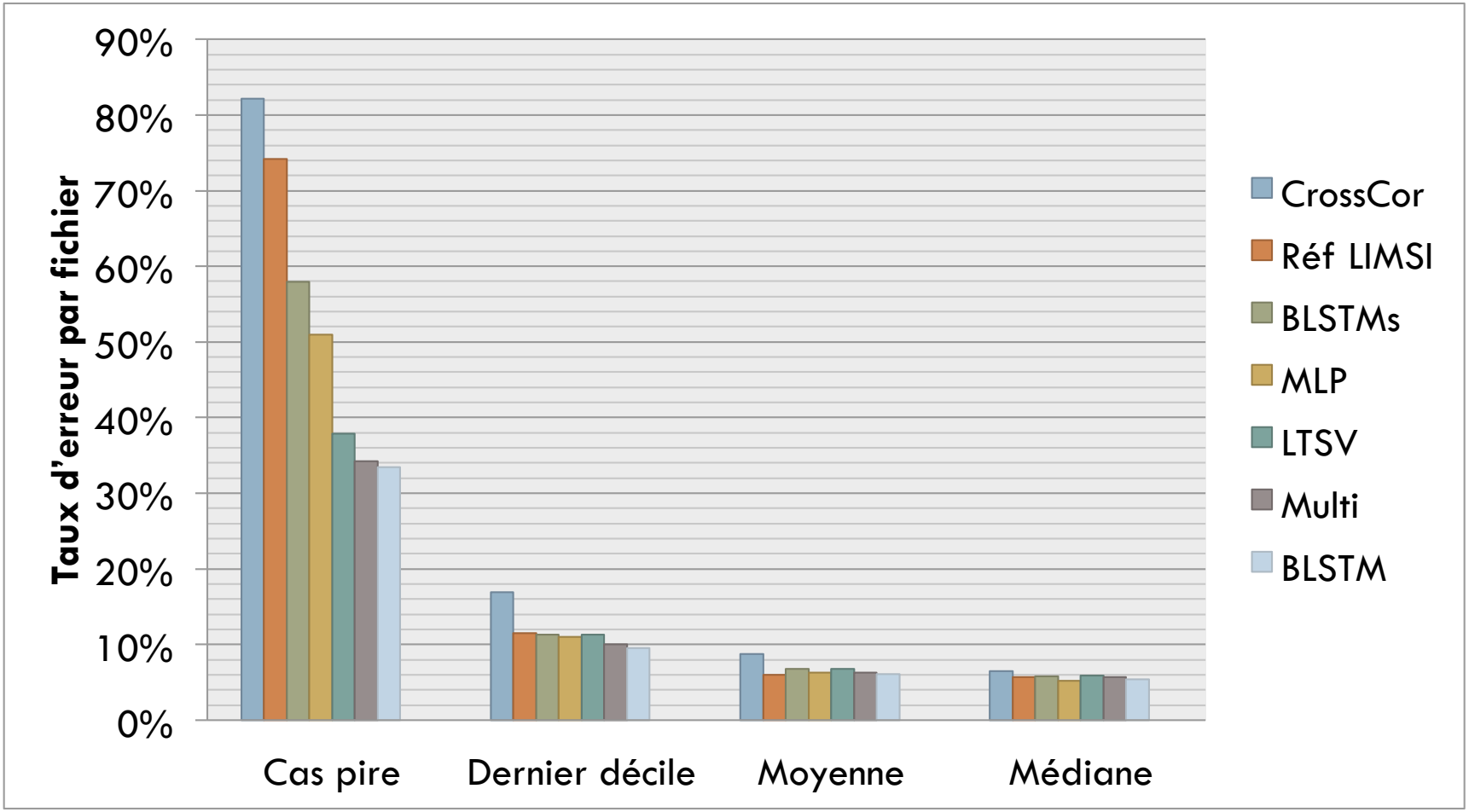
Résultats


90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Taux d'erreur par fichier
Cas pire
Dernier décile
Moyenne
Médiane
CrossCor
Réf LIMSI
BLSTMs
MLP
LTSV

