

Kaldi est une boîte à outils open source conçue pour le traitement des données vocales. Elle est utilisée dans les applications vocales, principalement pour la reconnaissance vocale, mais aussi pour d'autres tâches, comme la reconnaissance et la diarisation du locuteur.

La première étape consiste à préparer le corpus d'entraînement, il faut construire :

- 1-Un dataset d'enregistrement vocaux labellisé par le texte prononcé pour chaque enregistrement.
- 2-Un dictionnaire de prononciation (un lexicon): afin de le construire il faut découper chaque mot en séquence de phonème associé.

La deuxième étape consiste à avoir en disposant un cluster d'au moins 10 cœurs ou un cuda gpu, afin d'avoir les résultats dans un temps raisonnable.

Ayant le corpus d'entraînement de la langue cible, on va entraîner un modèle statistique de reconnaissance de parole, ce modèle va assigner des probabilités au phrasé (vecteur de mots) qui sont susceptibles de générer la courbe ondulatoire associée.

Result:

Après avoir fait l'entraînement en mettant à jour les variables de configuration et en exécutant `./run.sh`, on lance `./Result` pour avoir les ctm des différents tests.

```
Q alae@alae-Inspiron-13-5378: ~/Desktop/kaliditest
alae@alae-Inspiron-13-5378: ~/Desktop/kaliditest
WadeDavis_2003 A 1301.59 0.52 exist 1.00
WadeDavis_2003 A 1302.14 0.10 in 0.96
WadeDavis_2003 A 1302.24 0.06 a 0.99
WadeDavis_2003 A 1302.30 0.48 dtverse 1.00
WadeDavis_2003 A 1302.78 0.41 way 0.95
WadeDavis_2003 A 1303.44 0.10 that 0.97
WadeDavis_2003 A 1303.54 0.10 we 1.00
WadeDavis_2003 A 1303.64 0.15 can 1.00
WadeDavis_2003 A 1303.79 0.37 find 1.00
WadeDavis_2003 A 1304.16 0.03 a 1.00
WadeDavis_2003 A 1304.19 0.27 way 1.00
WadeDavis_2003 A 1304.46 0.14 to 1.00
WadeDavis_2003 A 1304.60 0.17 live 0.99
WadeDavis_2003 A 1304.77 0.07 in 0.93
WadeDavis_2003 A 1304.84 0.05 a 0.88
WadeDavis_2003 A 1304.89 0.39 truly 1.00
WadeDavis_2003 A 1305.28 0.86 multicultural 0.98
WadeDavis_2003 A 1306.14 0.82 pluralistic 1.00
WadeDavis_2003 A 1306.96 0.52 world 1.00
WadeDavis_2003 A 1307.78 0.13 we're 0.50
WadeDavis_2003 A 1307.91 0.50 all 1.00
WadeDavis_2003 A 1308.45 0.09 of 0.95
WadeDavis_2003 A 1308.54 0.08 the 1.00
WadeDavis_2003 A 1308.62 0.44 wisdom 1.00
WadeDavis_2003 A 1309.06 0.11 of 1.00
WadeDavis_2003 A 1309.17 0.21 all 0.99
WadeDavis_2003 A 1309.38 0.55 peoples 0.92
