

# Maitre yoda

- [Classement des modèles de wake-word](#)
- [Idées de phrases](#)
- [Lien en vrac pour le wake-word](#)
- [Tuto / commandes pour entrainer un wake-word](#)
- [Gruut problèmes](#)
- [Musique](#)
- [commande micros](#)
- [Réglage heure](#)
- [Errol](#)
  - [whiteboard](#)
  - [Neo4j](#)
  - [api](#)
- [Todo](#)
- [train tts](#)
- [tts v2 amd](#)

# Classement des modèles de wake-word

## YODA 16

- nombre d'époch : 1800
- Jeu d'entraînement: 92 wake words du 1er dataset (dont music), 489 non-wake-word melange noise et homemade

## Resultat

### **Wake word: 47/243**

- 1er dataset: 3 erreurs / 76
- 1er dataset with music: 0 erreurs / 36
- 2nd dataset: 13 erreurs / 112
- 4eme dataset: 32 erreurs / 76

### **Non wake word: 43/1623**

- homemade: 30/368
- homemade 2: 25/25
- open soud: 1/593
- amelia: 3/223
- athena: 9/483

# Idées de phrases

## Météo

- Quel temps fera t'il <jour de la semaine>
- Est ce qu'il va faire beau ?
- Est ce qu'il va pleuvoir ?

## Musique

## General

- qui est pierre ?
  - il y en a beaucoup: <liste des pierres>
- qu'est ce qu'une pierre ?
  - definition de wikipedia ?
- qui est francois mitterand ?
  - generation d'un phrase selon les données wikidata
- quand est né jean jaurés ?
  - il est né le <date>
- quel est la date de naissance de Francois 1er.

# Lien en vrac pour le wake-word

<https://github.com/mindorii/kws>

<https://community.rhasspy.org/t/record-raspberry-raspberrypi-for-a-distributed-dataset/2160/11>

<https://discourse.nodered.org/t/node-red-contrib-voice2json/37925/14>

<https://github.com/JuliaDSP>

<https://medium.com/snips-ai/machine-learning-on-voice-a-gentle-introduction-with-snips-personal-wake-word-detector-133bd6fb568e>

<https://speechbrain.github.io/>

<https://github.com/linto-ai/linto-desktoptools-hmg>

<https://github.com/MycroftAI/mycroft-precise/pull/141>

# Tuto / commandes pour entrainer un wake-word

## I - Trouver plein de bruits

### Extraire des bandes-sons de film:

```
ffmpeg -i /data/films/La\ grande\ vadrouille.avi -acodec pcm_s16le -ar 16000 -ac 1 -f wav la-grande-vadrouille.wav  
ffmpeg -i /data/films/Kill\ Bill\ Vol\ 1\ \[1080p\]\ MULTI\ 2003\ BluRay\ x264-Pop\ .mkv -acodec pcm_s16le -ar 16000 -ac 1 -f wav kill-bill.wav  
ffmpeg -i /data/films/Harry\ Potter\ 4\ Et\ La\ Coupe\ De\ Feu\ 2005\ \[1080p\]\ MULTI\ 2004\ Bluray\ X264\ lorraines.mkv -acodec pcm_s16le -ar 16000 -ac 1 -f wav hp-4.wav  
ffmpeg -i /data/films/Harry\ Potter\ 1\ A\ L'\Ecole\ Des\ Sorciers\ \[1080p\]\ MULTI\ 2001\ Bluray\ X264\ lorraines.mkv -acodec pcm_s16le -ar 16000 -ac 1 -f wav hp-1.wav  
ffmpeg -i /data/films/Hunger\ Games\ 1.mkv -acodec pcm_s16le -ar 16000 -ac 1 -f wav hunger-games-1.wav  
ffmpeg -i /data/films/Pulp\ Fiction.avi -acodec pcm_s16le -ar 16000 -ac 1 -f wav pulp-fiction.wav  
ffmpeg -i /data/films/DRAGONS\ \[2010\].avi -acodec pcm_s16le -ar 16000 -ac 1 -f wav dragon.wav  
ffmpeg -i /data/films/DRAGONS\ \[2010\].avi -acodec pcm_s16le -ar 16000 -ac 1 -f wav galadiator.wav  
ffmpeg -i /data/films/Gladiator\ 2000\ 10th\ Anniversary\ Edition\ MULTI\ TrueFrench\ 1080p\ HDLight\ x265\ HEVC.GHT.mkv -acodec pcm_s16le -ar 16000 -ac 1 -f wav galadiator.wav  
ffmpeg -i /data/films/Alice\ au\ pays\ des\ merveilles.avi -acodec pcm_s16le -ar 16000 -ac 1 -f wav alice.wav
```