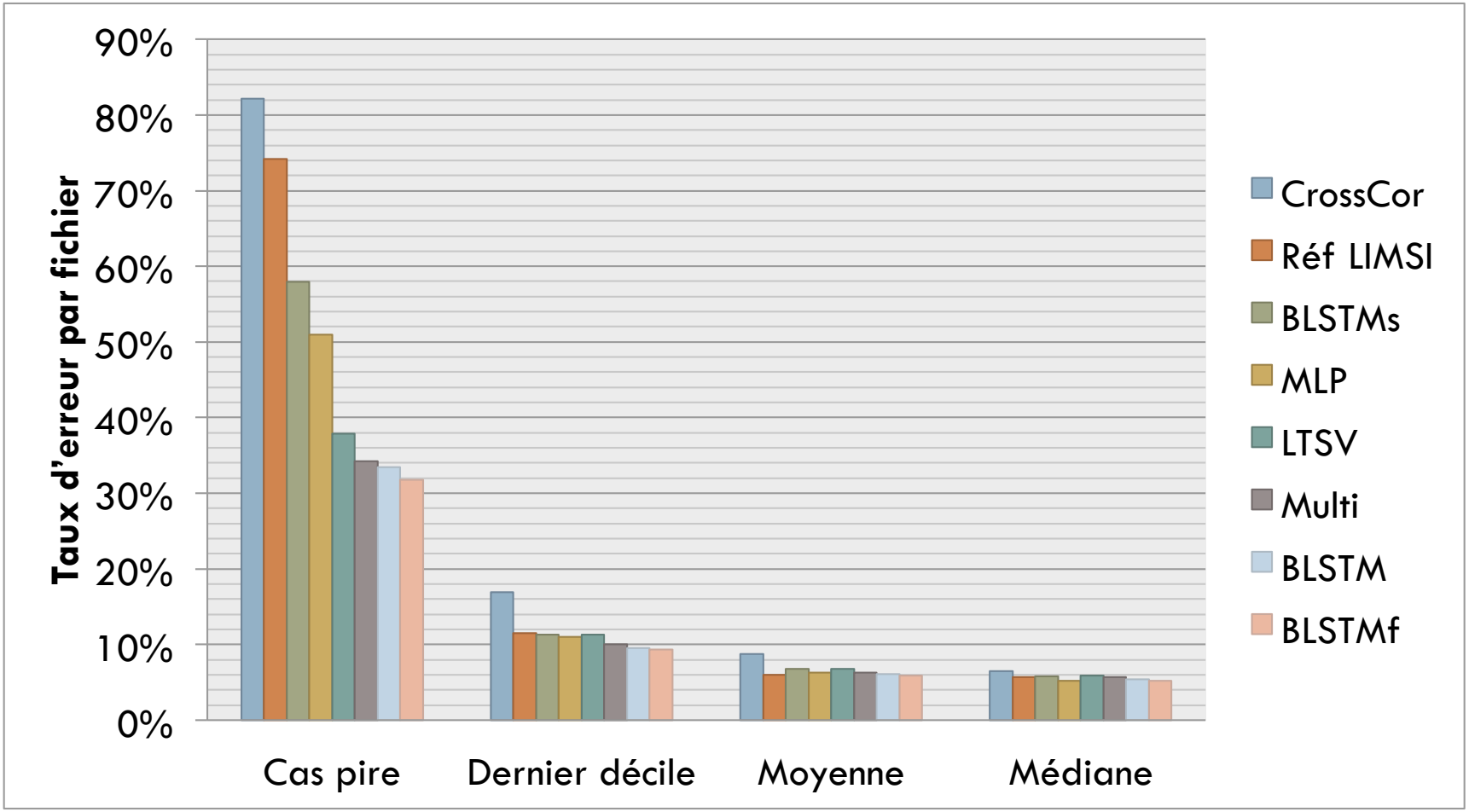
Résultats


90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Taux d'erreur par fichier
Cas pire
Dernier décile
Moyenne
Médiane
CrossCor
Réf LIMSI
BLSTMs
MLP
LTSV
Multi

Résultats


90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Taux d'erreur par fichier
Cas pire
Dernier décile
Moyenne
Médiane
CrossCor
Réf LIMSI
BLSTMs
MLP
LTSV
Multi
BLSTM

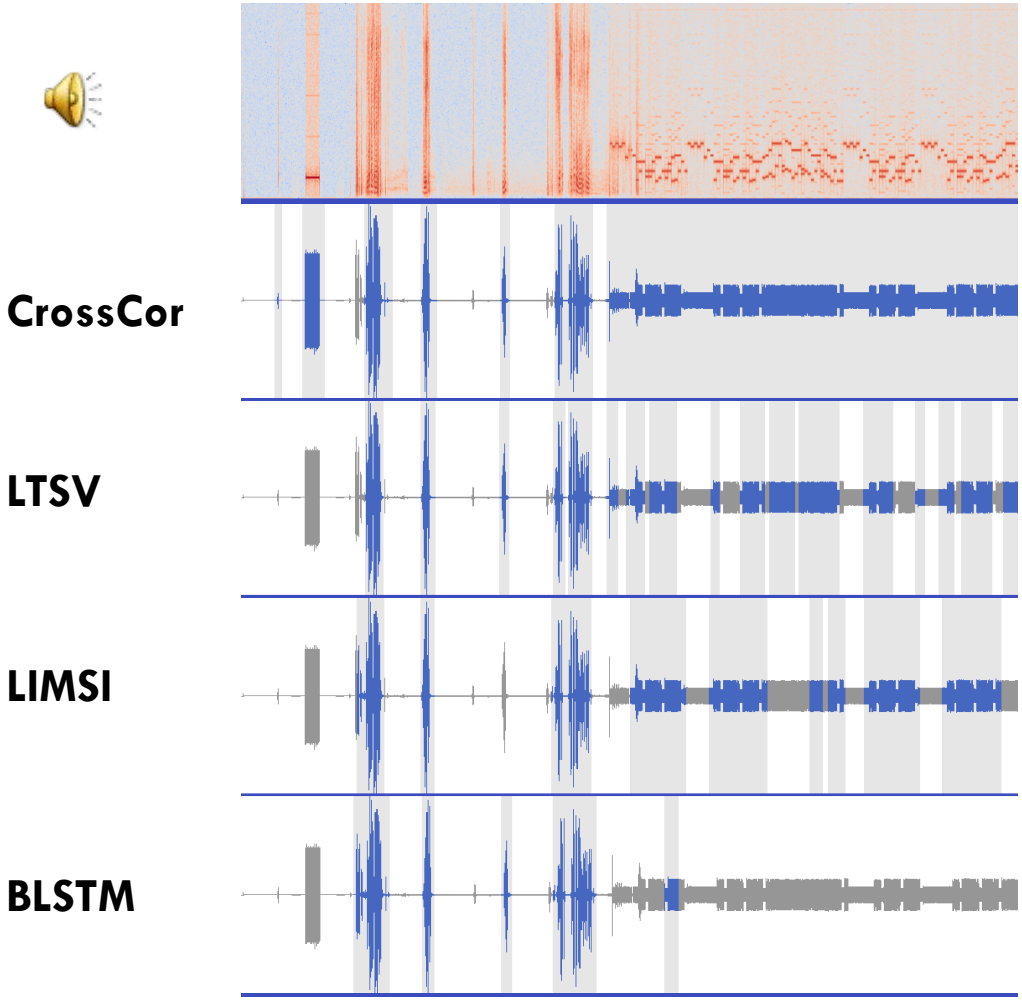
Résultats


90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Taux d'erreur par fichier
Cas pire
Dernier décile
Moyenne
Médiane
CrossCor
Réf LIMSI
BLSTMs
MLP
LTSV
Multi
BLSTM
BLSTMf