WadeDavis_2003 A 1309.93 0.18 can 1.00
WadeDavis_2003 A 1310.11 0.65 contribute 1.00
WadeDavis_2003 A 1310.97 0.11 to 1.00
WadeDavis_2003 A 1311.08 0.13 our 1.00
WadeDavis_2003 A 1311.21 0.47 collective 1.00
WadeDavis_2003 A 1311.68 0.20 well 0.95
WadeDavis_2003 A 1311.88 0.46 being 0.95
WadeDavis_2003 A 1312.45 0.18 thank 1.00
WadeDavis_2003 A 1312.63 0.06 you 1.00
WadeDavis_2003 A 1312.69 0.15 very 1.00
WadeDavis_2003 A 1312.84 0.23 much 1.00
(base) alae@alae-Inspiron-13-5378:~/Desktop/kaliditest$ cd ..
(base) alae@alae-Inspiron-13-5378:~/Desktop/kaliditest$ ./results.sh
%WER 27.7 | 507 17783 | 75.8 17.4 6.8 3.5 27.7 97.6 | 0.083 | exp/trt1/decode_nosp_dev_rescore/score_10_0.0/ctn.filt.filt.sys
%WER 26.4 | 507 17783 | 76.8 16.1 7.1 3.2 26.4 96.4 | 0.075 | exp/trt1/decode_nosp_dev_rescore/score_11_0.0/ctn.filt.filt.sys
%WER 27.3 | 1155 27500 | 75.4 18.2 6.4 2.8 27.3 93.2 | 0.121 | exp/trt1/decode_nosp_test_rescore/score_11_0.0/ctn.filt.filt.sys
%WER 25.7 | 1155 27500 | 76.9 17.0 6.1 2.7 25.7 92.5 | 0.087 | exp/trt1/decode_nosp_test_rescore/score_11_0.0/ctn.filt.filt.sys
%WER 23.0 | 507 17783 | 80.2 14.4 5.4 3.2 23.0 94.7 | 0.081 | exp/trt2/decode_dev_rescore/score_14_0.0/ctn.filt.filt.sys
%WER 21.9 | 507 17783 | 81.2 13.3 5.4 3.1 21.9 93.5 | 0.060 | exp/trt2/decode_dev_rescore/score_15_0.0/ctn.filt.filt.sys
%WER 23.7 | 507 17783 | 79.4 14.7 5.9 3.1 23.7 94.9 | 0.066 | exp/trt2/decode_nosp_dev_rescore/score_13_0.0/ctn.filt.filt.sys
%WER 22.4 | 507 17783 | 80.4 13.5 6.1 2.9 22.4 93.3 | 0.041 | exp/trt2/decode_nosp_dev_rescore/score_14_0.0/ctn.filt.filt.sys
%WER 23.1 | 1155 27500 | 79.3 15.1 5.6 2.4 23.1 91.2 | 0.106 | exp/trt2/decode_nosp_test_rescore/score_14_0.0/ctn.filt.filt.sys
%WER 22.0 | 1155 27500 | 80.5 14.6 4.8 2.6 22.0 90.2 | 0.039 | exp/trt2/decode_nosp_test_rescore/score_12_0.0/ctn.filt.filt.sys
%WER 22.1 | 1155 27500 | 80.6 14.8 4.6 2.7 22.1 90.8 | 0.122 | exp/trt2/decode_test_rescore/score_14_0.0/ctn.filt.filt.sys
%WER 20.8 | 1155 27500 | 81.9 13.9 4.3 2.6 20.8 90.3 | 0.068 | exp/trt2/decode_test_rescore/score_14_0.0/ctn.filt.filt.sys
(base) alae@alae-Inspiron-13-5378:~/Desktop/kaliditest$ cp exp/trt2/decode_test_rescore/score_14_0.0/ctn
(base) alae@alae-Inspiron-13-5378:~/Desktop/kaliditest$
```

Ainsi, on a récupéré les deux meilleurs scores, puis on affiche leur score :

