الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي و البحث العلمي

جامعة وهران للعلوم و التكنولوجيا محمد بوضياف

Université des Sciences et de la technologie d'Oran- Mohammed Boudiaf
Faculté : Mathématique et Informatique

Département : Informatique

Thèse de Doctorat en sciences

Contribution par des Méthodes Statistiques à l'amélioration de la Reconnaissance Automatique de la Parole

Présentée et defendue par :

Mme MEZZOUDJ Ep. BOUMAZZA Freha

Oran, le mardi 23/10/2018

Introduction

- La Reconnaissance Automatique de la Parole (RAP) vise à reproduire la capacité cognitive des humains à reconnaître le discours oral.

Introduction

- La Reconnaissance Automatique de la Parole (RAP) vise à reproduire la capacité cognitive des humains à reconnaître le discours oral.

- 2 classes de problèmes pour la RAP :
  1. les caractéristiques et les modèles acoustiques ;
  2. les **modèles de langage**.

Introduction

Les axes d'investigation, qui nous, intéressent sont :

- Reconnaissance Automatique de la Parole (RAP) ? ?
- Quelles sont les nouvelles techniques d'adaptation de la modélisation acoustiques ?

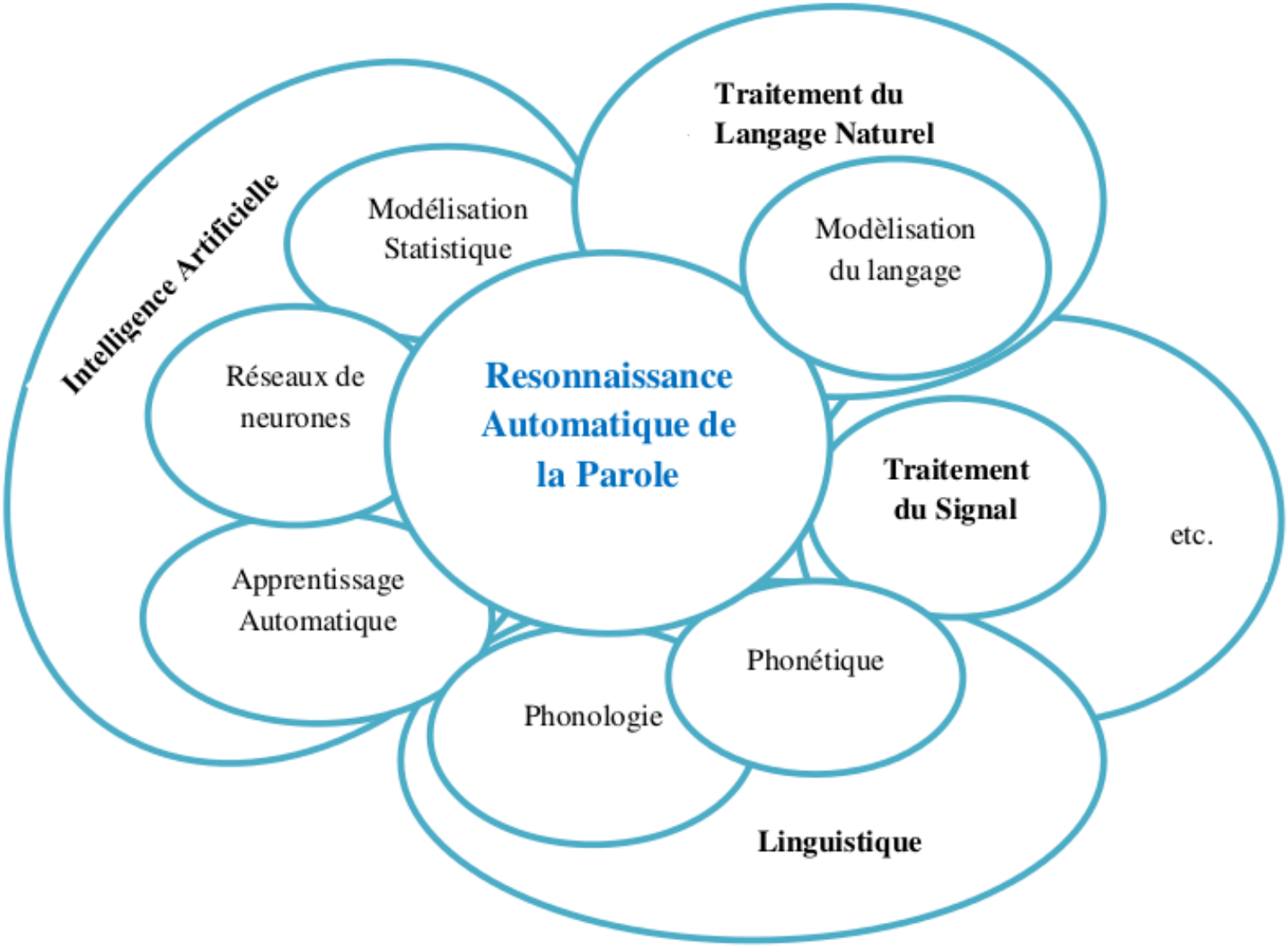
Introduction

Les axes d'investigation, qui nous, intéressent sont :

- Reconnaissance Automatique de la Parole (RAP) ? ?
- Quelles sont les nouvelles techniques d'adaptation de la modélisation acoustiques ?
- Quel est l'intéret d'un modèle de langage n-gramme vs. des modèles de langage neuronaux ?
- Quel est l'avantage d'appliquer la sélection sur des données textuelles pour la modèlisation du langage ? et comment nous avons procédé ?

Plan de travail

La RAP est liée à diverses disciplines :



Reconnaissance Automatique de la Parole

Reconnaissance Automatique de la Parole

La **RAP** (Reconnaissance Automatique de la Parole) est passée de la reconnaissance :

- des phonèmes, des mots isolés ou de chaînes de commandes isolées ;
- avec un petit vocabulaire fermé ;
- enregistrés dans un **environnement contrôlé** ;
- Monolocuteur/ Dépendant Locuteur ;

les automates d'états finis et les grammaires suffisaient ;

Au **LVCSR** (Large Vocabulary Continuous Speech Recognition) :

- la parole continue, et même spontanée et conversationelle ;
- avec grand vocabulaire (10 000 - 64 000 - 100 000 mots) ;
- enregistrée dans des **situations réelles** ;
- Multilocuteur/ Indépendant du Locuteur.

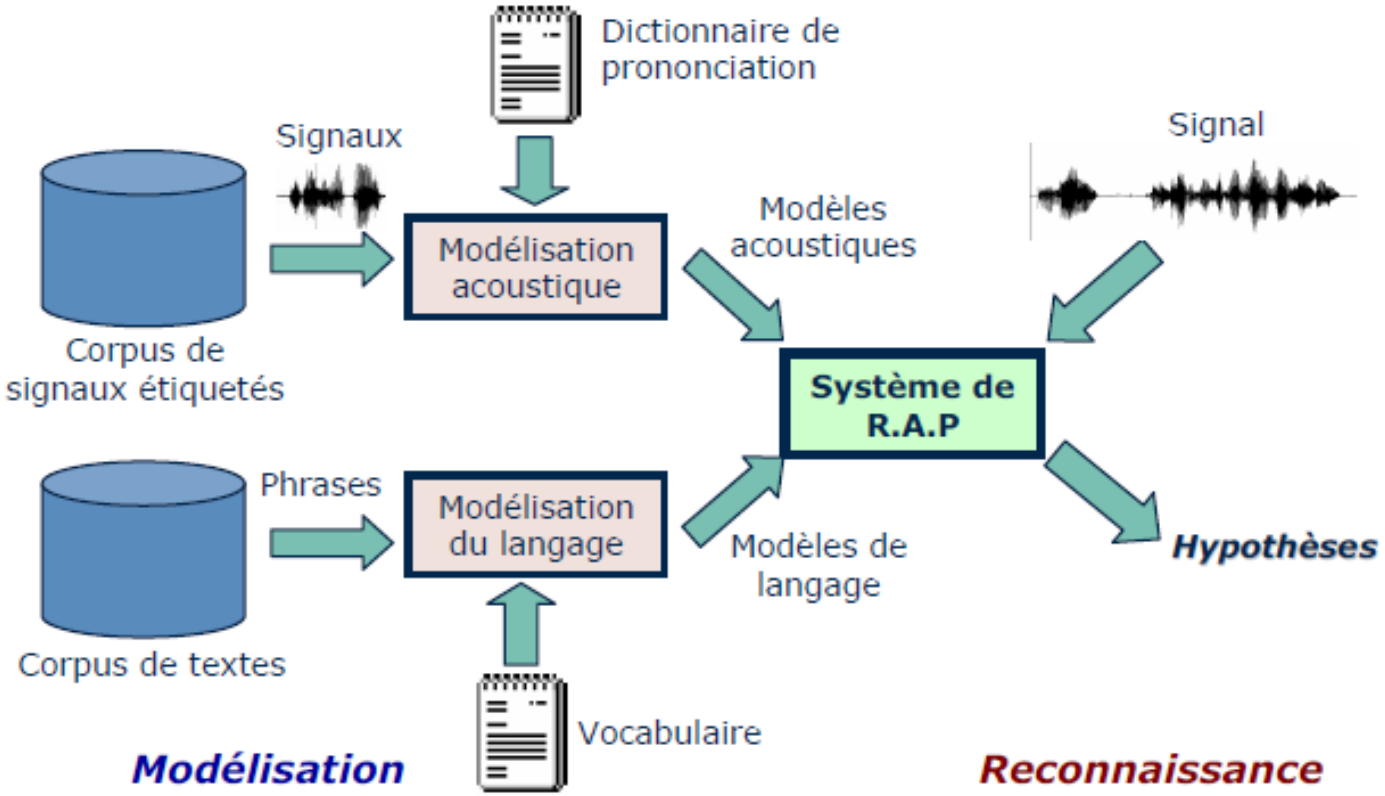
l'utilisation des modèles de langage est devenue nécessaire !

Idée avec LVCSR : dépasser le cadre statistique standard :

- accroître le volume des données d'apprentissage ;
- introduire (pré- & post-) adaptations pour améliorer la représentation probabiliste.
- résoudre ce défi en considérant 2 sous-problèmes :
  - Caractéristiques acoustiques et modèles acoustiques : concerne le traitement du signal vocal.
  - Modélisation du langage : aborde le problème de la modélisation du langage naturel.

Reconnaissance Automatique de la Parole

Un **SRAP** [1] permet de transcrire un **signal acoustique** de parole en **texte**.



1. ou dit Système de Transcription ou Système de Reconnaissance de la Parole Continue à Large Vocabulaire (LVCSR)

Formule Fondamentale du SRAP

- le signal acoustique prononcé par un utilisateur $\rightarrow$ une observation O
- la séquence de mots correspondante $\rightarrow$ W.
- Le SRAP recherche W la plus vraisemblable par rapport O en entrée.

  $\rightsquigarrow$ trouver la $\overset{\frown}{W}$ qui maximise la probabilité *à posteriori* $P(W|O)$,
  où ($L \rightarrow$ le langage considéré) :

$$\overset{\frown}{W} = \underset{W \in L}{argmax}\ P(W|O) \tag{1}$$

Formule Fondamentale du SRAP

- le signal acoustique prononcé par un utilisateur $\rightarrow$ une observation O
- la séquence de mots correspondante $\rightarrow$ W.
- Le SRAP recherche W la plus vraisemblable par rapport O en entrée.
  $\rightsquigarrow$ trouver la $\overset{\frown}{W}$ qui maximise la probabilité *à posteriori* $P(W|O)$,
  où ($L \rightarrow$ le langage considéré) :

$$\overset{\frown}{W} = \underset{W \in L}{argmax}\ P(W|O) \tag{1}$$

- L'utilisation de cette formule est difficile à cause de l'estimation de la probabilité $P(W|O)$.
  $\Rightarrow$ la difficulté réside dans la grande variabilité dans l'ensemble de départ des observations acoustiques.

Formule Fondamentale du SRAP

- Il est plus facile d'estimer la probabilité d'avoir une certaine O sachant une séquence de mots W. La formule de Bayes décompose $P(W|O)$ en :

$$\overset{\frown}{W} = \underset{W \in L}{argmax} \frac{P(O|W).P(W)}{P(O)} \tag{2}$$

- Le problème est réduit à une tâche d'optimisation par rapport à W.
- la probabilité de $P(O)$ ne dépend pas de W, ce qui induit à une eq. reliant seulement le modèle acoustique (**MA**) vraisemblant $P(O|W)$ et le modèle de langage (**ML**) à priori $P(W)$.

$$\overset{\frown}{W} = \underset{W \in L}{argmax}\ P(O|W).P(W) \tag{3}$$

Analyse & Modélisation Acoustique

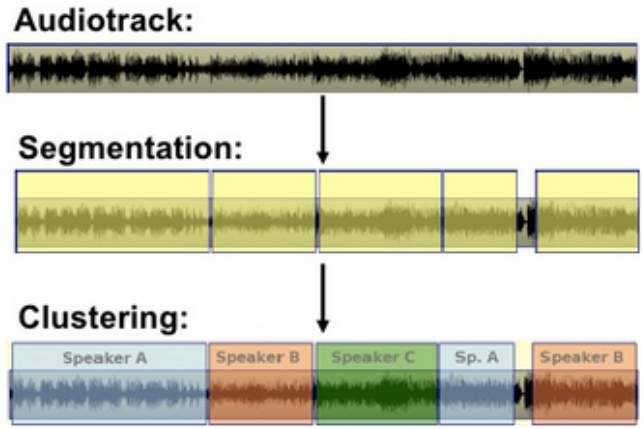
La transcription des émissions de radio et de TV à grand vocabulaire nécessite :

- La segmentation du signal audio, en segments homogènes extraits et classés en parole et en musique, est souvent a base des différences acoustiques entre les deux types de sons.
- Le signal acoustique de la parole est traité puis décomposé en bandes de fréquences (téléphone et non-téléphone).