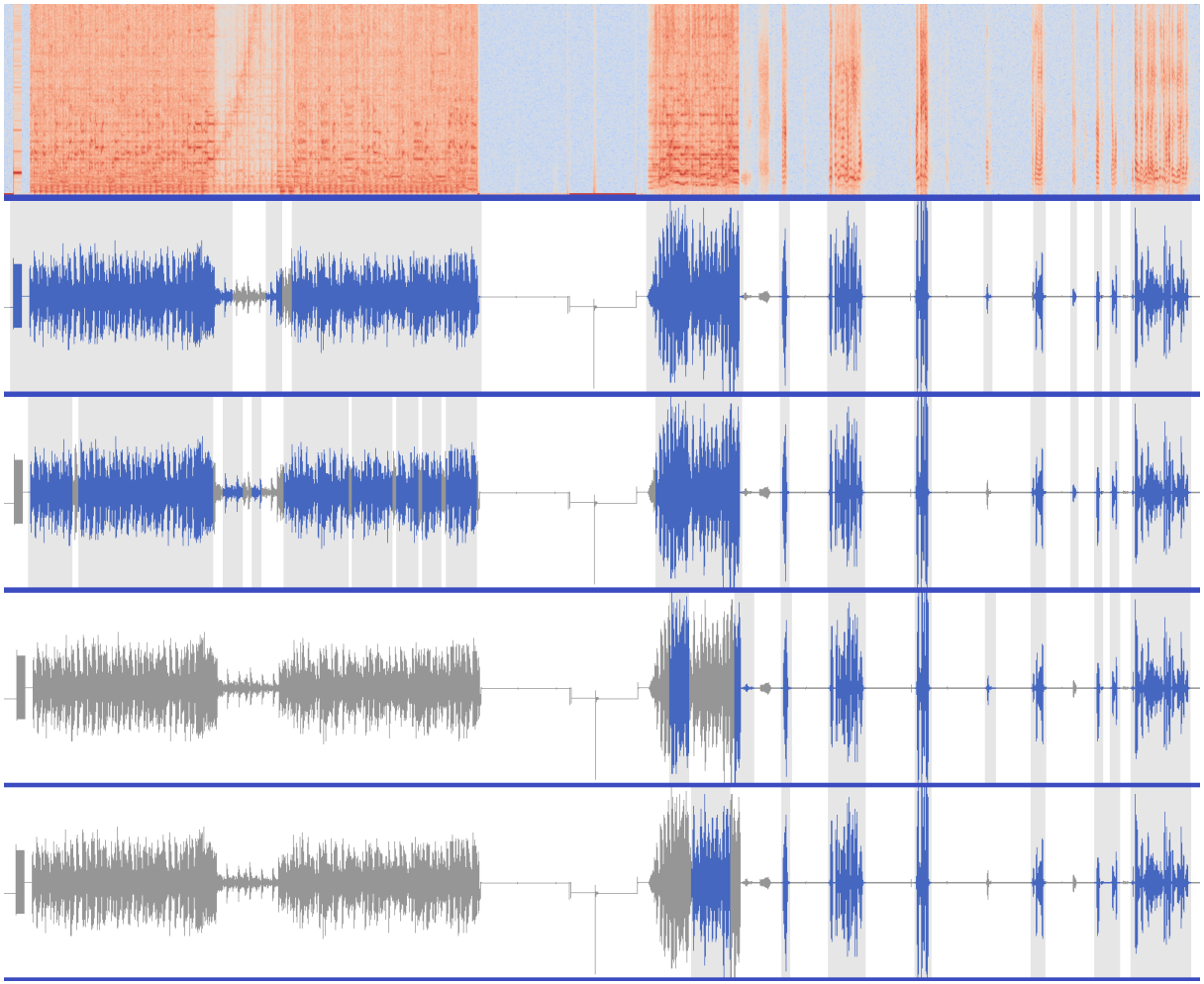
Résultats

	CrossCor	Réf LIMSI	BLSTMs	MLP	LTSV	Multi	BLSTM	BLSTMf
Cas pire	82,2%	74,2%	58,0%	51,0%	37,9%	34,2%	**33,4%**	**31,8%**
Dernier décile	16,9%	11,5%	11,3%	11,1%	11,3%	10,0%	**9,5%**	**9,3%**
Moyenne	8,8%	6,0%	6,8%	6,3%	6,8%	6,3%	**6,1%**	**5,9%**
Médiane	6,5%	5,7%	5,8%	5,3%	5,9%	5,7%	**5,4%**	**5,2%**

Cas intéressant



Cas intéressant



Expérimentations

Transcription

Corpus et métrique

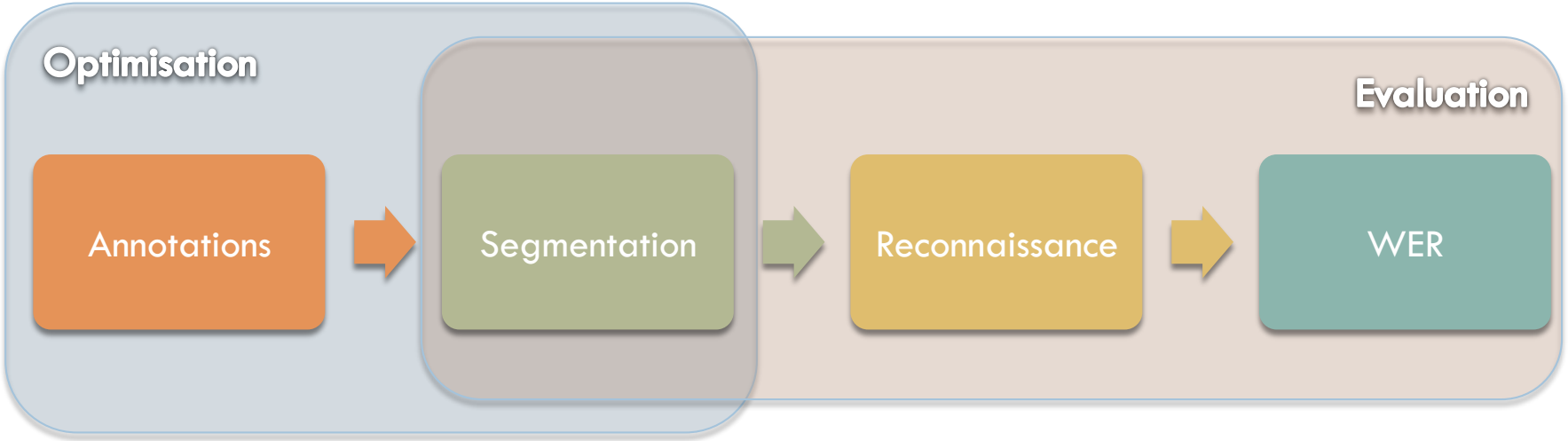
- Programme IARPA Babel :
  - flat-start obligatoire
  - parole spontanée