./cat exp/tri2/decode_test_rescore/score_14_0.0/ctm.filt.filt.sys

```
Q alae@alae-Inspiron-13-5378: ~/Desktop/kalidtest
alae@alae-Inspiron-13-5378: ~/Desktop/kalidtest
#WER 22.4 | 507 17783 | 89.4 13.5 6.1 2.9 22.4 93.3 | 0.041 | exp/tri2/decode_nosp_dev_rescore/score_14_0.0/ctm.filt.filt.sys
#WER 23.1 | 1155 27500 | 79.3 15.1 5.6 2.4 23.1 91.2 | 0.106 | exp/tri2/decode_nosp_test_rescore/score_14_0.0/ctm.filt.filt.sys
#WER 22.0 | 1155 27500 | 80.5 14.6 4.8 2.6 22.0 90.2 | 0.039 | exp/tri2/decode_nosp_test_rescore/score_12_0.0/ctm.filt.filt.sys
#WER 22.1 | 1155 27500 | 80.6 14.8 4.6 2.7 22.1 90.8 | 0.122 | exp/tri2/decode_test/score_14_0.0/ctm.filt.filt.sys
#WER 20.8 | 1155 27500 | 81.8 13.9 4.3 2.6 20.8 90.3 | 0.008 | exp/tri2/decode_test_rescore/score_14_0.0/ctm.filt.filt.sys
(base) alae@alae-Inspiron-13-5378:~/Desktop/kalidtest$ cp exp/tri2/decode_test_rescore/score_14_0.0/ctm
(base) alae@alae-Inspiron-13-5378:~/Desktop/kalidtest$ cat exp/tri2/decode_test_rescore/score_14_0.0/ctm.filt.filt.sys

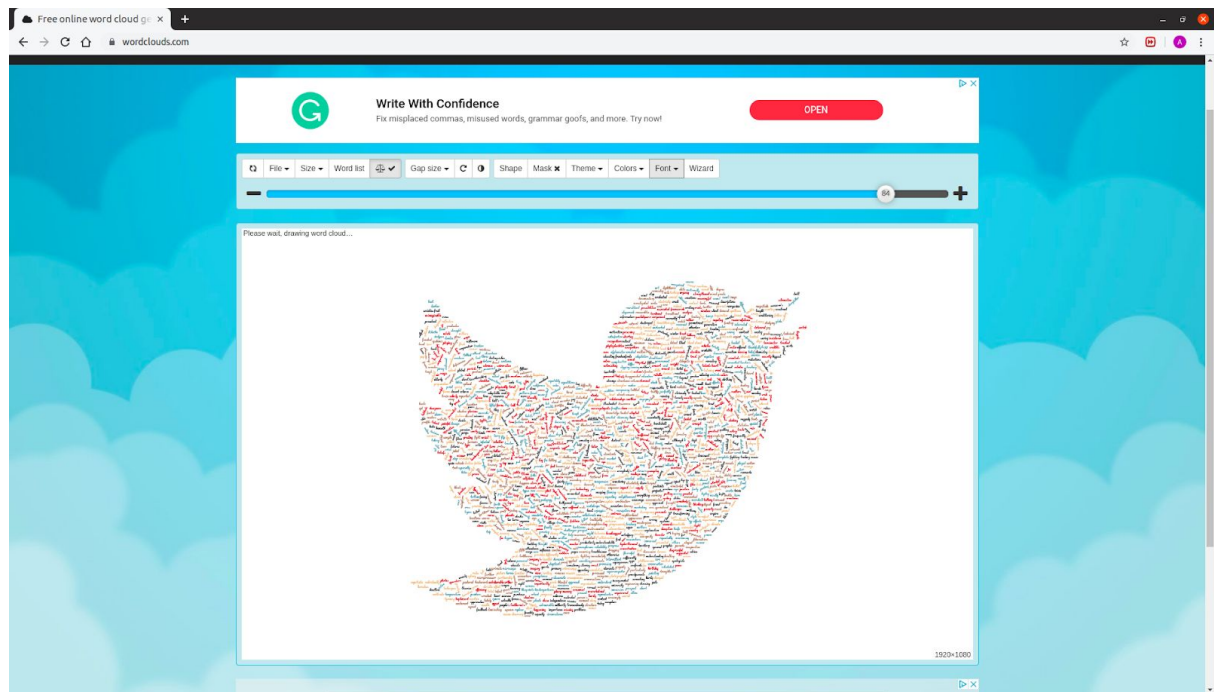
=====
SYSTEM SUMMARY PERCENTAGES by SPEAKER
=====
exp/tri2/decode_test_rescore/score_14_0.0/ctm.filt
=====
SPKR | # Snt # Wrd | Corr Sub Del Ins Err S.Err | NCE
-----|-----