	Spectre du signal	F_{ech}	**Applications**
Qualité téléphonique	[300-3400 Hz]	8 kHz	Téléphonie
Qualité "bande élargie"	[50-7000 Hz]	16 ou 22 kHz	PC, audio-conférence (ADPCM)
Haute qualité en radiodiffusion	[50 - 15 000 Hz]	32 kHz	DAB, NICAM
Qualité "Hi-Fi"	[20 - 20 000 Hz]	44.1 ou 48 kHz	CD Audio, Studio numérique, DAT

Diarization du locuteur

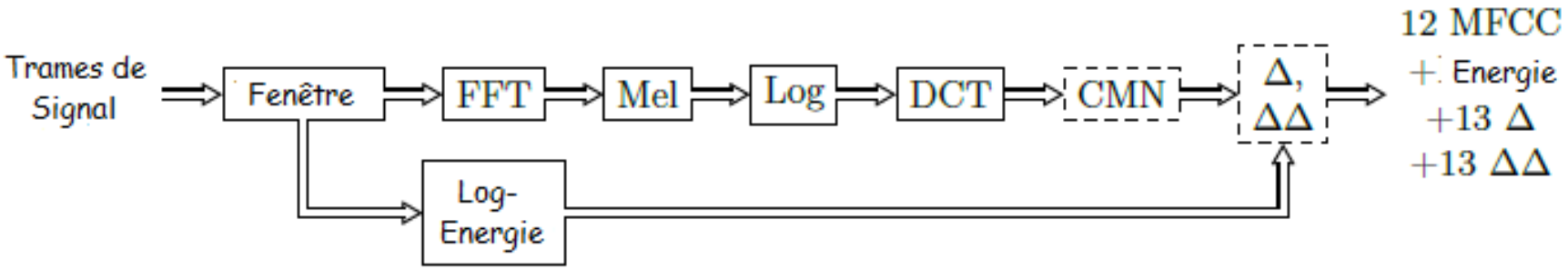
- Dans la transcription enrichie :
  - Traitement d'un flux hétérogène (parole de plusieurs locuteurs, bruit, etc.)
  - avec le recouvrement entre locuteurs (cas de plusieurs microphones).
- $\Rightarrow$: **Diarization des locuteurs**



- Les tours de locuteurs sont détectés dans les segments de parole.
- diviser le signal audio en petits segments contenant un seul locuteur.
- Un schéma de classification hiérarchique est effectué pour la fusion des segments appartenant au même locuteur.

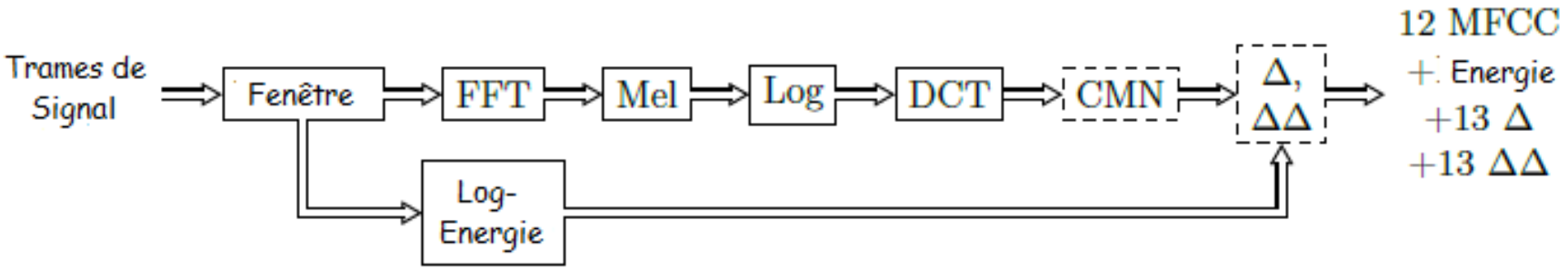
Analyse Acoustique

- Extraction des caractéristiques acoustiques **MFCC** (*Mel Frequency Cepstrum Coefficients*) du signal acoustique,
- Pour une pré-adaptation au canal et/ou le locuteur :
  Utilisation de la Normalisation de la Moyenne Cepstral (**CMN**), qui permet de réduire l'influence du canal de transmission, en normalisant la moyenne des caractéristiques cepstrales.



Analyse Acoustique

- Extraction des caractéristiques acoustiques **MFCC** (*Mel Frequency Cepstrum Coefficients*) du signal acoustique,
- Pour une pré-adaptation au canal et/ou le locuteur :
  Utilisation de la Normalisation de la Moyenne Cepstral (**CMN**), qui permet de réduire l'influence du canal de transmission, en normalisant la moyenne des caractéristiques cepstrales.



- Utilisation de la Normalisation acoustique par la Longueur du conduit Vocal (**VTNL**), en appliquant un coefficient de dilatation sur l'échelle des fréquences.

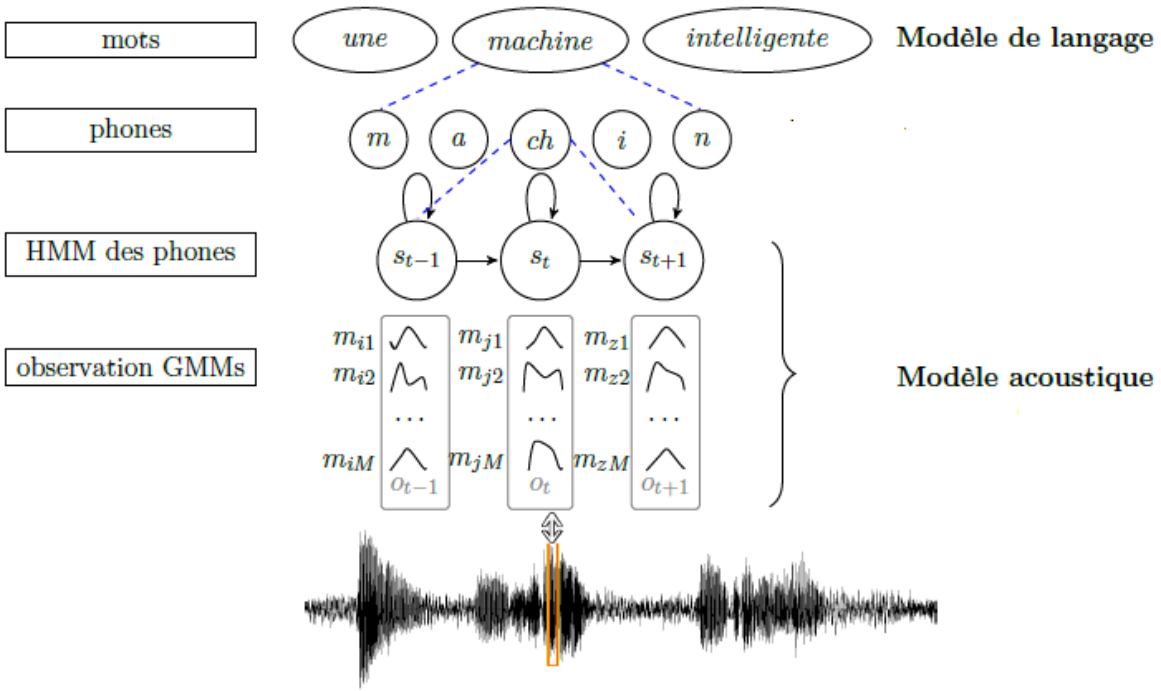
Modèles de Markov Cachés

- Un modèle de Markov Caché (HMM) est défini par (S, O, π, **A**, **B**)
  - S : est un ensemble fini d'états,
  - π : le vecteur des probabilités initiales ;
  - **A** : la matrice de probabilités de transitions entre les états $P(s_{i+1}|s_i)$, en respectant l'ordre temporel dans lequel les formes doivent être observées.
  - O : est l'espace des observations ;
  - **B** : Loi des observations pour les états.

  $\Rightarrow$ chaque état **HMM** modélise un segment de séquence sonore ;
- la loi des observations **B** : est un mélange de +ieurs gaussiennes (**GMM**).
  $\Rightarrow$ les composantes des GMM modèlisent différentes classes de locuteurs.

Apprentissage des HMM-GMM

- Segmentation :
  - Chaque état doit être associé à l'ensemble des trames qu'il émet.
  - Décodage Acoustic-phonétique selon la séquence de mots solutions et leurs variantes phonétiques possibles.
- Estimation :
  - Ajustement des paramètres GMM : la variance, la moyenne et l'amplitude de chaque gaussienne en maximisant la vraisemblance.



Adaptation acoustique

Aujourd'hui les performances en reconnaissance mono-locuteur sont élevées !

- Il est difficle de construire des MA universls ;
- une dégradation importante pour un autre locuteur, surtout du sexe opposé ;
- La collecte des données pour chaque locuteur est coûteuse !

Un système LVCSR indépendant du locuteur nécessite :
une adaptation du système multi-locuteurs au locuteur et/ou aux conditions acoustiques.

Adaptation acoustique

Une adaptation des paramètres des MA en utilisant une quantité limitée de données :

- Adaptation Maximum A Posteriori **MAP** : adaptation fondée sur le maximum de vraisemblance Bayésien.
  - L'idée revient à modifier les variances des MA (GMM-HMM).
  - pour construire des modèles génériques aux locuteurs masculins et féminins.
  - important pour la transcription d'émissions radio-TV, où il y a peu de locuteurs feminins.

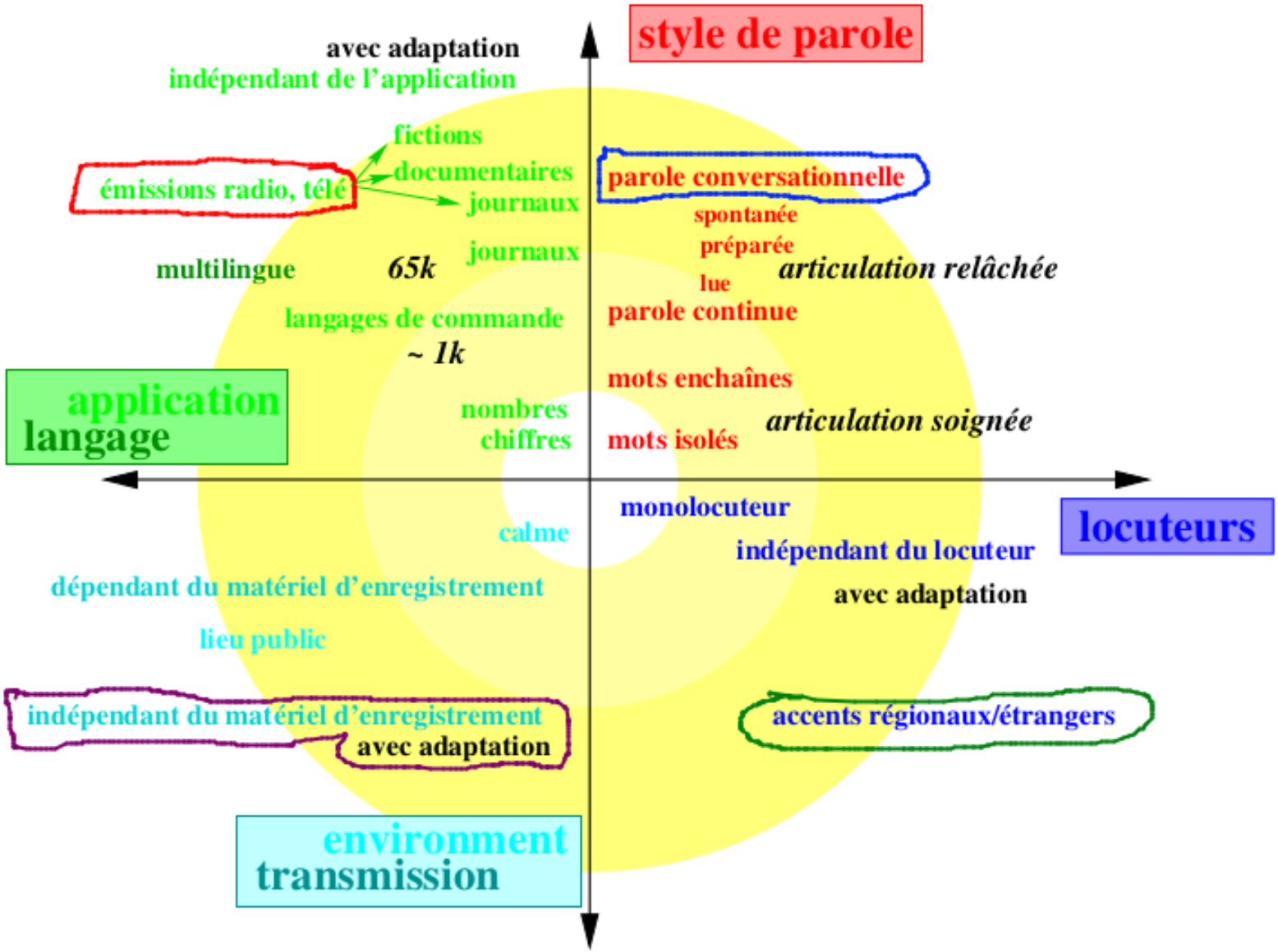
Adaptation acoustique

Une adaptation des paramètres des MA en utilisant une quantité limitée de données :

- Adaptation Maximum A Posteriori **MAP** : adaptation fondée sur le maximum de vraisemblance Bayésien.
  - L'idée revient à modifier les variances des MA (GMM-HMM).
  - pour construire des modèles génériques aux locuteurs masculins et féminins.
  - important pour la transcription d'émissions radio-TV, où il y a peu de locuteurs feminins.
- Adaptation par regression linéaire des moyennes **MLLR**.
  - utilisation des données de test pour déplacer par transformation affine les vecteurs de moyennes des GMM $\mu^* = A\mu + b$.
  - Regroupement des gaussiennes des MA en classes de regression
    $\Rightarrow$ Une trasformation MLLR apprise pour chacune des classes
    $\Rightarrow$ de façon à ce que chaque état représente mieux ces données d'adaptation.

Compagne d'évaluation - Europe

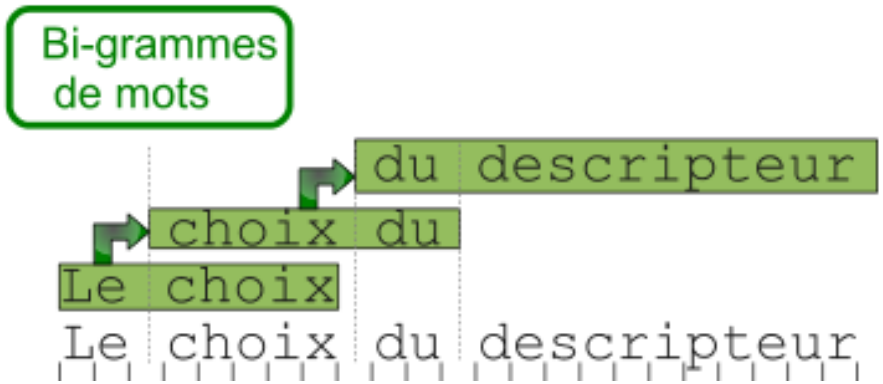
- ESTER1 (2003-2005) $\Rightarrow$ Evaluation des systèmes de Transcription enrichie d'Emissions Radiophoniques ;
- ESTER2 (2008-2009) $\Rightarrow$ + d'émissions de locuteurs avec des accents étrangers, et des émissions de parole spontanée ;
- EPAC (2010) Exploration des audio pour l'extraction et le traitement de la PArole Conversationnelle ;
- **ETAPE** (2012-2015), **É**valuations en **T**raitement **A**utomatique de la **P**arol**E**, vise à mesurer les performances des technologies vocales sur l'analyse des flux d'émissions radiophoniques et télévisés en langue française.
  - détection de la superposition parole-parole et parole-musique,
  - détection des changements de locuteurs natifs et non-natifs,
  - transcription de parole bruitée avec des contenus divers, la parole spontanée (lors de débats) et la parole proche d'un texte lu (présentation d'un journal).


style de parole
avec adaptation
indépendant de l'application
fictions
documentaires
journaux
émissions radio, télé
parole conversationnelle
spontanée
préparée
lue
journaux
multilingue
65k
articulation relâchée
langages de commande
~ 1k
parole continue
application
langage
mots enchaînes
nombres
chiffres
mots isolés
articulation soignée
monolocuteur
locuteurs
calme
indépendant du locuteur
dépendant du matériel d'enregistrement
avec adaptation
lieu public
indépendant du matériel d'enregistrement
avec adaptation
accents régionaux/étrangers
environment
transmission

Modèles de Langage

Modèle de Langage

- La modélisation du langage permet de caractériser, capturer et exploiter les régularités dans le langage naturel.
- Le modèle de Langage (ML) est un important module dans un SRAP, TA, *Natural language learning*, etc.
- ML est nécessaire pour guider vers une **bonne reconnaissance** !
- les MLs n-gramme statistiques sont utilisés dû à leur simplicité et leur robustesse.