- Corpus vietnamien transcrit par l'homme composé de :
  - Entrainement : 1171 fichiers (180 h de signal ; 80 h de parole)
  - Evaluation : 194 fichiers (30 h de signal ; 12 h de parole)

- Métrique :
  - Word Error Rate = (S+D+I)/N

- Seule la segmentation est optimisée et les méthodes testées sont systématiquement comparées à l'outil de segmentation de référence du LIMSI et à celui de BBN.

Schéma du problème



- Critères d'optimisation :
  - QPSO : $O = \alpha \sum_{p \in P} (y < s_P) + (1-\alpha) \sum_{b \in B} (y > s_B)$
  - Rétro-propagation : $O = -\alpha \sum_{p \in P} \ln(y) - (1-\alpha) \sum_{b \in B} \ln(1-y)$

Résultats

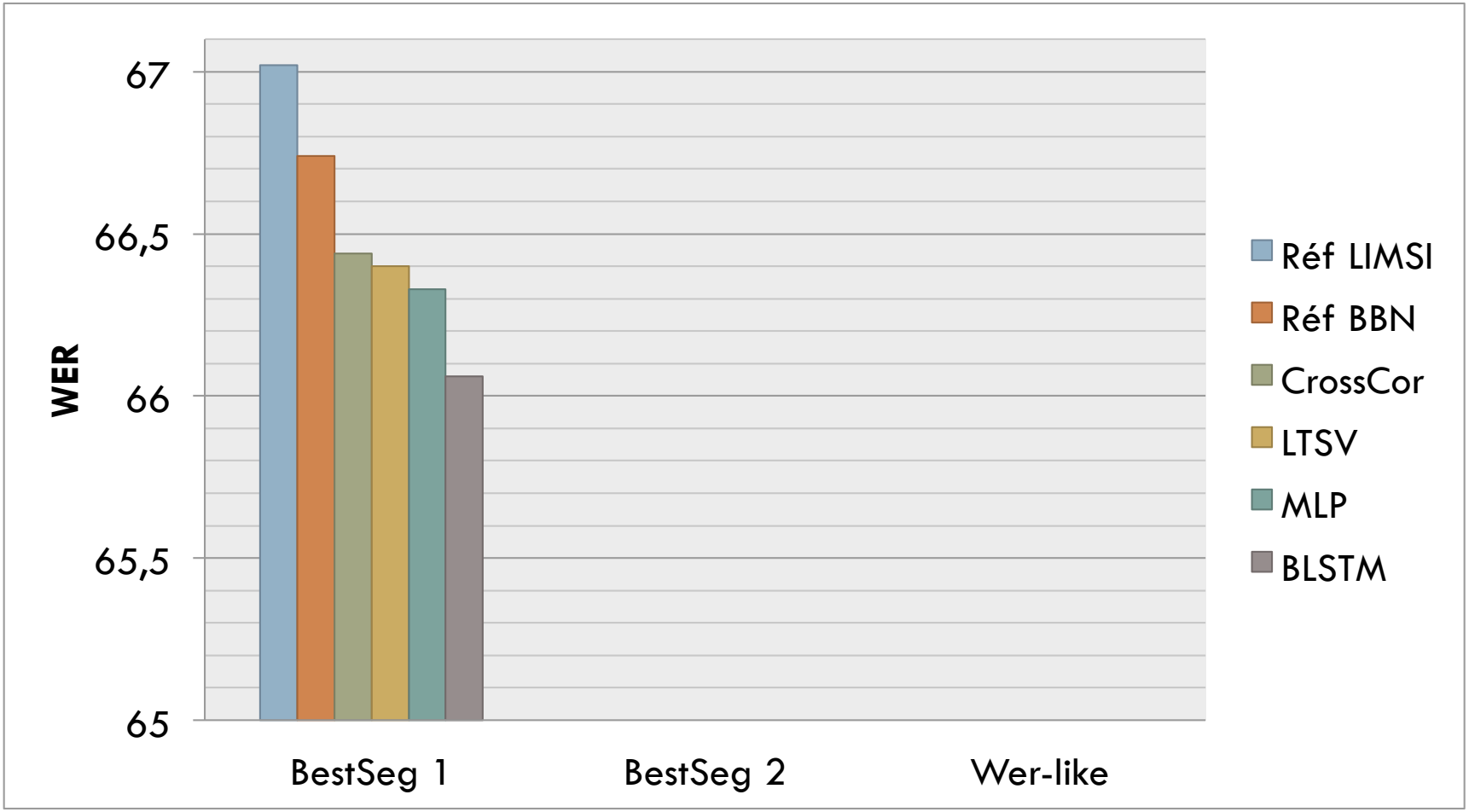
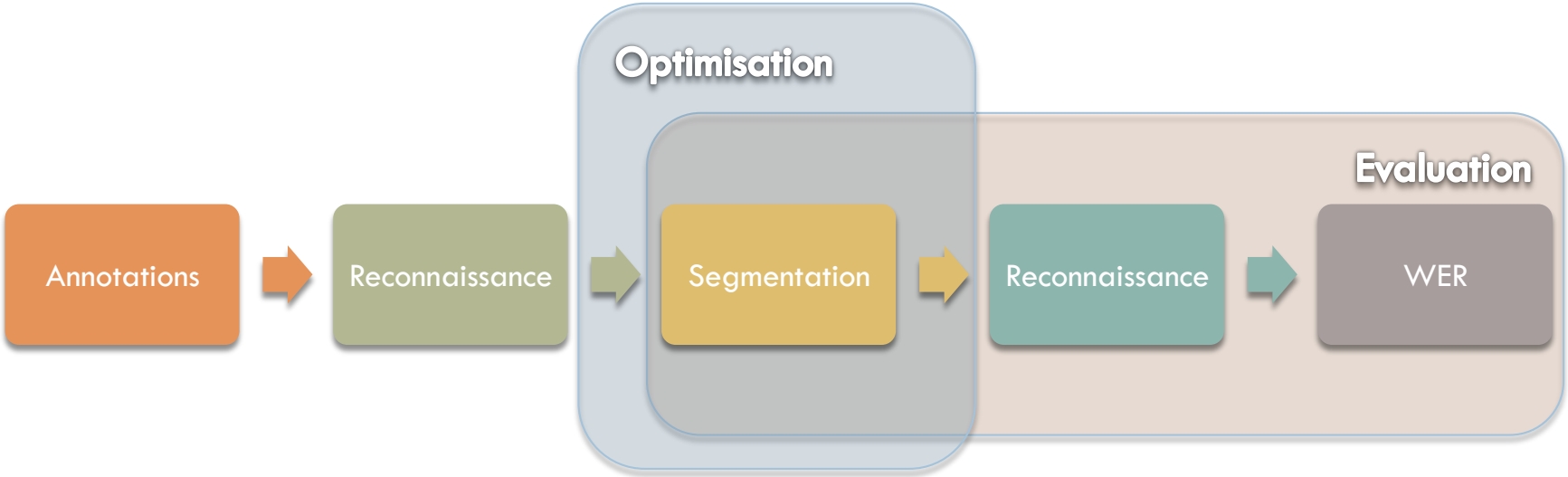