almeenuilins | 129 2897 | 81.5 14.5 4.0 2.7 21.3 87.6 | 0.091
billgates | 151 4342 | 83.6 13.1 3.3 3.3 19.7 96.0 | -0.046
s182 | 14 302 | 74.8 19.2 6.0 3.3 28.5 85.7 | -0.579
danbarber | 235 2383 | 71.8 21.2 7.0 2.5 30.7 81.3 | 0.074
danbarber_2010_s103 | 1 24 | 54.2 29.2 16.7 0.0 45.8 100.0 | 0.273
danielkahnenan | 127 3015 | 83.8 13.6 2.5 2.3 18.5 89.8 | 0.096
s164 | 11 168 | 51.8 31.0 17.3 3.0 51.2 100.0 | -0.398
ertlnead_2009p_ertlnead | 52 1510 | 81.7 12.3 6.0 2.1 20.4 94.2 | 0.115
garyflake | 35 1102 | 84.1 12.9 3.0 1.9 17.8 91.4 | -0.117
jamescameron | 95 2972 | 83.6 10.9 5.4 2.4 18.8 95.8 | 0.099
janengongtal | 108 3819 | 82.7 13.7 3.6 2.5 19.7 97.2 | 0.071
michaelspecter | 124 2960 | 81.1 14.0 5.0 2.2 21.1 93.5 | 0.125
robertgupta | 37 876 | 89.0 7.6 3.3 2.2 13.1 75.7 | 0.338
robertgupta_2010u_s57 | 1 2 | 0.0 0.0 100.0 0.0 100.0 100.0 | 0.000
tomwujec | 35 1122 | 83.8 12.7 3.5 3.8 20.1 97.1 | 0.174
Sun/Avg | 1155 27500 | 81.8 13.9 4.3 2.6 20.8 90.3 | 0.068
Mean | 77.0 1833.3 | 72.5 15.1 12.4 2.3 29.8 92.4 | 0.021
S.D. | 68.0 1450.5 | 22.8 7.7 24.7 1.1 22.1 7.2 | 0.237
Median | 52.0 1510.0 | 81.7 13.6 5.0 2.4 20.4 94.2 | 0.091
=====
(base) alae@alae-Inspiron-13-5378:~/Desktop/kalidtest$
```