Modèle de langage n-gramme

- L'apprentissage d'un ML nécessite une **quantité importante** de données textuelles.
- $\Rightarrow$ calculer les fréquences d'occurrences (puis les probabilités) des sous-unités de n-grammes (n-mots).

Modèle de langage n-gramme

- L'apprentissage d'un ML nécessite une **quantité importante** de données textuelles.
- $\Rightarrow$ calculer les fréquences d'occurrences (puis les probabilités) des sous-unités de n-grammes (n-mots).
- Si S est un phrase du corpus d'apprentissage, $S = w_1\ w_2\ \cdots w_k$

$$p(S) = \prod_{i=1}^{|S|} p(w_i | w_1 \cdots w_{i-1}) \tag{4}$$

Selon le **principe markovien** :

$$p(S) \simeq \prod_{i=1}^{|S|} p(w_i | w_{i-n+1} \cdots w_{i-1}) \tag{5}$$

Modèle de langage n-gramme

- L'apprentissage d'un ML n-gramme s'effectue en 2 étapes :
  1. une opération de décompte des n-grammes ; on considère le nombre d'occurrence du n-gramme $w_{i-n+1}...w_i$ dans le corpus d'apprentissage et puis on normalise par $\sum c(w_{i-n+1}...w_i)$, qui n'est rien que la fréquence d'occurrences de son historique $c(w_{i-n+1}...w_{i-1})$:

$$p(w_i \mid w_{i-n+1}, ..., w_{i-1}) = \frac{c(w_{i-n+1}...wi)}{c(w_{i-n+1}, ..., w_{i-1})} \tag{6}$$

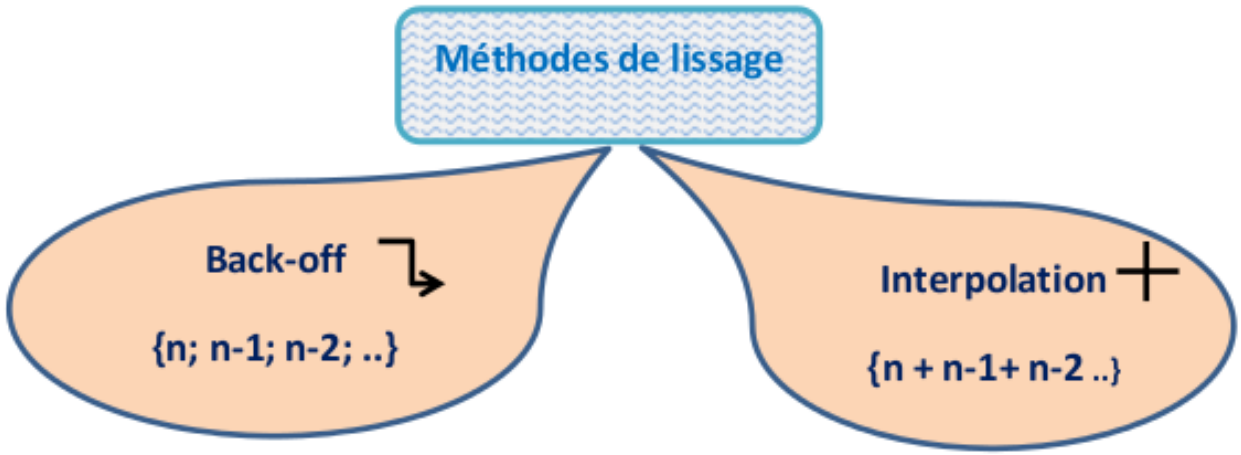
  2. La méthode d'estimation, par le maximum de vraisemblance, prend en compte toutes les suites de n mots observés dans le corpus d'apprentissage afin de calculer leurs probabilités d'apparition.

Techniques de lissage

- n $\nearrow$ peut capturer de longue dependence entre les séquences de mots.
- le nb des n-grammes augmente exponentiellement avec $n \Rightarrow$ ce qui nécessite de plus en plus de données d'apprentissage.
- $\exists$ des suites de n mots non observés dans le corpus d'apprentissage $\Rightarrow$ qui ont une probabilité $= 0$ alors qu'il se peut qu'elles soient possibles dans la langue considérée mais simplement absentes dans ce corpus.
- pour alléger la despersité des données (*Data sparsity*), des techniques de lissage sont utilisées :
  - la méthode de Good-Turing,
  - le méthode de Witten-Bell,
  - la méthode de Kenser-Ney,
  - la méthode Kneser-Ney modifiée,
  - etc.

Techniques de lissage

- Lissage par repli : on utilise la distribution du n-gramme si le nombre de ses occurences dans l'ensemble d'apprentissage est non nulle, sinon on se replie vers le niveau inférieur (backoff) et on utilise plutôt le décompte du $n-1$-gramme
- Lissage par interpolation : on utilise une interpolation entre l'information de différents niveaux des nombres d'occurrences des n-grammes. En général, on se retrouve avec une mixture des décomptes des trigrammes, bigrammes et unigrammes.



Modèle de langage n-gramme

- La mesure d'évaluation des MLs est la perplexité.
- plus la perplexité d'un ML est basse, moins il est perplexe (indécide) pour le choix des prochains mots dans une phrase.
- la perplexité : $PP(T) = b^{H_p(T)}$ où b est le bit et T les données du test.
- L'entropie croisée $H_p(T)$ du modèle considérée sur les données de test T est défini par l'équation : $H_p(T) = \frac{-1}{W_T} log_b p(T)$ tq. W_T est le taille du corpus de test par mots.

Modélisation du langage (par phrase) : exemple

```
vous savez pourquoi je vais gagner aux régionales
        p( vous | <s> )         = [2gram] 0.0205915 [ -1.68631 ]
        p( savez | vous ...)    = [3gram] 0.0656499 [ -1.18277 ]
        p( pourquoi | savez ...)        = [1gram] 0.000209739 [ -3.67832 ]
        p( je | pourquoi ...)   = [2gram] 0.00905136 [ -2.04329 ]
        p( vais | je ...)       = [2gram] 0.0312062 [ -1.50576 ]
        p( gagner | vais ...)   = [1gram] 5.24371e-05 [ -4.28036 ]
        p( aux | gagner ...)    = [1gram] 0.00145353 [ -2.83758 ]
        p( régionales | aux ...)        = [2gram] 0.000698509 [ -3.15583 ]
        p( </s> | régionales ...)       = [2gram] 0.0573104 [ -1.24177 ]
1 sentences, 8 words, 0 OOVs
0 zeroprobs, logprob= -21.612 ppl= 251.96 ppl1= 502.918

je vais gagner parce_que je suis un winner et que j' ai la classe
        p( je | <s> )   = [2gram] 0.0307388 [ -1.51231 ]
        p( vais | je ...)       = [3gram] 0.0476596 [ -1.32185 ]
        p( gagner | vais ...)   = [1gram] 3.34446e-05 [ -4.47567 ]
        p( parce_que | gagner ...)      = [2gram] 0.010629 [ -1.97351 ]
        p( je | parce_que ...)  = [2gram] 0.0471865 [ -1.32618 ]
        p( suis | je ...)       = [3gram] 0.110093 [ -0.958241 ]
        p( un | suis ...)       = [3gram] 0.0344677 [ -1.46259 ]
        p( winner | un ...)     = [1gram] 2.05544e-06 [ -5.6871 ]
        p( et | winner ...)     = [1gram] 0.0199817 [ -1.69937 ]
        p( que | et ...)        = [2gram] 0.031913 [ -1.49603 ]
        p( j' | que ...)        = [3gram] 0.0234545 [ -1.62977 ]
        p( ai | j' ...)         = [4gram] 0.737421 [ -0.132284 ]
        p( la | ai ...)         = [3gram] 0.01014 [ -1.99396 ]
        p( classe | la ...)     = [2gram] 5.2099e-05 [ -4.28317 ]
        p( </s> | classe ...)   = [2gram] 0.222207 [ -0.653242 ]
1 sentences, 14 words, 0 OOVs
0 zeroprobs, logprob= -30.6053 ppl= 109.737 ppl1= 153.494
```

Modélisation du langage (par n-gramme) : exemple

```
1
2 \data\
3 ngram 1=15174
4 ngram 2=98657
5 ngram 3=20841
6 ngram 4=11094
7
8 \1-grams:
9 -3.037791      &ah      -0.4010272
10 -3.258249     &bah     -0.3292798
11 -3.198699     &ben     -0.3262669
12 -4.86313      &bing    -0.1114681
13 -4.347228     &bon     -0.1455508
14 -4.122843     &bé      -0.1767393
15 -3.643201     &eh      -0.4757156
16 -4.710825     &ha      -0.2446922
17 -2.509209     &hein    -0.5368083
18 -2.791544     &hm      -0.3830116
19 -4.710825     &hop     -0.2446922
20 -4.86313      &hou     -0.1114681
21 -4.86313      &hè      -0.1114681
22 -3.242579     &hé      -0.5071586
23 -4.710825     &la      -0.1114681
24 -4.86313      &of      -0.1114681
25 -3.540274     &oh      -0.2298987
26 -4.86313      &ohla    -0.1114681
27 -3.146339     &ouais   -0.415956
28 -4.607618     &oulala  -0.1114681
29 -4.607618     &pff     -0.1114681
30 -4.86313      &pfft    -0.1114681
31 -4.86313      &plop    -0.1114681
32 -4.86313      &pof     -0.1114681
33 -4.258923     -ce      -0.152998
34 -4.607618     -ci      -0.1114681
35 -4.258923     -elle    -0.1114681
36 -4.710825     -elles   -0.1114681
```

. . .

```
108887 -2.379222      votre fidélité  -0.1238849
108888 -2.884256      votre film      -0.05499753
108889 -2.950804      votre foi       -0.05499753
108890 -2.9439 votre foncier   -0.05499753
108891 -2.928203      votre force     -0.05499753
108892 -1.570462      votre gouvernement      -0.05499753
108893 -2.941622      votre génération        -0.05499753
108894 -2.950804      votre honneur   -0.05499753
108895 -2.956678      votre imparfait -0.05499754
108896 -2.953131      votre interlocuteur     -0.05499752
108897 -2.122335      votre journal   -0.05499753
108898 -2.379543      votre lauréat   -0.05499753
108899 -2.366954      votre livre     -0.05499753
108900 -2.955469      votre magasin   -0.05499752
108901 -2.253602      votre mais      -0.05499753
108902 -2.374273      votre majorité  -0.05499752
108903 -2.928203      votre message   -0.05499753
108904 -2.950804      votre milieu    -0.05499753
108905 -2.941622      votre mms       -0.05499753
108906 -2.939357      votre modèle    -0.05499753
108907 -2.930411      votre monde     -0.05499753
108908 -2.934861      votre mot       -0.05499753
108909 -2.939357      votre mouvement -0.05499753
108910 -2.950804      votre mur       -0.05499753
108911 -2.376123      votre méthode   -0.05499752
108912 -2.946189      votre métier    -0.05499753
108913 -2.124752      votre nom       -0.1495101
108914 -2.953131      votre opération -0.05499752
108915 -2.937103      votre papier    -0.05499754
108916 -1.765506      votre parole    -0.1946408
108917 -2.908822      votre part      -0.05499753
108918 -2.892291      votre petit     -0.05499753
108919 -2.919482      votre petite    -0.05499753
108920 -2.955469      votre pharmacien        -0.05499753
108921 -2.939357      votre pièce     -0.05499753
108922 -2.950804      votre plateau   -0.05499753
```

. . .

Modèle de Langage Statistique

Autres modèles de langage :

- MLs n-gramme de classe de Brown(1992) ;
- MLs cache de Kuhn et De Mori (1990) ;
- MLs tigger introduits par Lau et al.
- MLs à maximum d'entropie - exponentials de Rosenfeld, 1996) ;
- MLs de sous-mots morphologiques ;

Modèle de Langage Statistique

Autres modèles de langage :

- MLs n-gramme de classe de Brown(1992) ;
- MLs cache de Kuhn et De Mori (1990) ;
- MLs tigger introduits par Lau et al.
- MLs à maximum d'entropie - exponentials de Rosenfeld, 1996) ;
- MLs de sous-mots morphologiques ;
- MLs neuronaux proposés par (Bengio, 2003).
- etc.

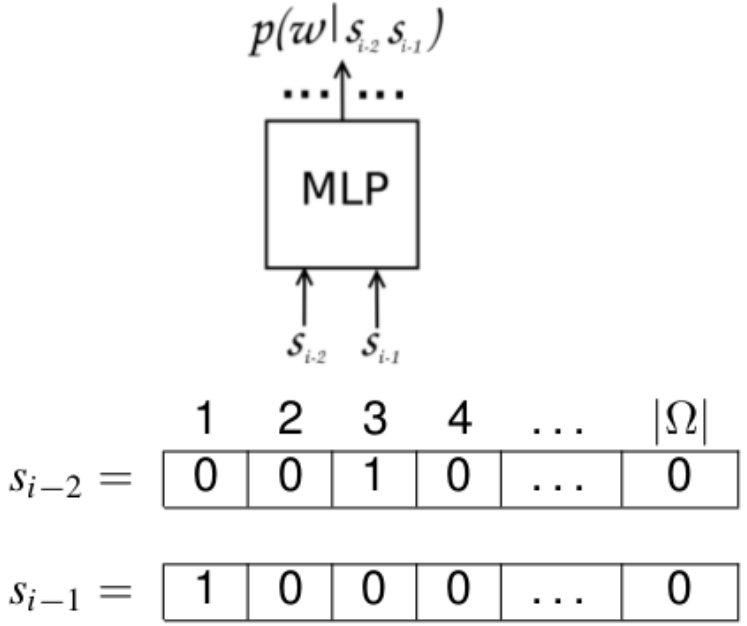
Modèles de langage neuronaux- NNLM

- Le principe de représentation continue, qui est le point fort des NNLM, était introduite par Bengio (2003) et reprise par Schwenk (2007).
- La représentation continue du mot sont les probabilités qui composent son contexte.
- Ces représentations, ainsi que les paramètres de la fonction d'estimation, sont apprises conjointement par un réseau de neurones.
- Chaque mot du vocabulaire est représenté comme un point dans un espace métrique.
- Cette stratégie d'estimation permet aux mots similaires d'avoir des représentations proches.