67
66,5
66
65,5
65
WER
BestSeg 1
BestSeg 2
Wer-like
Réf LIMSI
Réf BBN
CrossCor
LTSV
MLP
BLSTM

Schéma du problème



- Critères d'optimisation :
  - QPSO : $O = \alpha \sum_{p \in P} (y < s_P) + (1-\alpha) \sum_{b \in B} (y > s_B)$
  - Rétro-propagation : $O = -\alpha \sum_{p \in P} \ln(y) - (1-\alpha) \sum_{b \in B} \ln(1-y)$

Résultats

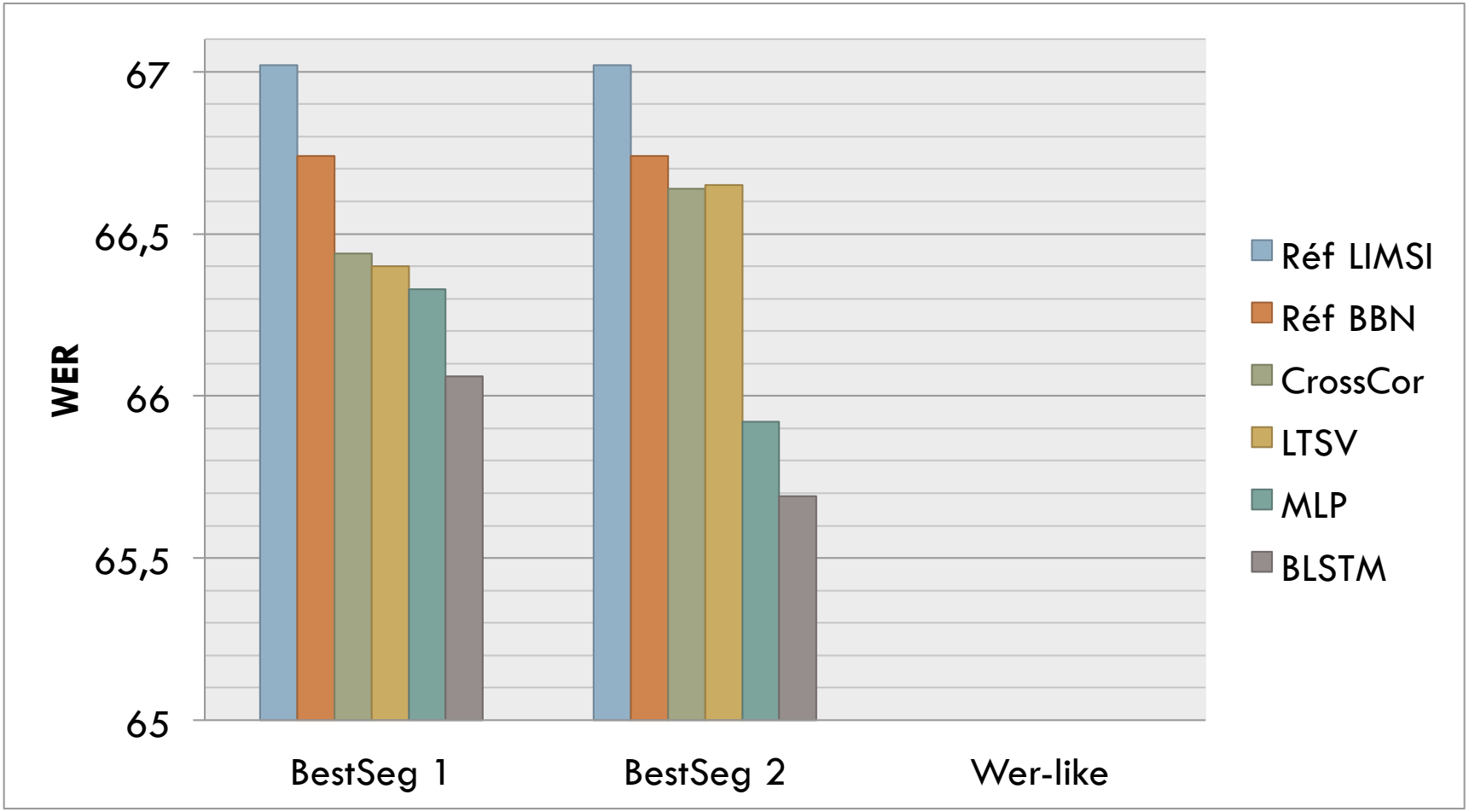
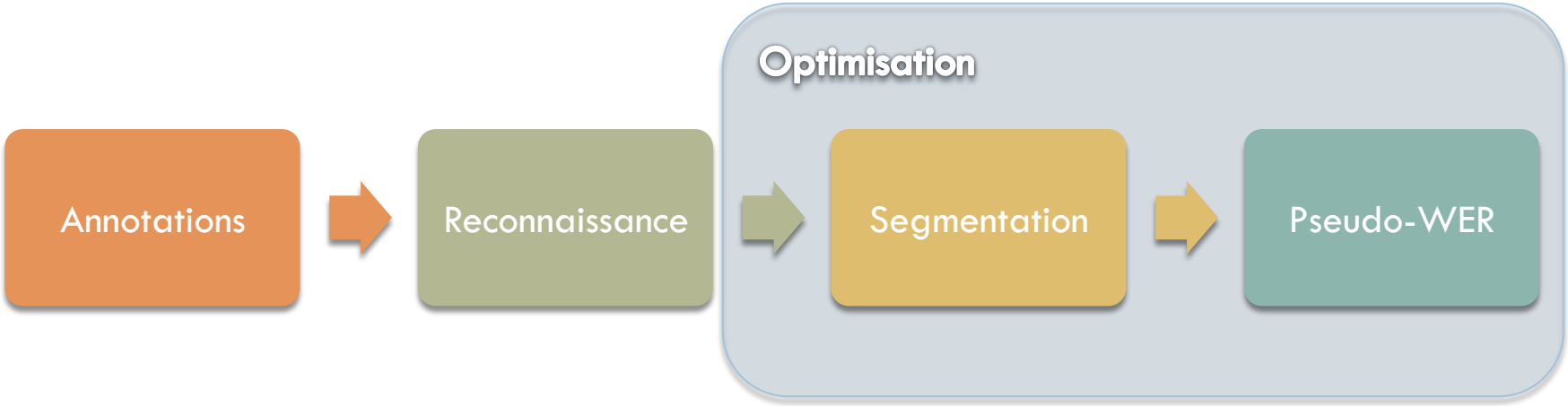