cat exp/tri2/decode_nosp_test_rescore/score_12_0.0/ctm.filt.filt.sys

```
Q alae@alae-Inspiron-13-5378: ~/Desktop/kalidtest
alae@alae-Inspiron-13-5378: ~/Desktop/kalidtest
Sun/Avg | 1155 27500 | 81.8 13.9 4.3 2.6 20.8 90.3 | 0.068
Mean | 77.0 1833.3 | 72.5 15.1 12.4 2.3 29.8 92.4 | 0.021
S.D. | 68.0 1450.5 | 22.8 7.7 24.7 1.1 22.1 7.2 | 0.237
Median | 52.0 1510.0 | 81.7 13.6 5.0 2.4 20.4 94.2 | 0.091
=====
SYSTEM SUMMARY PERCENTAGES by SPEAKER
=====
exp/tri2/decode_nosp_test_rescore/score_12_0.0/ctm.filt
=====
SPKR | # Snt # Wrd | Corr Sub Del Ins Err S.Err | NCE
-----|-----
almeenuilins | 129 2897 | 80.4 15.2 4.5 2.9 22.5 89.1 | 0.068
billgates | 151 4342 | 82.3 14.0 3.7 3.5 21.2 96.0 | -0.082
s182 | 14 302 | 71.5 20.5 7.9 4.0 32.5 85.7 | -0.432
danbarber | 235 2383 | 69.4 22.5 8.1 2.6 33.2 80.4 | 0.014
danbarber_2010_s103 | 1 24 | 41.7 37.5 20.8 0.0 58.3 100.0 | 0.093
danielkahnenan | 127 3015 | 83.1 14.4 2.6 2.6 19.5 88.2 | 0.075
s164 | 11 168 | 51.2 30.4 18.5 3.0 51.8 100.0 | -0.526
ertlnead_2009p_ertlnead | 52 1510 | 81.3 12.0 6.8 2.0 20.7 94.2 | 0.106
garyflake | 35 1102 | 83.5 13.2 3.4 2.0 18.5 91.4 | 0.026
jamescameron | 95 2972 | 82.9 11.5 5.6 2.2 19.3 95.8 | 0.105
janengongtal | 108 3819 | 81.5 14.3 4.2 2.5 21.0 94.4 | -0.016
janengongtal | 108 3819 | 81.5 14.3 4.2 2.5 21.0 94.4 | -0.016
michaelspecter | 124 2960 | 79.2 15.2 5.7 1.7 22.5 94.4 | 0.068
robertgupta | 37 876 | 88.2 8.3 3.4 2.3 14.6 83.8 | 0.290
robertgupta_2010u_s57 | 1 2 | 0.0 0.0 100.0 0.0 100.0 100.0 | 0.000
tomwujec | 35 1122 | 83.4 13.4 3.2 3.0 19.6 97.1 | 0.105
Sun/Avg | 1155 27500 | 80.5 14.6 4.8 2.6 22.0 90.2 | 0.039
Mean | 77.0 1833.3 | 70.6 16.2 13.2 2.3 31.7 92.7 | -0.003
S.D. | 68.0 1450.5 | 23.4 8.9 24.6 1.1 22.7 6.1 | 0.213
Median | 52.0 1510.0 | 81.3 14.3 5.6 2.5 21.2 94.4 | 0.068
=====
(base) alae@alae-Inspiron-13-5378:~/Desktop/kalidtest$
```