Modèles de langage neuronaux- NNLM

Le principe de représentation des mots par les réseaux de neurones directs (MLP) : se caractérisent par l'estimation alternative des probabilités avec une représentation continue.

- Les entrées : les mots du Vocabulaire $w_{i-n+1}, \cdots, w_{i-1}$, utilisés pour les ML n-grammes $p(w_i|w_{i-n+1} \cdots w_{i-1})$, représentés par un vecteur *one-hot*.



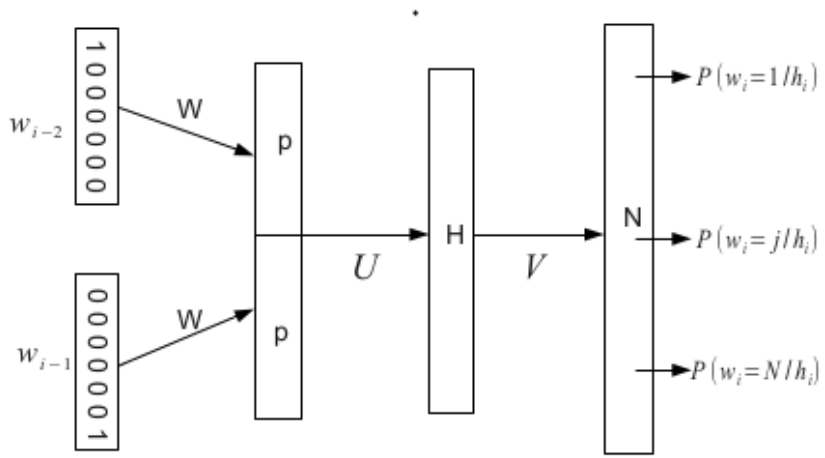
Modèles de Langage Neuronaux- CSLM

- La couche de projection **P** permet d'avoir une table de taille $|V|$ qui sert à une représentation distribuée des mots du vocabulaire.
- Une représentation distribuée d'un mot est un vecteur de valeurs réelles, de taille fixe, elle est stockée dans une table de Contexte "C".

Mot(w)	Représentation distribuée
"le"	[0.67 -0.96 0.36 -0.24]
"la"	[0.68 -0.92 0.37 -0.21]
"hall"	[0.58 0.91 0.04 0.76]
"salle"	[0.59 0.91 0.08 0.77]
"être"	[0.16 -0.15 0.03 -0.33]
"avoir"	[0.17 -0.13 0.07 -0.29]
...	...

Modèles de Langage Neuronaux- CSLM

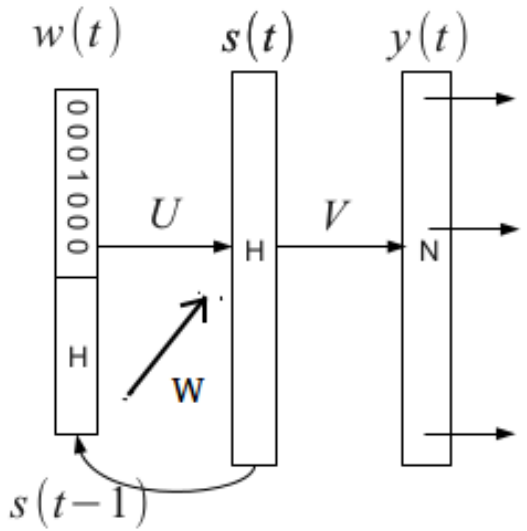
- **H** est la couche cachée avec un nombre empirique de neurones (simple / large).
- **N** est la couche de sortie avec $|V|$ neurones. La fonction d'activation softmax assure que la $\sum$ valeurs de sortie $= 1$.



- Le MLP prédit les probabilités *a posteriori* de chaque mot du vocabulaire sachant son historique.
- L'apprentissage du MLP par l'algorithme de rétro-propagation (*Back-Propagation* - **BP**).

Modèles de Langage Neuronaux- RNNLM

- Les MLs basés sur le RNN dites **RNNLM** sont introduits par Mikolov (2010).
- Les neurones cachés du RNN reçoivent des valeurs d'entrée à la fois des neurones d'entrée et des neurones cachés.
- L'apprentissage par l'algorithme de réto-propagation à travers le temps (*Back-Propagation Through Time* - **BPTT**) (werbos,1990).



Expérimentation : ML n-gramme & NNLM

Expérimentation-ML

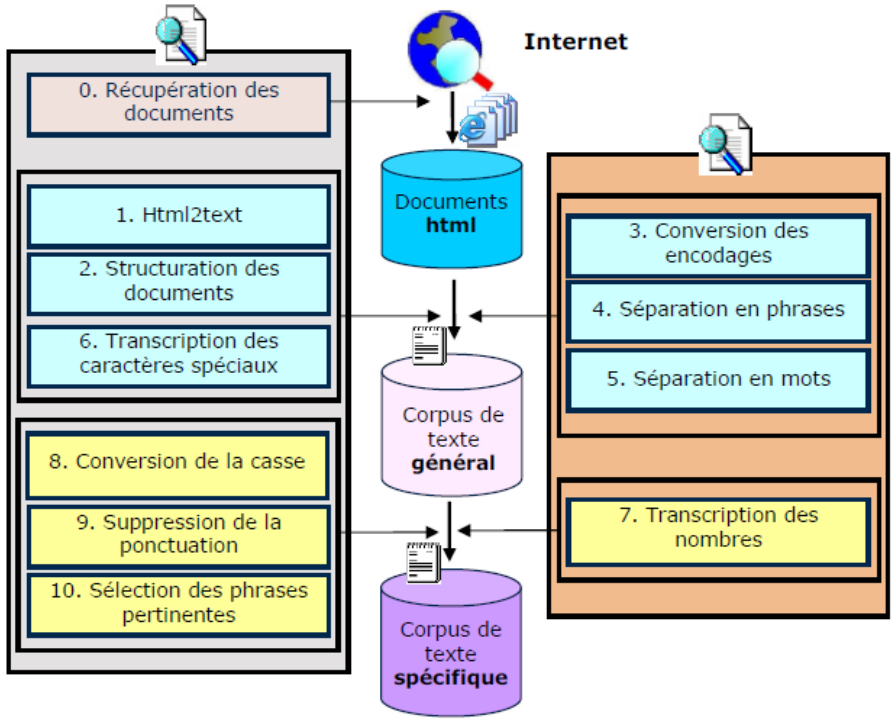
Outils logiciels utilisés :

- Machine- Processeur : Intel Core i5-2 CPU 2.40GHz $*$ 4 ; RAM : 6 Go.
- SE :Ubuntu 14.04 LTS.
- ML n-gramme $\Rightarrow$ **SRILM** de Andreas (2002, 2011). Paquetage est développé, maintenu et distribué sous une libre licence par SRI de Californie.
- MLs spatial continus NNLM $\Rightarrow$ la boite à outils libre **CSLM**. Il repose sur des bibliothèques math. tq. BLAS, le support possible des GPU et la liste courte.
- MLs récurrents RNNLM $\Rightarrow$ la boîte à outils libre de Mikolv **RNNLM** écrit en $C/C++$, simple à installer et à adapter.
- le langage **SHELL** : le mini-langage de programmation intégré à Linux.

Corpus textuel

Un corpus textuel est une large collection de textes, de langage naturel, structurés et enregistré électroniquement gratuit/ payant.

- transformation des pages html vers du texte ;
- passage au Unicode Utf-8 ;
- séparation en phrases puis en mots ;
- transcription des caractères spéciaux et des nombres ;
- conversion des majuscules ;
- suppression de la ponctuation ;
- etc.



Expérimentation-ML

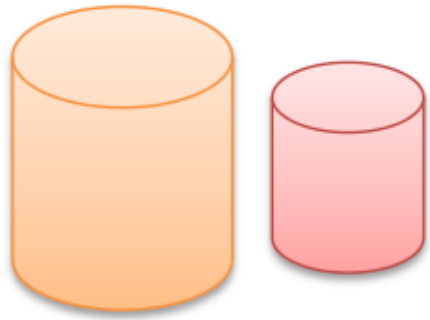
Corpora textuels utilisés :

- Data 1 - **ETAPE** (Compagne éval. LVCSR) :

Corpus	# phrase	# mots
Etape-train	18 083	249 569
Etape-dev.	1 004	10 782
Etape-test	1 004	16 419

- Data 2 - **NAACL** (Workshop TA 2012) :

Corpus	*# phrases*	*# mots*
nc7-train	212 517	5 085 447
eparl7-train	2 218 201	59 940 634
newstest2010-dev.	2 489	61 924
newstest2011-test	3 003	74 833



Expérimentation-MLs utilisés ETAPE

- LM N-grammes : 4-grammes ; $\neq$ Méthodes de Lissage.
- CSLM sont entraînés pendant 10 itérations.
  - CSLM simples : La couche d'entrée est de 15 662 neurones (la taille de vocabulaire) ; la couche de projection est de 256 neurones pour chaque mot, suivie d'une couche cachée *tanh* de (768 x 192) et d'une couche de sortie *softmax* de 1 024 neurones (la taille de la liste courte).
  - CSLM larges : La couche de projection est de 256 neurones, la couche cachée *tanh* est de (768 x 192) et d'une couche de sortie *softmax* de 8192 neurones.
  - CSLM profonds : La couche de projection est de 256, trois couches cachées *tanh* de (768 x 512), (512 x 256) et (256 x 192) respec. une couche de sortie *softmax* de 8192 neurones.
- RNNLM : La couche cachée de 100 neurones, 100 classes pour accélérer l'apprentissage. L'algorithme BPTT : mode mini-lots avec une taille de 10 bloc pour 4 étapes.

Expérimentation-ML n-gramme

Perplexité des MLs avec différents lissages sur les données de ETAPE.

Méthode	λ, δ	*ppl_dev*	*ppl_test*
Good-Turing_nonParamètre		380,3	387,7
Good-Turing_standard		253,3	252,9
Backoff-décompte absolu	0, 8	242,8	242,5
Interp-décompte absolu	0, 9	294,6	298,8
Backoff-Witten-Bell		237,6	238,4
Interp-Witten-Bell		334,1	337,7
Ristad décompte naturel		294,2	293,9
Backoff-nonModifié-KN		219,5	**220,1**
Interp-nonModifié-KN		255,6	258,4
Backoff-Modifié-KN		221,2	**221,3**
Interp-Modifié-KN		242,5	244,6

Expérimentation-ML n-gramme

Perpléxité des MLs avec différents lissages sur les données NAACL

Méthode	λ, δ	*ppl_dev*	*ppl_test*
GoodTuring_nonParamtère		487,4	533,6
GoodTuring_standard		413,5	453,9
Backoff-décompte absolu	0,7	406,3	443,6
Interp-décompte absolu	0,7	454,9	502,7
Backoff- Witten-Bell		387,5	421,4
Interp-Witten-Bell		458,4	510,5
Ristad NaturelDiscount		492,6	542,6
Backoff-nonmodified-KN		338,2	367,3
Interpo.Unmodified-KN		356,4	392,3
Backoff Modified-KN		338,2	**365,8**
Interpo.Modified-KN		343,8	376,4

Expérimentation-ML

Résultats :

- Data 1 - **ETAPE** :

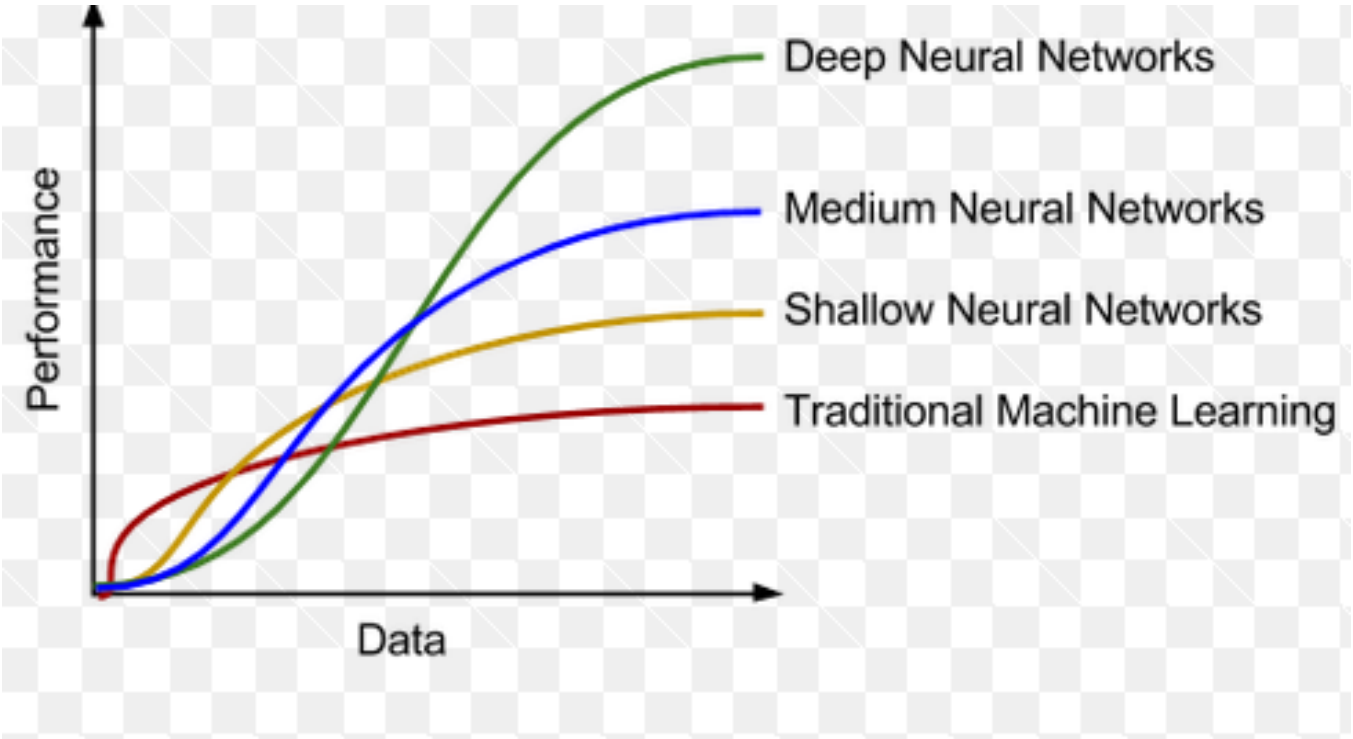
ML	*ppl_dev*	*ppl_test*	*Temps[min]*
Backoff Modified-KN ML	221.2	221.3	15
Simple CSLM	278.2	295.6	30
Large CSLM	432.6	455.3	75
Deep CSLM	253.3	272.7	49
RNNLM	159.4	**153.6**	101

- Data 2 - **NAACL** :

ML	*ppl_dev*	*ppl_test*	*Temps[h]*
Backoff Modified-KN ML	338,2	365,8	$\simeq$ 1
Simple CSLM	327,9	356,4	49
Large CSLM	331,8	359,5	90
Deep CSLM	334,4	362,5	72
RNNLM	248,8	**264,9**	133

ML n-gramme vs. NNLM

Selon la literature :

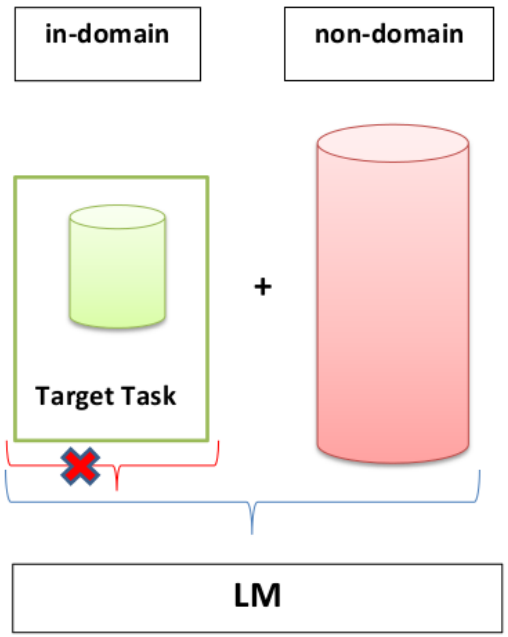


Sélection des données textuelles

Principe d'Apprentissage des MLs

L'apprentissage d'un ML nécessite une quantité importante de données textuelles.