67
66,5
66
65,5
65
WER
BestSeg 1
BestSeg 2
Wer-like
Réf LIMSI
Réf BBN
CrossCor
LTSV
MLP
BLSTM

Schéma du problème



- Critère d'optimisation :
  - QPSO : WERmod = $(S+D+I+\tau_c+\tau_i)/N$

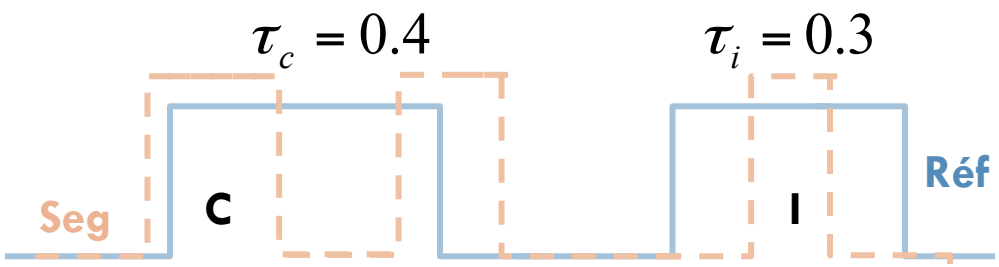
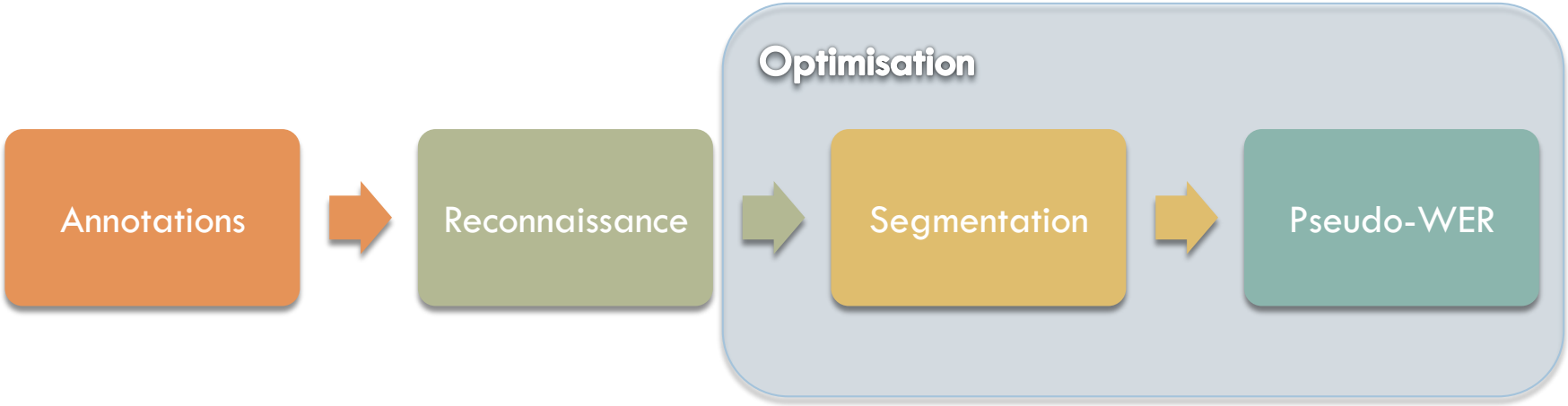
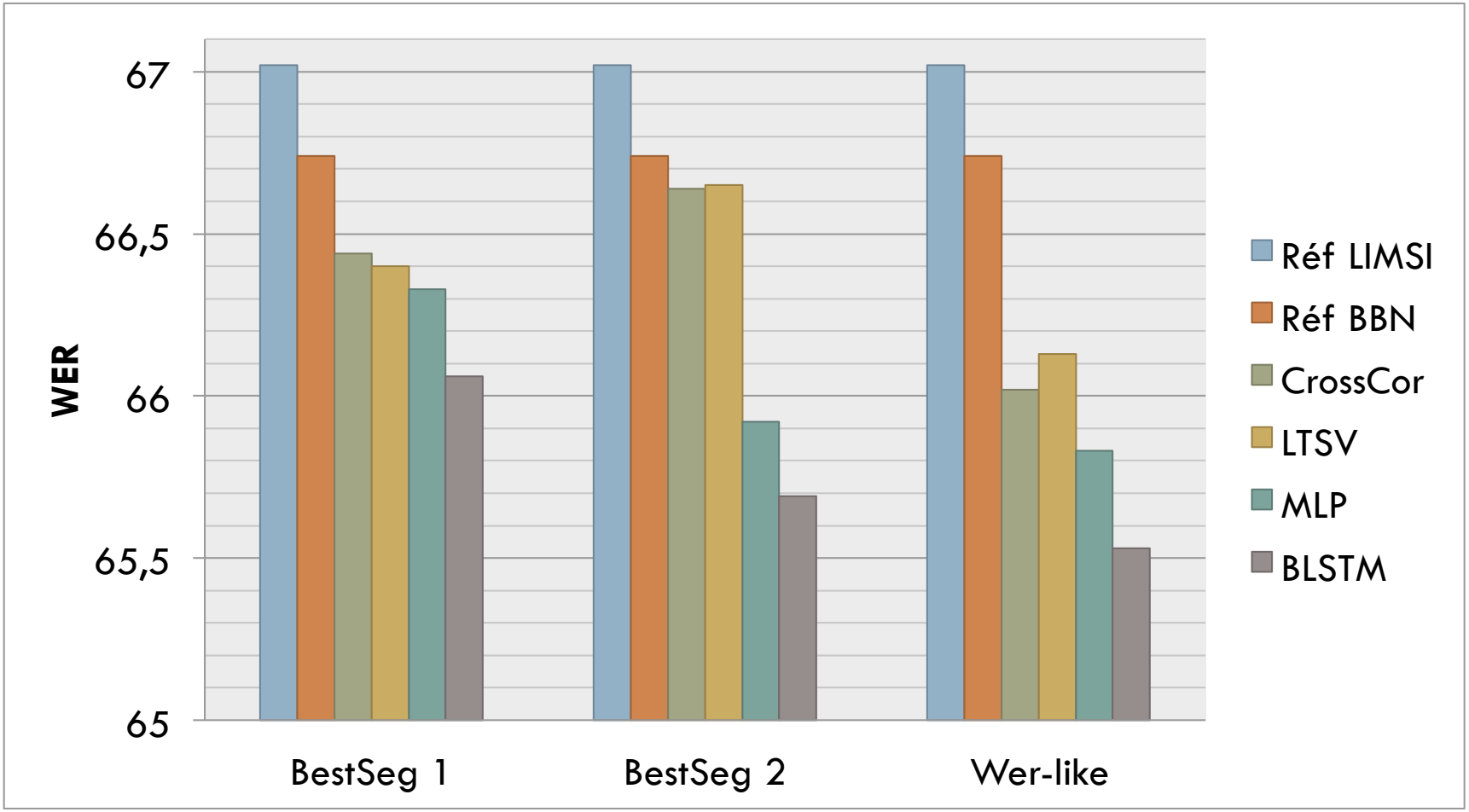


Schéma du problème



- Critère d'optimisation :
  - QPSO : WERmod = $(S+D+I+\tau_c+\tau_i)/N$
  - Rétro-propagation : $$O = -\sum_{w \in W_P}\left(\frac{1}{\delta_w}\sum \ln(y)\right) - \sum_{w \in W_I}\left(\frac{1}{\delta_w}\sum \ln(1-y)\right) - \frac{1}{\delta_B}\sum_{b \in B}\ln(1-y)$$
    $$\forall w \in W_P \cup W_I \ , \quad \delta_B >> \delta_w$$

Résultats


67
66,5
66
65,5
65
WER
BestSeg 1
BestSeg 2
Wer-like
Réf LIMSI
Réf BBN
CrossCor
LTSV
MLP
BLSTM

Résultats

- Respect de la contrainte flat-start

- Gains obtenus avec les BLSTM avant Reco :
  - Insertions ↘ 1600 mots ;
  - Supressions ↗ 700 mots.

- Gains obtenus avec les BLSTM après Reco :
  - Insertions ↘ 1800 mots ;
  - Supressions ↘ 500 mots.