Nuage de mot:

Le nuage de mot pour les test associé au meilleur score:

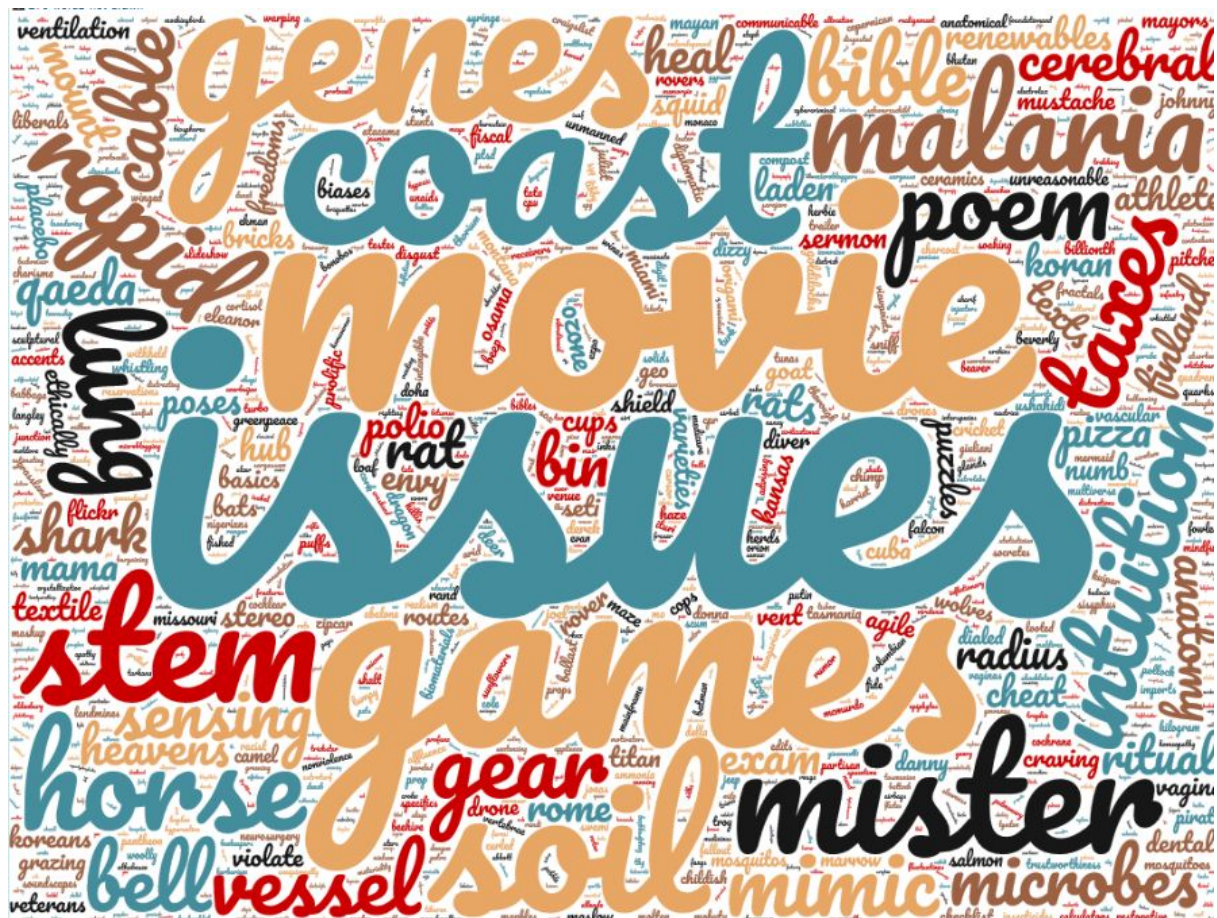


Afin de générer les nuages de mot , on a codé ce petit script Bash:

```
cat ctm | cut -d' ' -f5- > testctm
```

```
awk '{arr[$1]+=$2} END {for (i in arr) {print arr[i],i}}' testctm > newctm
```

Le nuage de mot pour les stms de la ref:



Stop-list établie grâce au tf-idf sur le corpus en prenant les minimums de la liste des scores idf des mots.

Tf-idf:

Formule utilisé pour le tf-idf :

$$f_{t,d} \cdot \log \frac{N}{n_t}$$

Avec:

- $F(t,d)$ = le score Term-fréquence d'un mot dans le corpus
- N , le nombre de documents dans le corpus
- $N(t)$, le nombre de documents dans le corpus contenant le mot "t"

On opte pour la formule avec un log pour faire une sorte que les mots apparaissant dans beaucoup de documents (voir tous pour les stopwords) ai un score proche de 0 (ou égale à 0 pour les stopwords).

De plus, nous utilisons un Stemmer afin de radicaliser les mots en pré-traitement afin d'obtenir de meilleurs résultat sur le tf-idf.

```
-----file1485-----
| directori | 0.3970904463836666
| supercomput | 0.3796771700329087
| web | 0.36412839691923243
-----file1486-----
| postur | 2.010222305901332
| scrotal | 1.404708492327632
| bodygroom | 1.404708492327632
-----file1487-----
| design | 0.5442494459060828
| condom | 0.46074317207845733
| jawbon | 0.4188180992955737
-----file1488-----
| dismay | 3.583912542532603
| motherto | 2.4420316866618834
| japanso | 2.210470151535744
-----file1489-----
| wing | 1.8568346229889905
| yves | 1.6250120108077026
| altitud | 1.0703019180653077
-----file1490-----
| magnet | 1.1633582387397607
| unexpect | 0.5999955000682841
| ink | 0.5683113970667258
-----file1491-----
| merci | 1.8951121504666217
| libyan | 1.8781772529388674
| wake | 1.3454487364290089
-----file1492-----
| war | 1.0756932691156003
| women | 0.7549856430309916
| peac | 0.6635336447528548
-----file1493-----
| it's | 0.5631173967334816
| ze | 0.5073705212427811
| draw | 0.3571154523891858
-----file1494-----
| song | 0.645907677821292
| audio | 0.43835403615264984
| voicemail | 0.38682362317569213
-----file1495-----
| selam | 0.6227839514782635
| teeth | 0.5895216614908214
| fossil | 0.5479116987844257
```

Le screenshot ci-dessus montre les résultats de notre algorithme TF-IDF pour quelques fichiers du corpus. On affiche uniquement les 3 mots avec le meilleur score tf-idf. On voit bien que les mots en résultats décrivent bien la conversation. En comparant les best-scores on retrouve facilement les conversations ayant le même thème.

Notre code est disponible dans les PJs :) !