- Un ML à bonne performance est entrainé sur :
  1. petit corpus proche à la tâche cible (in-domain)
  2. un grand corpus général non-proche à cette tâche (non-domain).
- Le corpus non-domain peut être bruité dû à la différence en qualités des sources des données.



La Sélection des données textuelles pertinentes proche à la tâche cible.

Sélection des données textuelles

- **Klakow (2000)** utilise le critère log-likelihood pour sélectionner les articles de presse.
- **Wang et al.(2002)** sélectionnent des unités (syllabes) du corpus non-domain à faible perplexitie selon ML in-domain.
- **Moore et al. (2010)** sélectionent des phrases du corpus non-domain avec de faible difference en entropie croisée entre 2 MLs de tailles similaires, représentant les données in-domain et non-domain resp.
- **Wong (2016)**, RNN pour apprendre la présentation continue avec une procédure interne de sélection de données.

Critère dXent- 4 sources

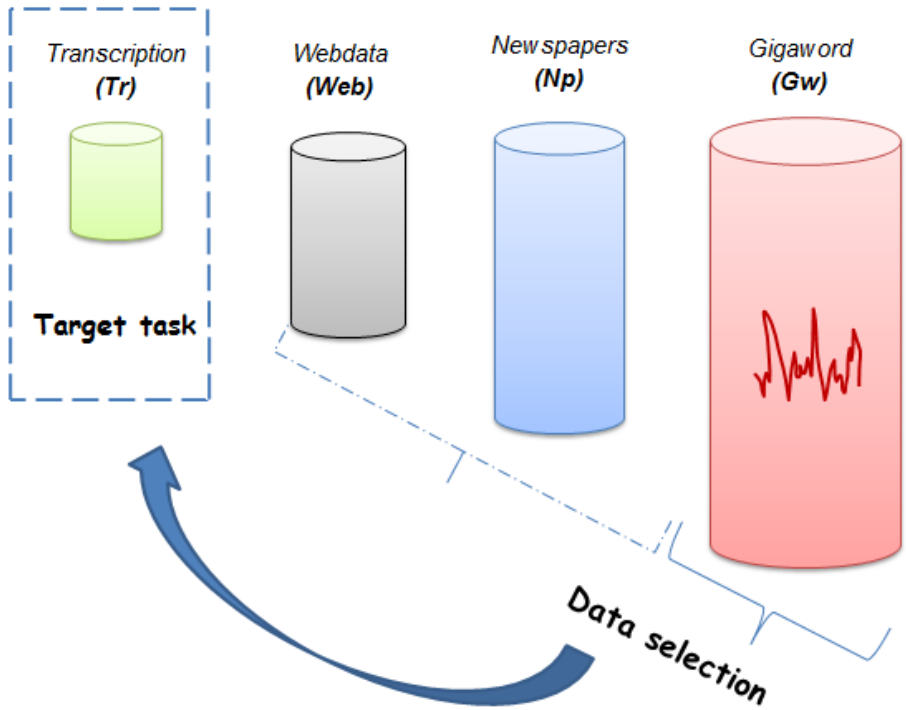
Sélection de données textuelles

Outils logiciels utilisés :

- Machine (Elephant de loria)-**Processeur** :intel Xeon R, CPU ES 1620 V4 3.50Ghz $*$ 18 ; **RAM** :100 Go ; **Carte graphique** : GcForce GTX 1060 6Go-PCIe SSE2.
- SE :Ubuntu 16.04 LTS.
- ML n-gramme $\Rightarrow$ **SRILM**.
  - Les MLs utilisées sont des 3-grammes appris par le toolkit **SRILM** ;
  - Lissage : la technique de lissage Kneser-Ney Modifié ;
- Le langage de prog. **PERL** est un langage de programmation de Larry Wall (1987), pour traiter des données textuelles.
- le langage **Shell** de Linux.

Nouvelle situation

- Transcriptions manuelles des Bulletins d'info. Radio & émissions TV ;
- Données du Web (*Webdata*) (pris des sites Web : Magazines, TV) ;
- Journaux (*Newspapers*) (Le Monde, L'Humanité) ;
- Le Corpus Gigaword 2^{nd} edition de LDC (*Linguistic Data Consortium*).



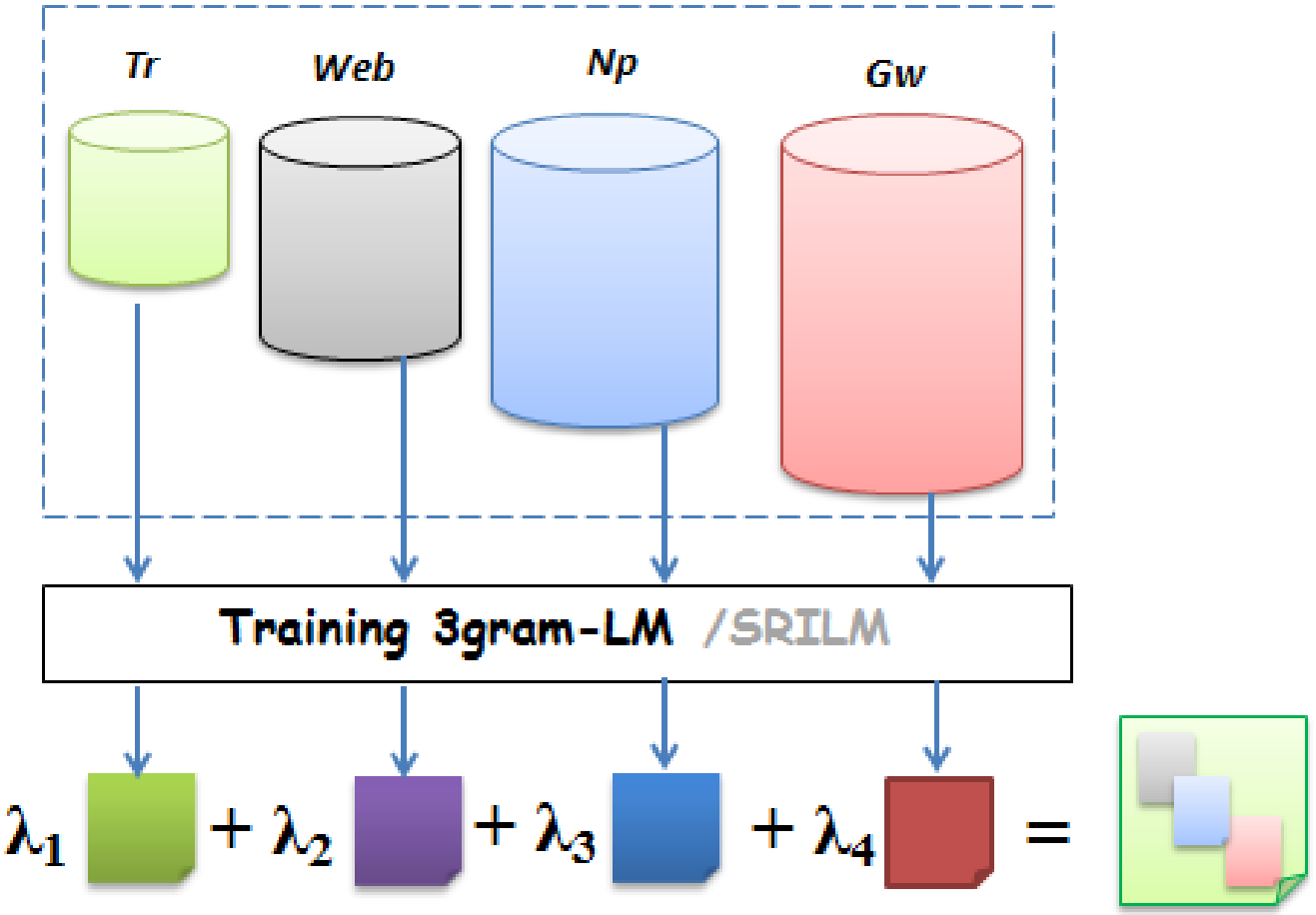
Experimentations - SD

- **Corpora d'apprentissage**

Sources	# mots
Tr (Transcriptions des emissions-radio)	113 986 727
Web (Web data)	334 057 000
Np (NewsPapers)	526 450 228
Gw (Gigaword corpus)	783 380 463
$Tr + Web + Np + Gw$	1 757 874 418

- **Corpus de Validation**, *DevLM* : 276 770 mots
- **Corpus de Test**, *TestLM* : 85 191 mots
- **Vocabulaire** : 97 349 mots

Experimentation / ML de base



$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1$$

Experimentation - ML de base

Table : **LM de base**, interpolé depuis les MLs individuels.

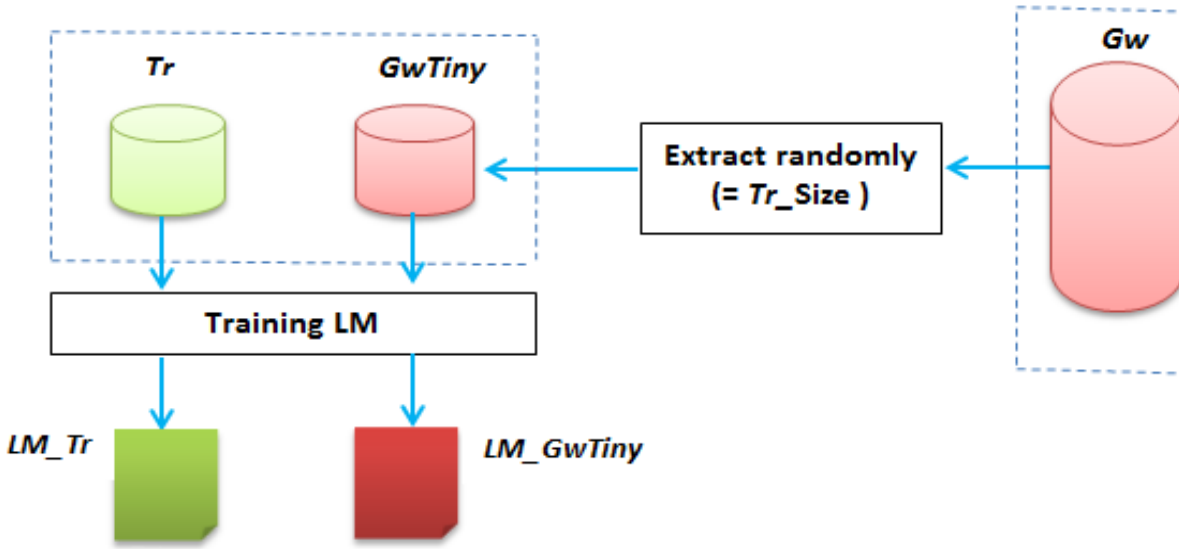
Sources	ML interpolé		
	poids	ppl *DevLM*	ppl *TestLM*
Tr	0.685	185.7	**218.9**
Web	0.246		
Np	0.062		
GW	0.007		

- Une grande différence des poids pour les MLs individuels ;
- *Gw* apporte une faible contribution pour le ML final.

Experimentations - SD

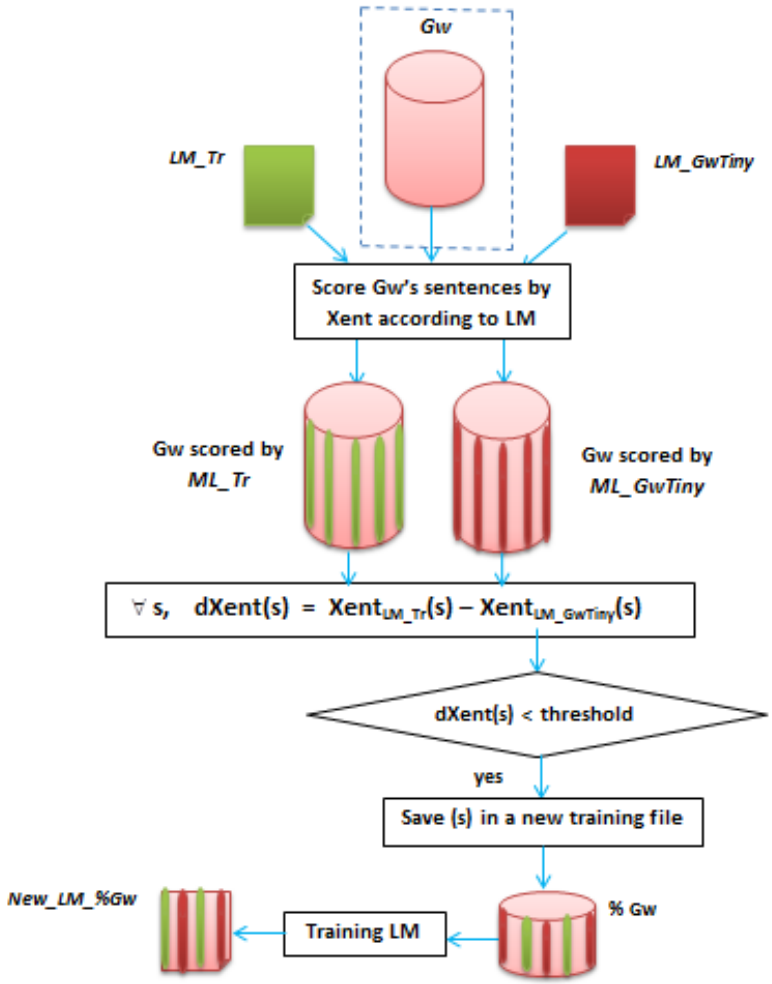
Etape 1

- # source utilisée : 2 ;
- selection des données sur : ***Gw*** ;
- "in-domain" : *LM_Tr* ;
- "non-domain" : *LM_GwTiny*.



Experimentations - SD

Etape 2



Experimentations - SD

- La ppl du ML obtenue en utilisant le corpus *Gw* complet est 671.4 ;
- De petits sous-ensembles sélectionnées **aléatoirement** du corpus *Gw* dégradent la ppl ;
- La sélection à base de la difference de l'entropie croisée (**dXent)** sur le corpus Gw améliore la ppl.