Résultats

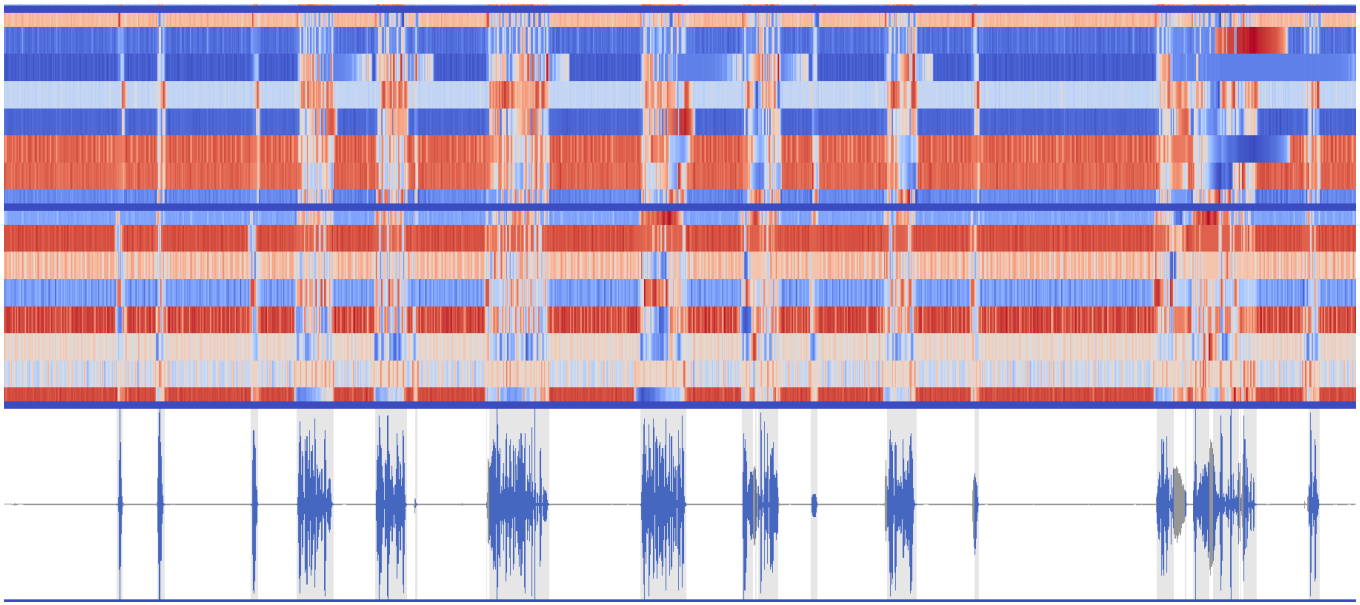
- Impact des différentes techniques d'optimisation :

Optimisation	WER
Référence	66,74
BackProp	67,02
BackProp+QPSO	66,53
QPSO	65,87
QPSO+BackProp	66,00
QPSO+BackProp+QPSO	65,62
Hybride QPSO-BackProp	**65,53**

Bonnes pratiques

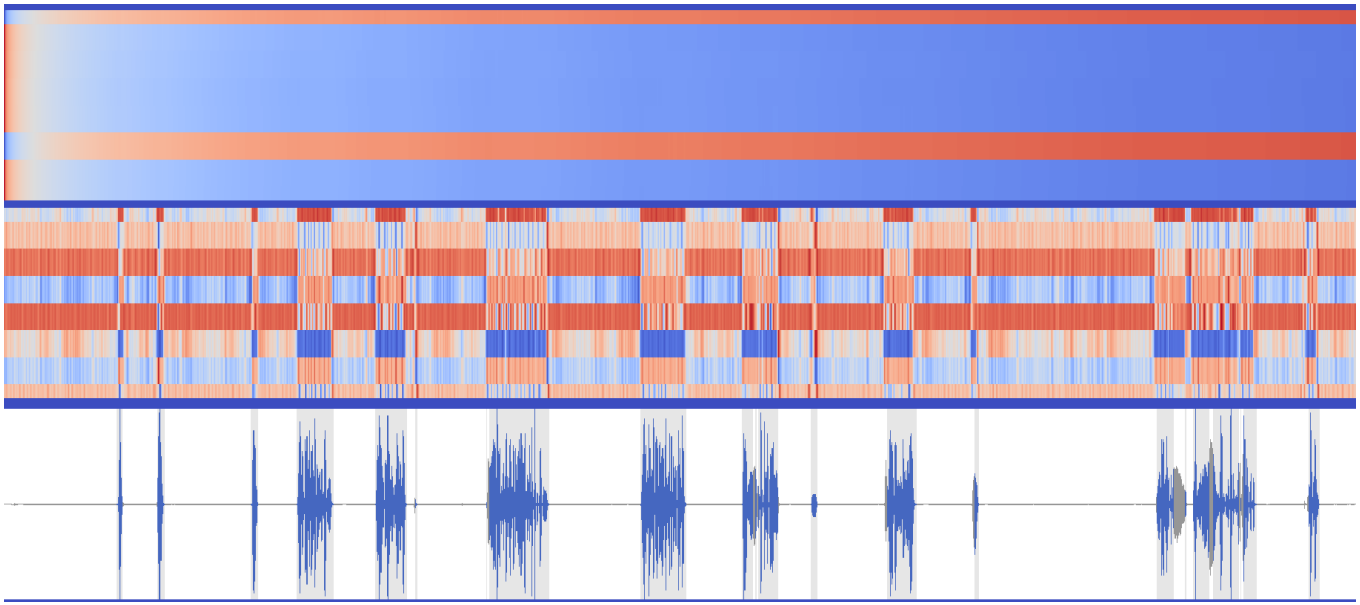
Comportement interne des LSTMs

- Attention aux séquences longues avec un motif cible récurrent !
  - comportement équilibré :



Comportement interne des LSTMs

- Attention aux séquences longues avec un motif cible récurrent !
  - dérive lors de l'apprentissage :



Conclusions et perspectives

Conclusions & perspectives

- Conclusions :
  - Très grand intérêt des LSTM pour le traitement automatique de la parole
  - Technique générique pouvant être utilisée sur tout type de séquences
  - Relativement peu couteux en temps de calcul :
    - segmentation en ~0.001 durée de signal

- Perspectives :
  - Valider le processus d'optimisation sur d'autres langues du programme Babel
  - Comparer CMA-ES à QPSO
  - Utiliser les LSTM pour la reconnaissance de la langue

MERCI POUR VOTRE ATTENTION !