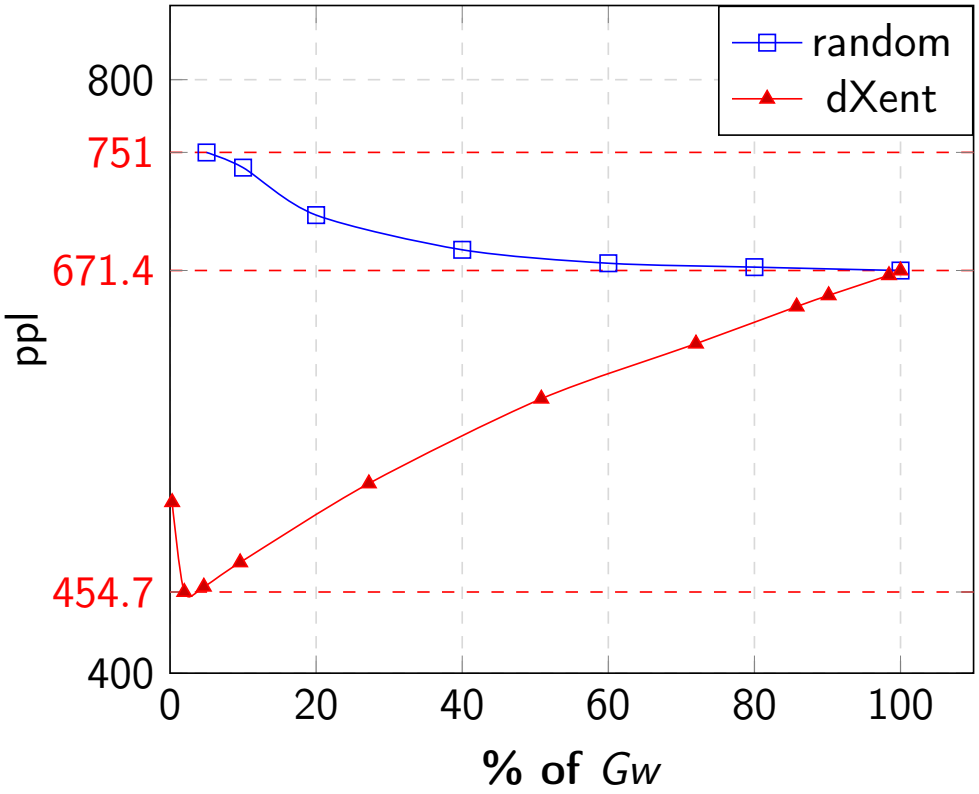
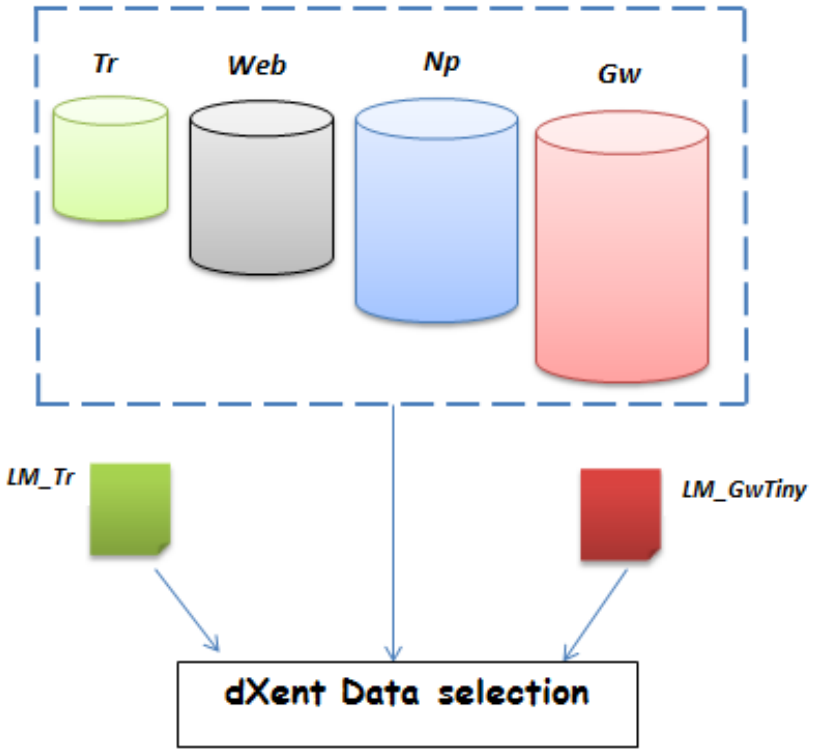


Figure : La selection est appliquée sur Gw : **(1)** selection Aléatoire (random) et **(2)** avec le critère *dXent* évalué sur *LM_Tr* et *LM_GwTiny* (dXent).

Multisource - SD / approche 1

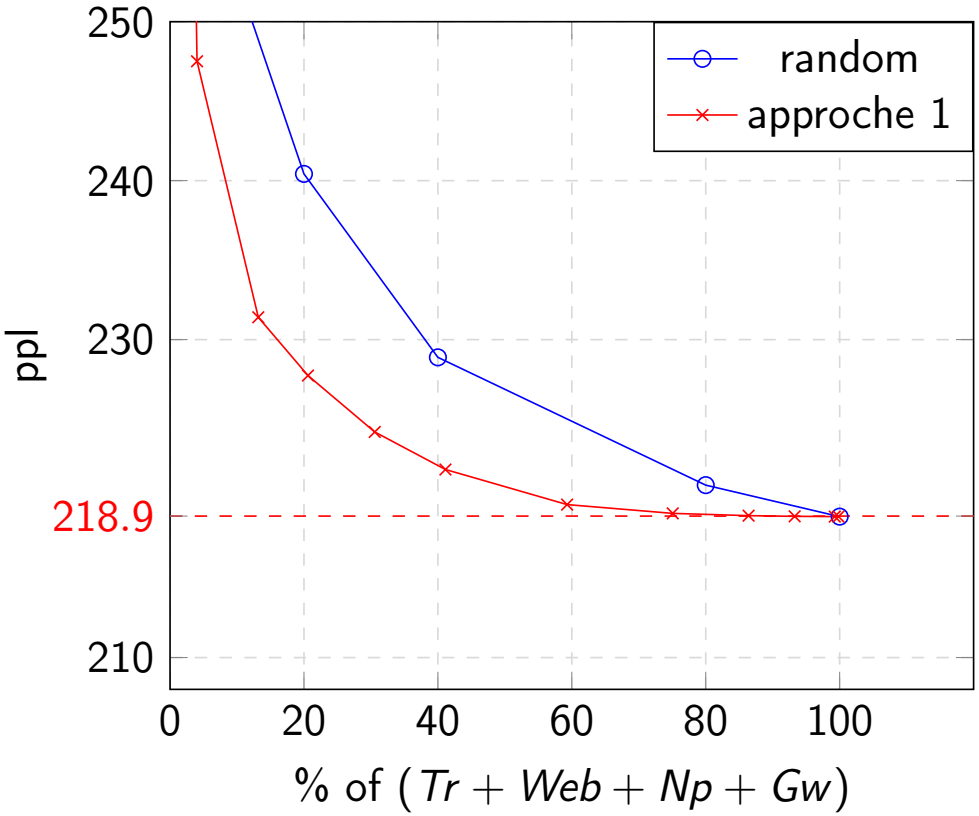
approche 1

- # source utilisée : 4 ;
- Données selectionnées : ***(Tr+ Web+ Np+ GW)*** ;
- "in-domain" : *LM_Tr* ;
- "non-domain" : *LM_GwTiny* ;



Multisource - SD / approche 1

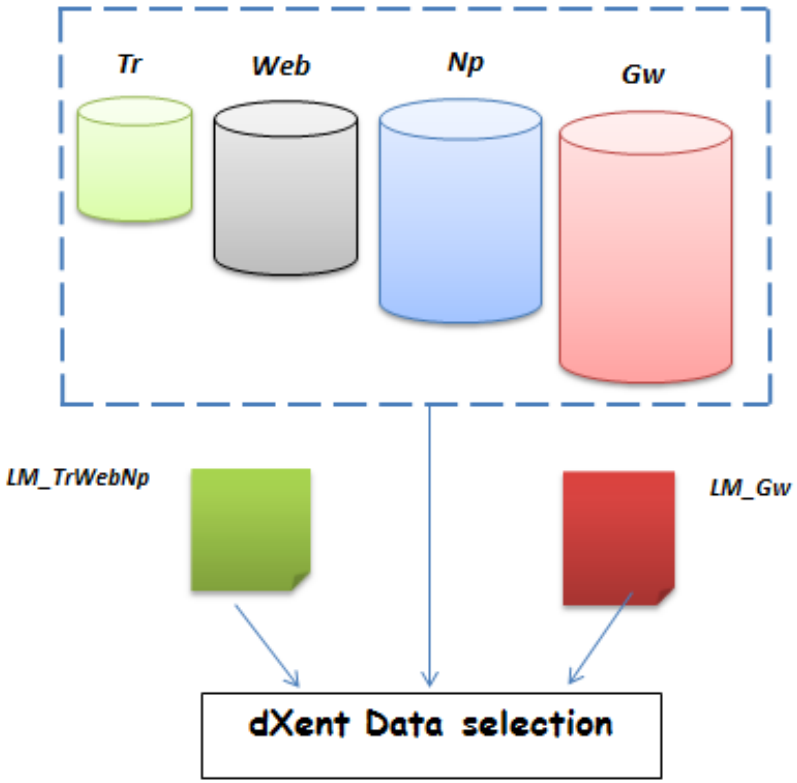
- La ppl du ML obtenue en utilisant les 4 corpora est 218.9
- La sélection **aléatoire** sur les données dégrade la ppl ;
- La sélection à base de ***dXent*** appliquée sur les données est calculée entre $LM\_(Tr)$ et LM_GwTiny (approche 1) n'améliore pas la ppl.



Multisource - SD / approche 2

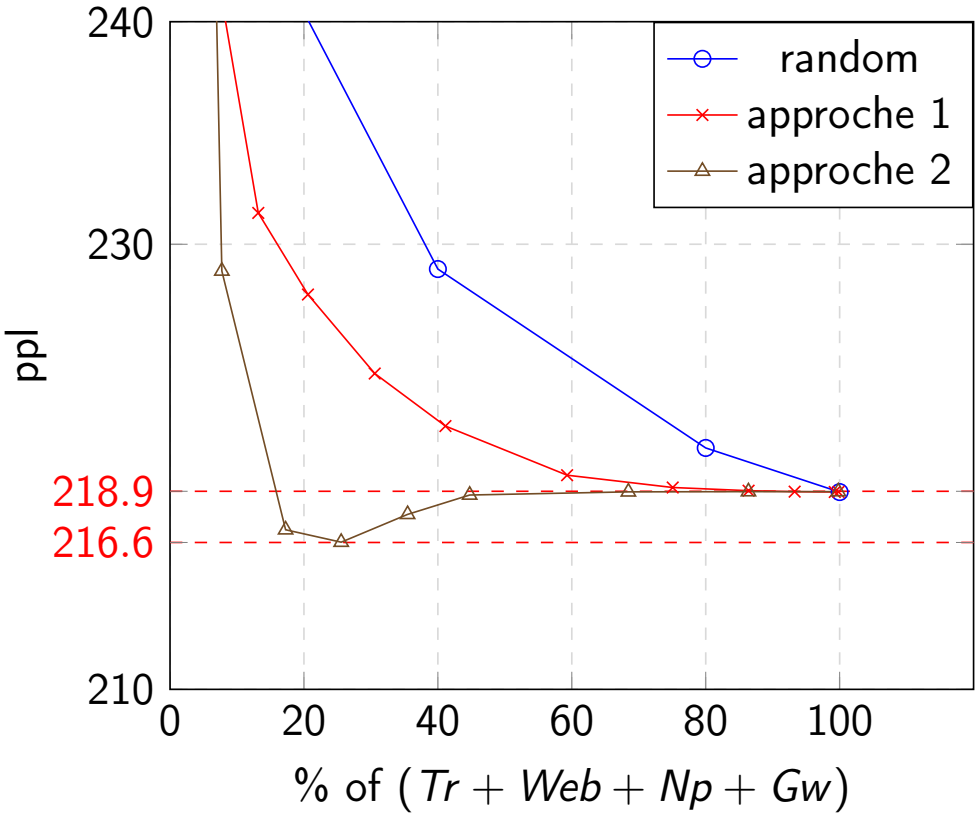
approche 2

- # source utilisée : 4 ;
- Données sélectionée :
  (***Tr+ Web+ Np+ GW***) ;
- "in-domain" : *LM_TrWebNp* ;
- "non-domain" : *LM_Gw*.



Multisource - SD / approche 2

- La selection (approche 2) est appliquée sur les données (Tr, Web, Np, Gw), par la ***dXent*** [calculée entre $LM\_(TrWebNp)$ et LM_Gw] améliore la ppl.



Multisource - SD / approche 2

Table : Le meilleur ML, interpolé par des MLs individuels appris sur les différentes sources de données, avec les données sélectionnées par l'**approche 2**.

Sources	ML interpolé		
	poids	ppl *DevLM*	ppl *TestLM*
Tr (88%)	0.608	185.1	**216.6**
Web (62%)	0.234		
Np (26%)	0.062		
Gw (0.2%)	**0.096**		

Multisource - SD/ approche 3

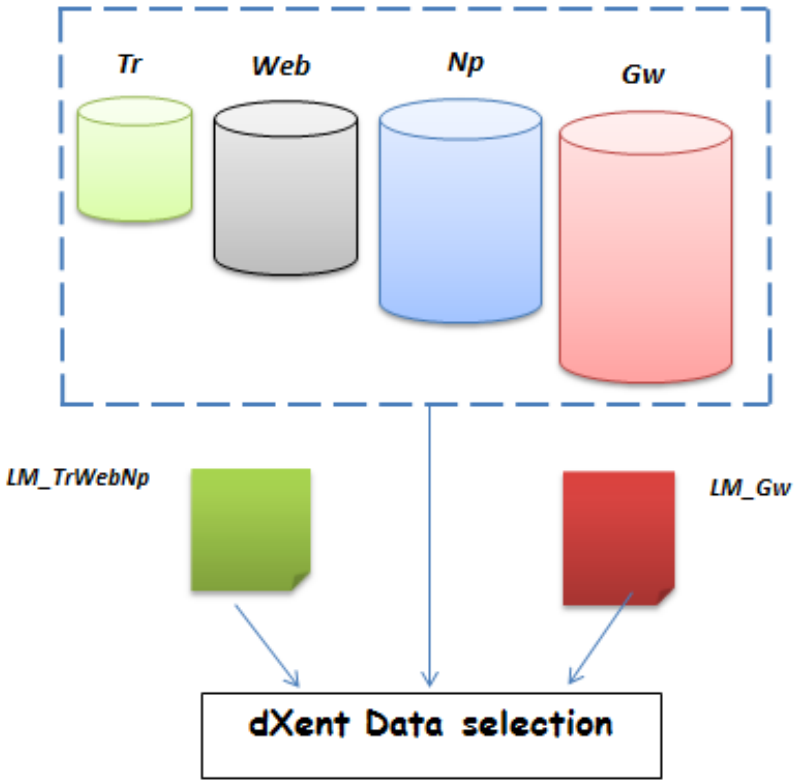
approche 3

- # sources utilisées : 4 ;
- Données sélectionnées : ***GW*** ;
- "in-domain" : *LM_TrWebNp* ;
- "non-domain" : *LM_Gw*.

vs approche 2

approche 2

- # sources utilisées : 4 ;
- Données selectionées :
  (***Tr+ Web+ Np+ GW***) ;
- "in-domain" : *LM_TrWebNp* ;
- "non-domain" : *LM_Gw*.



Multisource - SD / approche 3

- La sélection est appliquée sur le corpus *Gw* [avec la ***dXent*** calculée entre *LM_(TrWebNp)* et *LM_Gw*] et les corpora *Tr*, *Web* et *Np* (approche 3) améliore la ppl.

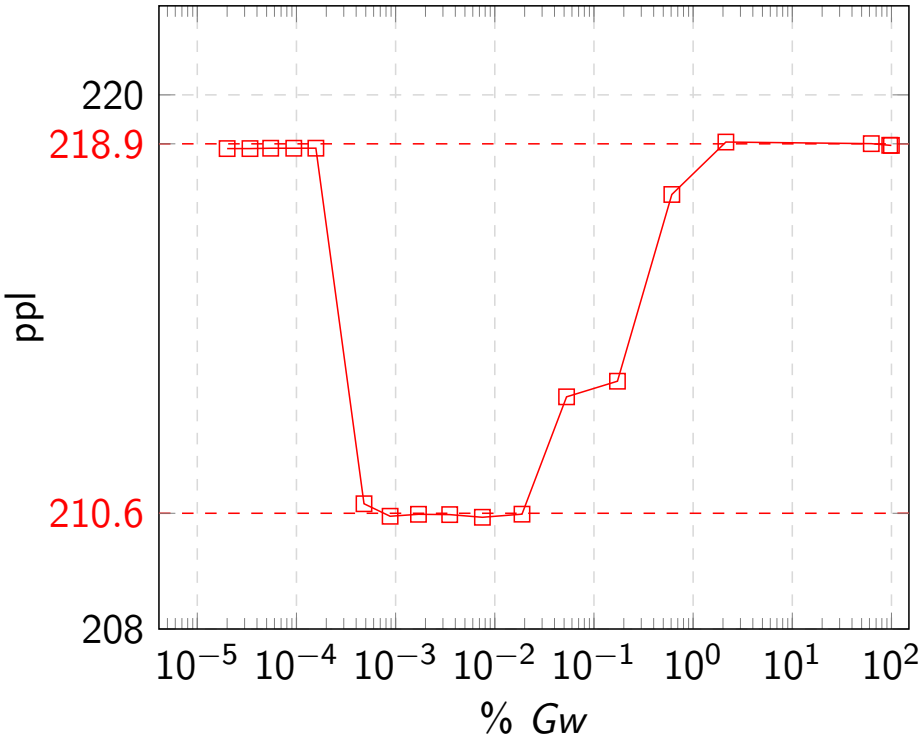


Figure : La sélection est appliquée sur Gw par *dXent* sur *LM_(TrWebNp)* et *LM_Gw* (approach 3) + les sources (Tr, Web, Np).

Multisource-SD / approche 3

Table : Le meuilleur ML appris sur les données sélectionnées avec l'**approche 3**.

Sources	ML interpolé		
	poids	ppl *DevLM*	ppl *TestLM*
Tr(100%)	0.660	179.9	**210.6**
Web(100%)	0.240		
Np(100%)	0.065		
Gw(0.05%)	**0.054**		

Experimentation des Transcriptions

- Les corpora audio $\subseteq$ ESTER2, ETAPE et le projet EPAC.
- Des paramètres 39 HTK MFCC sont utilisés (+ **CMN** + 1^{st} & 2^{nd} derivations + **VTLN**).
- Le système de transcription utilise la diarization du locuteur ;
- les adaptations **MAP**, **MLLR**, etc.
- Phonetisation du vocabulaire (BDLEX, In-house, Graheme2phonéme).

Expérimentations des Transcriptions

ML	Taille (gz file)	Corpus d'Etape Dev	
		ppl	WER[%]
$(Tr + Web + Np + Gw)$	1.2 Gb	218.9	27.84
$(Tr + Web + Np)$	809.8 Mb	218.9	27.82
ML(app.2, seuil. -0.3)	391.3 Mb	217.2	28.07
ML(app.2, seuil. -0.2)	501.6 Mb	216.6	27.89
ML(app.3, seuil. -0.6)	809.3 Mb	**210.6**	**27.68**

- Le meuilleur ML est entrainé avec 55.4% du (Tr,Web,Np,Gw).

Critère MSDP sur 2 sources

Sélection de données textuelles

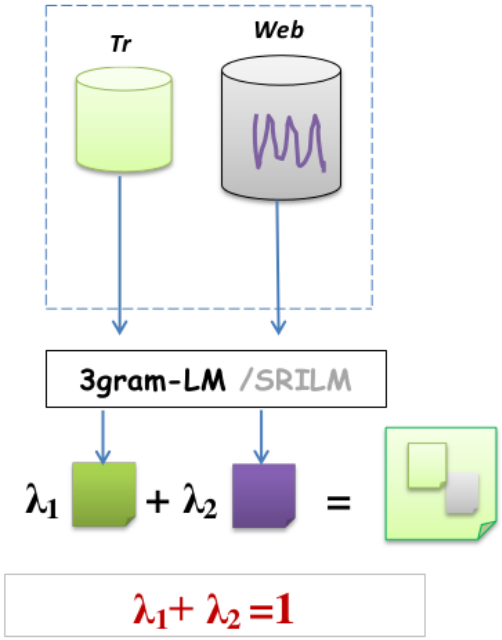
Outils logiciels utilisés :

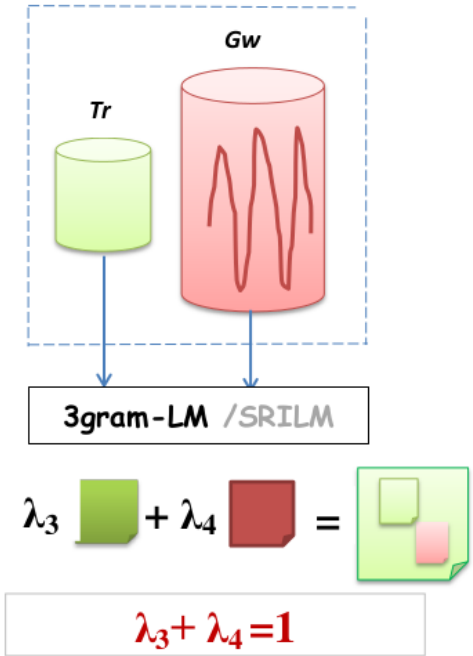
- Machine- Processeur : Intel Core i5-2 CPU 2.40GHz $*$ 4 ; RAM : 6 Go ; ;
- SE :Ubuntu 14.04 LTS.
- ML n-gramme $\Rightarrow$ **SRILM**.
- Les MLs utilisées sont des 3-grammes appris par le toolkit **SRILM** ;
- Lissage : la technique de lissage Kneser-Ney Modifié ;
- Le langage de prog. **PERL** ;
- le langage **SHELL** de Linux.

Séléction des données textuelles

Corpus d'apprentissage : énorme corpus textuel (2 sources de données) ;

- Pour chaque source de données ⇒ un ML est appris ;
- Par interpolation linéaire ⇒ MLs de base.





Sélection des données textuelles

But : la sélection des données sur un corpus de textes Français pour améliorer les MLs de transcription des bulletins d'info.radiophoniques.

1. Sélection aléatoire (**Random**) : du corpus non-domain.

Sélection des données textuelles

But : la sélection des données sur un corpus de textes Français pour améliorer les MLs de transcription des bulletins d'info.radiophoniques.

1. Sélection aléatoire (**Random**) : du corpus non-domain.
2. Sélection par différence d'entropie croisée (**dXent**) : du corpus non-domain, où chaque phrase est scorée par le dXent par le $ML_{in-domain}$ et le $ML_{tiny-nondomain}$

Sélection des données textuelles

But : la sélection des données sur un corpus de textes Français pour améliorer les MLs de transcription des bulletins d'info.radiophoniques.

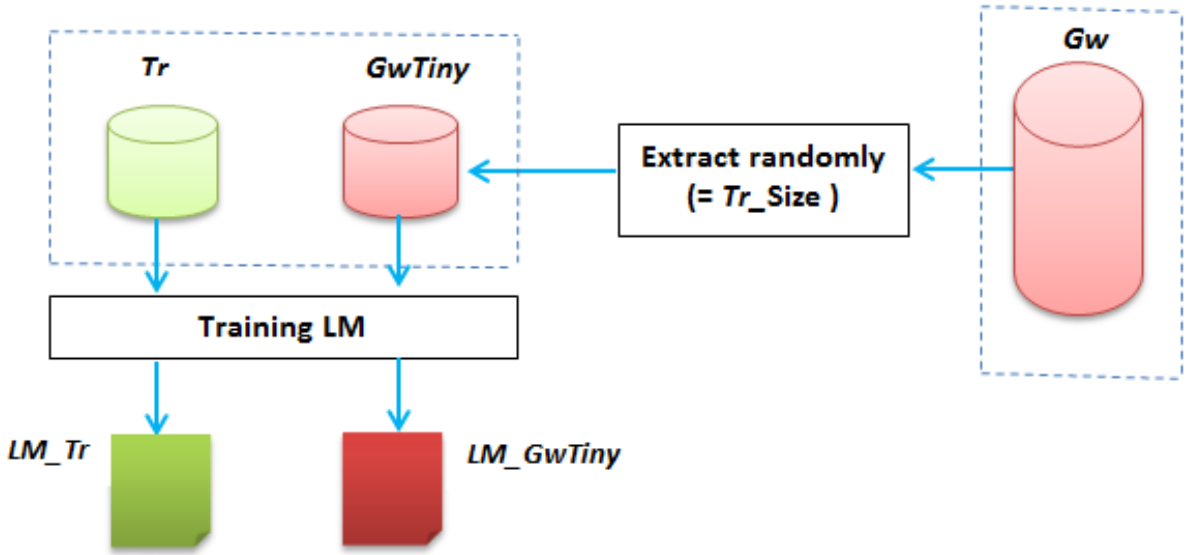
1. Sélection aléatoire (**Random**) : du corpus non-domain.
2. Sélection par différence d'entropie croisée (**dXent**) : du corpus non-domain, où chaque phrase est scorée par le dXent par le $ML_{in-domain}$ et le $ML_{tiny-nondomain}$
3. Sélection par différence quadratique de Probabilités (**MSDP**) : du corpus non-domain, où chaque phrase est scorée par le MSDP par le $ML_{in-domain}$ et le $ML_{tiny-non-domain}$.

Sélection Aléatoire des données textuelles

- Sélection aléatoire (**Random**) : sur le corpus non-domain Gw avec des portions 10%, 20%, 40% etc.

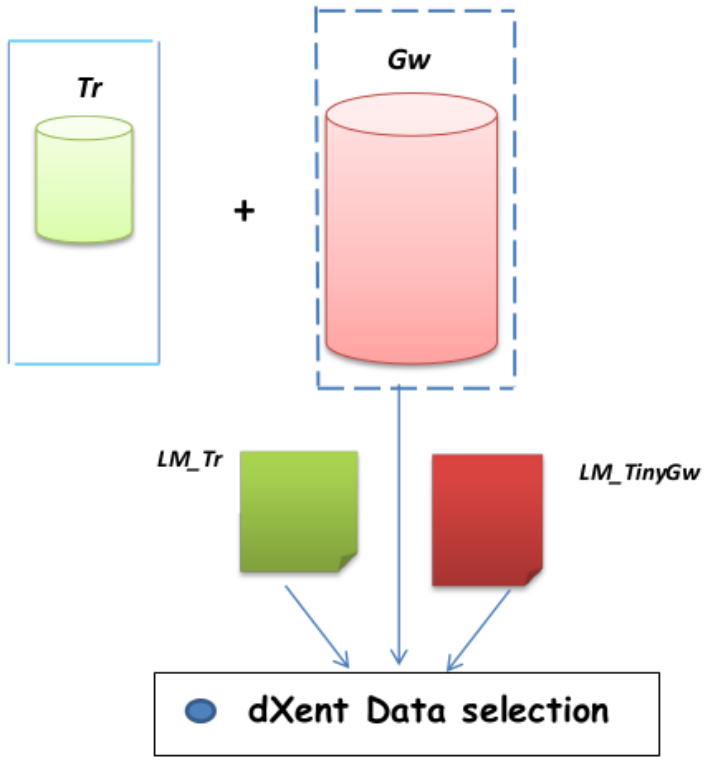
Sélection des données textuelles.

La sélection par **dXent** et **MSDP** nécessitent l'apprentissage des MLs pour le calcul des scores du corpus non-domain **Gw**, de même pour le Np & Web.



Sélection de données text.- dXent

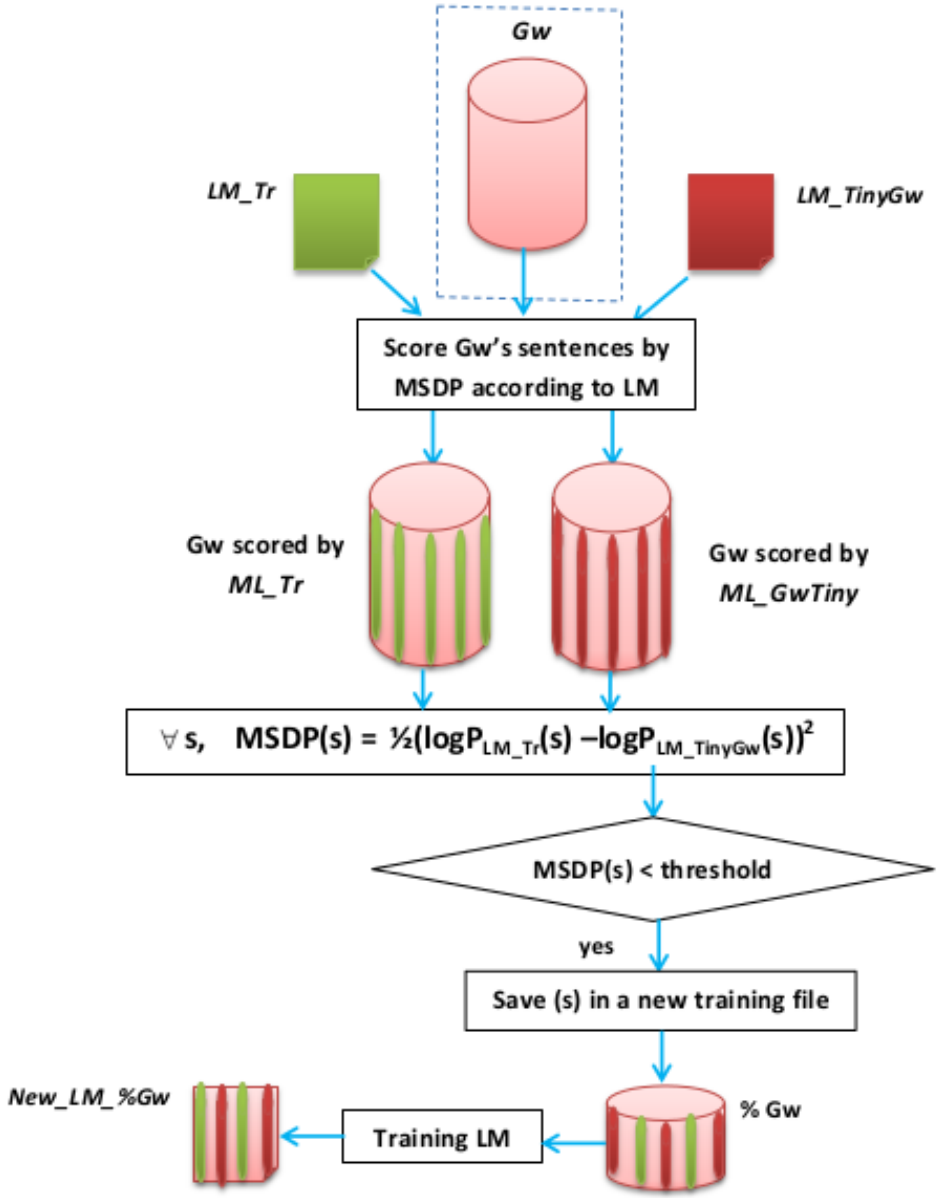
- Sélection par différence d'entropie croisée (**dXent**) : du corpus non-domain, où chaque phrase est scorée par le dXent par le $ML_{in-domain}$ et le $ML_{tiny-non-domain} \in \{\text{Web, Np, Gw}\}$
- **Formule 1 :** $\forall$ S $\in Gw$, $dXent(S) = H_{ML-Tr}(s) - H_{ML-TinyGw}(S)$



Sélection de données textuelles- MSDP

- Sélection par différence quadratique de propbabilités (**MSDP**) : du **corpus non-domain**, où chaque phrase est scorée par la MSDP par le $ML_{in-domain}$ et le $ML_{tiny-non-domain} \in \{\text{Web ou Gw}\}$
- **Formule 2** :$\forall$ S $\in$ *Gw*,
  $MSDP(S) = \frac{1}{2}(logP_{ML-Tr}(S) - logP_{ML-TinyGw}(S))^2$

Sélection des données textuelles – Gw



Sélection des données MSDP – Web

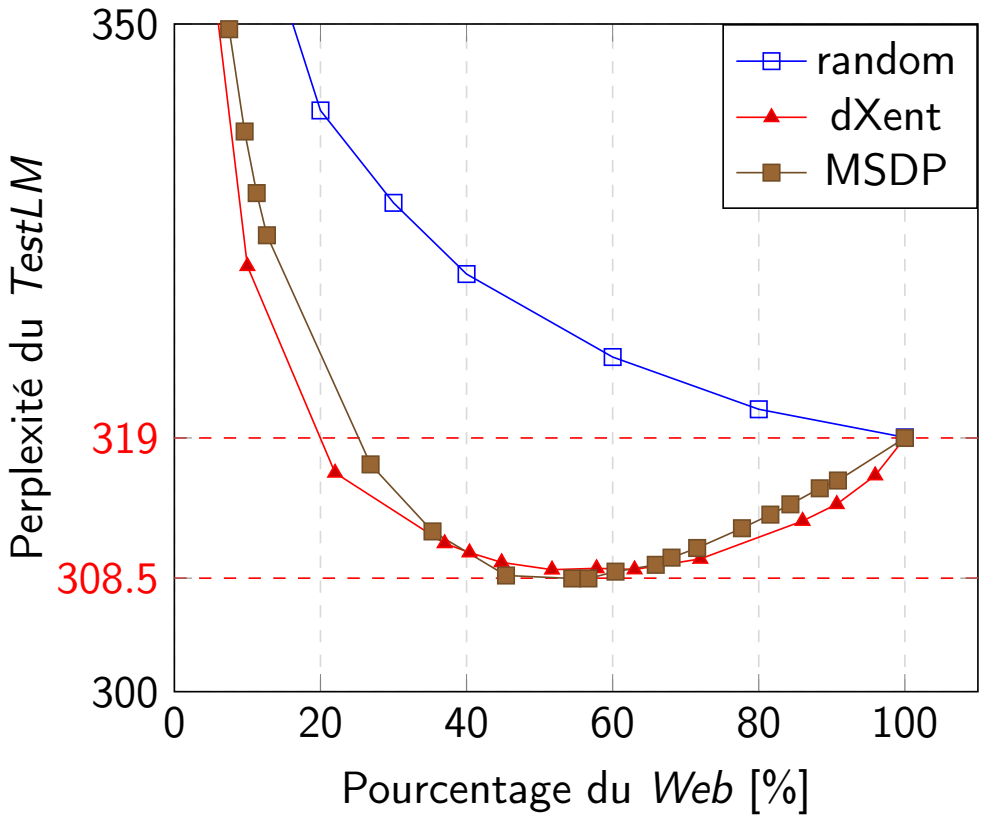


Figure : Perplexité du ML-Test appris sur les données sélectionnées du *Web*.

La sélection est appliquée sur les données du *Web* par : (1) la sélection aléatoire (random), (2) la différence de l'entropie croisée *dXent* calculée sur modèles $ML - Tr$ et $ML - WebTiny$, (3) la différence quadratique du log-probabilité *MSE* calculée entre les modèles $ML - Tr$ et $ML - WebTiny$.

Sélection des données MSDP – Np

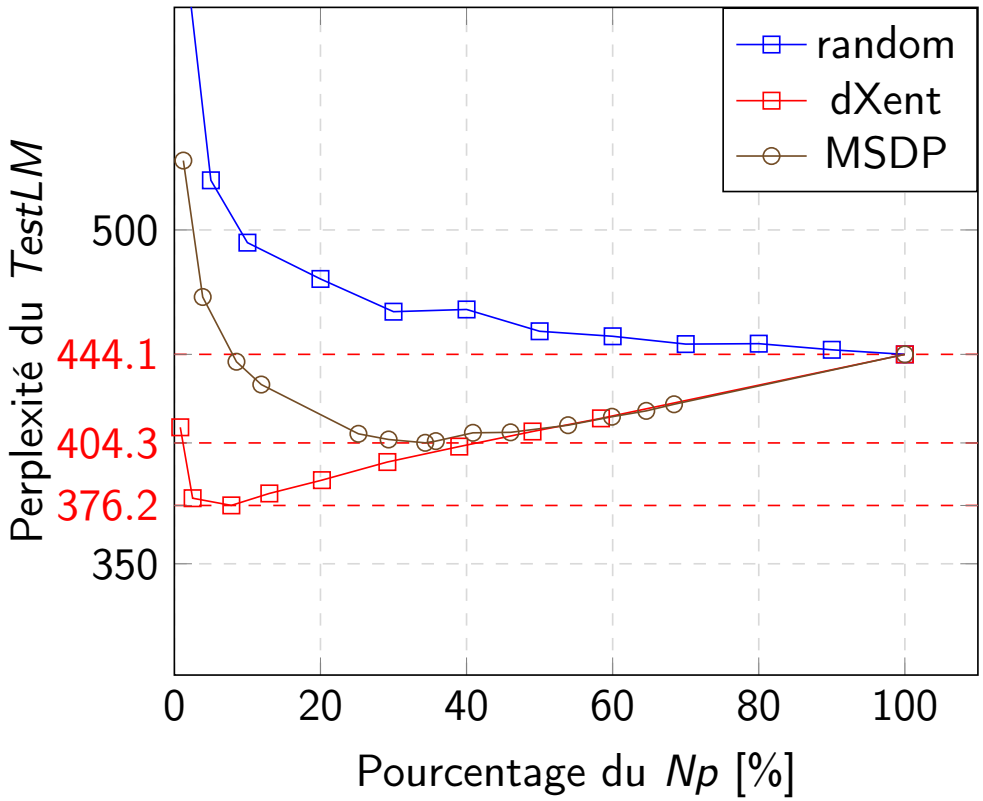


Figure : Perplexité du ML-Test appris sur les données sélectionnées du *Np*.

La sélection est appliquée sur les données du *Np* par : (1) la sélection aléatoire (random), (2) la différence de l'entropie croisée *dXent* calculée sur modèles $ML - Tr$ et $ML - NpTiny$, (3) la différence quadratique du log-probabilité *MSE* calculée entre les modèles $ML - Tr$ et $ML - NpTiny$.

Sélection des données MSDP – Gw

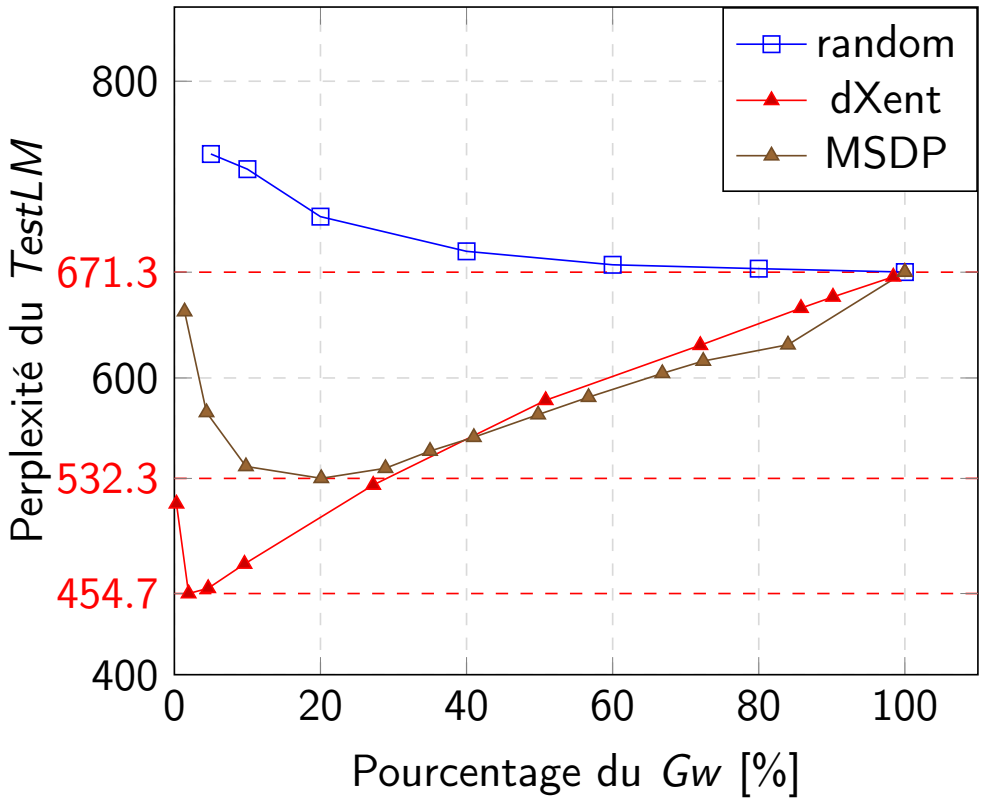


Figure : Perplexité du ML-Test appris sur les données sélectionnées du *Gw*.

La sélection est appliquée sur les données du *Gw* par : (1) la sélection aléatoire (random), (2) la différence de l'entropie croisée *dXent* calculée sur modèles $ML - Tr$ et $ML - GwTiny$, (3) la différence quadratique du log-probabilité *MSDP* calculée entre les modèles $ML - Tr$ et $ML - GwTiny$.

Conclusion

- Les MLs n-gramme restent utilisés et préferés dans des situations modérées (Matériel et qt données) et même dans des SRAP de l'état de l'art (vitesse & performance).
- Pour la selection des données textuelles, le choix des MLs réprésentant le in-domain et le non-domain est important.
- La sélection des données sur le **Gigaword** est plus critique que la sélection des corpora Web & Np avec le critère **dXent MSDP**.
- La sélection de données a base de MSDP est compétitive à la sélection de l'état de l'art a base de **dXent**.

Conclusion

- Les MLs n-gramme restent utilisés et préferés dans des situations modérées (Matériel et qt données) et même dans des SRAP de l'état de l'art (vitesse & performance).
- Pour la selection des données textuelles, le choix des MLs réprésentant le in-domain et le non-domain est important.
- La sélection des données sur le **Gigaword** est plus critique que la sélection des corpora Web & Np avec le critère **dXent MSDP**.
- La sélection de données a base de MSDP est compétitive à la sélection de l'état de l'art a base de **dXent**.
- d'autres techniques sont envisagées pour la sélection des données textuelles.
- Explorer la modélisation du langage Arabe dans le contexte de la reconnaissance de la parole.

Contribution Personnelle

- F. Mezzoudj, et al. On the Optimization of Multiclass SVMs Dedicated to Speech Recognition. ICONIP 2012, Part II, Lncs 7664, pp. 1-8, **2012** . Qatar, Doha.

Contribution Personnelle

- F. Mezzoudj, et al. On the Optimization of Multiclass SVMs Dedicated to Speech Recognition. ICONIP 2012, Part II, Lncs 7664, pp. 1-8, **2012** . Qatar, Doha.
- F. Mezzoudj, et al. On an empirical study of smoothing techniques for a tiny LM. In IPAC, Batna, Algeria, ACM, **2015**a.
- F. Mezzoudj, et al. Textual data selection for language modelling in the scope of automatic speech recognition. In ICNLSP, Algeria, **2015**b.
- F. Mezzoudj, et al. Textual data selection for language modelling in the scope of automatic speech recognition. In PCS, **2018**a.
- F. Mezzoudj, et A. Benyettou. Textual data selection on MSDP for Language Modeling. CITIM, 9-10 October, **2018**b. Mascara, Algeria.

Contribution Personnelle

- F. Mezzoudj, et al. On the Optimization of Multiclass SVMs Dedicated to Speech Recognition. ICONIP 2012, Part II, Lncs 7664, pp. 1-8, **2012** . Qatar, Doha.
- F. Mezzoudj, et al. On an empirical study of smoothing techniques for a tiny LM. In IPAC, Batna, Algeria, ACM, **2015**a.
- F. Mezzoudj, et al. Textual data selection for language modelling in the scope of automatic speech recognition. In ICNLSP, Algeria, **2015**b.
- F. Mezzoudj, et al. Textual data selection for language modelling in the scope of automatic speech recognition. In PCS, **2018**a.
- F. Mezzoudj, et A. Benyettou. Textual data selection on MSDP for Language Modeling. CITIM, 9-10 October, **2018**b. Mascara, Algeria.
- F. Mezzoudj and A. Benyettou. An empirical study of statistical language models : n-grams LMs vs. neural network LMs. In IJICA, Vol.9, No.4, pp.189-202, **2018**c. Doi : 10.1504/IJICA.2018.10016827.

Merci
pour votre attention !