### Télécharger des sons de pluie, et de bruit extérieur depuis Youtube:

```
youtube-dl -x "https://www.youtube.com/watch?v=mPZkdNFkNps"  
youtube-dl -x "https://www.youtube.com/watch?v=xNN7iTA57jM"  
youtube-dl -x "https://www.youtube.com/watch?v=BOdLmxy06H0"  
youtube-dl -x "https://www.youtube.com/watch?v=BOdLmxy06H0"
```

# Télécharger des sons de pubs et de voiture depuis soundarchive

## Télécharger des podcasts

J'ai utilisé l'appli de podcast de mon telephone: AntennaPod.

- telecharger le podcast
- Appui long sur la piste -> partager -> envoi avec kde connect

J'ai 20 fichiers qui font chacun entre 30 minutes et 1h30 de long.

## Récupérer de la musique:

J'ai copié 140 pistes de differents style d'albums (j'ai mis une grosse part de chanson francaise).

## Récupérer des wake-word opensource:

<https://github.com/MycroftAI/precise-community-data>

j'ai pris ceux qui ressemblaient au miens, à savoir:

- athena
- amelia

## II - Enregistrer plein de wake word

- j'en ai enregistré une centaine via le micro de la reconnaissance vocal en me mettant à plein d'endroit dans ma chambre, avec `precise-collect`
- j'en ai enregistré ensuite 10 par personnes volontaire via le micro de mon pc:
  - Ma famille (6 personnes)
  - Des amis (3 personnes)
- J'en ai enregistré une bonne 50aine, avec les micros de differents pcs

# III - Nettoyer les enregistrements de wake word

J'ai utilisé audacity, et je l'ai fait à la main (ptêtre qu'avec des script ca se fait aussi).

En gros, j'ai verifié qu'il n'y ai que le mot-clé dans l'enregistrement, et j'ai coupé si il y avait trop de silence au début ou à la fin.

Ca me fait des enregistrement qui font entre 0.8s et 2s, avec un peu de blanc au début et à la fin.

# IV - Convertir et découper les source de bruit

## Convertir en wav:

```
SOURCE_DIR=music
DEST_DIR=music-wav/

# selon les extensions de fichier, changez *.mp3 par *.<extension> (youtube-dl télécharge en m4a)
for i in $SOURCE_DIR/*.mp3; do echo "Converting $i..."; fn=${i##*/}; ffmpeg -i "$i" -acodec pcm_s16le -ar 16000 -ac 1 -f wav "$DEST_DIR/${fn%.*}.wav"; done
for i in $SOURCE_DIR/*.flac; do echo "Converting $i..."; fn=${i##*/}; ffmpeg -i "$i" -acodec pcm_s16le -ar 16000 -ac 1 -f wav "$DEST_DIR/${fn%.*}.wav"; done
```

## Decouper par tranche de 30s:

```
for f in *.wav; do sox "$f" "split.$f" trim 0 30 : newfile : restart ; done
```

# V - Augmenter artificiellement vos mots-clés:

Il faut d'abord choisir certaines partie du bruit précédents qu'on va ajouter par dessus nos enregistrement de mots-clé J'ai pris:

- Les videos youtube
- Le son de soundarchive
- Et des morceaux de classique

Pour être tranquille, on les normalise avec sox:

```
SOURCE_DIR=noise-short/  
DEST_DIR=noise-norm/  
for i in $SOURCE_DIR/*.wav; do echo "Converting $i..."; fn=${i##*/}; sox --norm "$i"  
"$DEST_DIR/${fn%.*}.wav"; done
```

Pis pour chaque groupe, on va les appliquer avec ce script sur nos wake-word (vous pouvez changer le coeff 0.1 par 0.05 ou 0.2 ou 0.3 selon vos enregistrement):

```
#!/bin/bash  
  
NOISEDIR=$1  
  
DIR="$( cd "$( dirname "${BASH_SOURCE[0]}" )" >/dev/null 2>&1 && pwd )"  
  
for f in *.wav  
do  
    NOISEFILE=$(find ${NOISEDIR} -type f | shuf -n 1)  
  
    sox -m -v 0.9 $f -v 0.1 ${NOISEFILE} noise.$f trim 0 `soxi -D $f`  
done
```

Normalement après ça on a triplé nos wake-words.

## VI - Installer mycroft precise:

<https://github.com/MycroftAI/mycroft-precise#source-install>

1. il faut python 3.6 (pour pas se prendre la tête installer ubuntu 18.04)
2. il y a un bug dans le requirements.txt, après l'installation: `pip install h5py<3.0.0`

## VII - Entraînement:

- Mettre 80/90% de vos wake-word dans le dossier wake-word et le reste dans test/wake-word
- `precise-train yoda1.net maitre_yoda/ -e 100 -s 0.5`
- Faire un dossier avec tout vos noise puis `precise-train-incremental yoda1.net maitre_yoda/ -r path/to/noise-folder -e 50 -th 0.4 -s 0.5`
- Attendre une bonne heure avec un pc de gamer de l'an 2019



IIX - Profitez !

# Gruut problèmes

- de
- train
- accueillir
- 2021

# Musique

- play
- pause (copliqué avec la musique)
- jouer un artiste
- jouer un genre
  - jouer un album classé jazz?
  - faire une playlist composé de chansons random qui proviennent des trois 1ere piste de tout les albums classé jazz
  -

## MPD

- [The music database] listall => `genres/<le_genre>` ou `biblio/<artiste>` ou `Favori/` => <https://python-mpd2.readthedocs.io/en/latest/topics/commands.html#MPDClient.listFiles>
- select a random dir (not played recently ?)
- [queue] add <uri/random\_selected> => <https://python-mpd2.readthedocs.io/en/latest/topics/commands.html#MPDClient.add>

## search

- find and searchadd

# commande micros

`arecord -f L16_LE --rate 16000 toto.wav` : enregistrer un fichier avec le micro en 16 kHz 256kb/s

`precise-listen ~/.config/rhasspy/profiles/fr/precise/yodav3.1.pb` : tester precise

`pacmd list-sources` : lister les entrées audio

## ALSA

### Lister les cartes audio:

`arecord -l`

`aplay -l`

### Mettre par défaut une carte:

`/etc/asound.conf`

```
pcm.!default {  
    type plug  
    slave.pcm {  
        type hw  
        card 2  
    }  
}
```

## liens

<https://github.com/MycroftAI/mycroft-precise/wiki/Training-your-own-wake-word#how-to-train-your-own-wake-word>

<https://github.com/respeaker/seeed-voicecard>

# Réglage heure

- ne pas dire "zero" quand il est "04h00"
- dire "une" dans "21h..."

# Errol

# whiteboard

- combien de sous me reste t'il ?
- combien coute une baguette de pain ?
- combien y a t'il de proteines dans 100g de pomme de terre ?

1. Analyse de la grammaire. On cherche une quantité de "protéine" contenu dans "100g de pomme de terre"
2. Recherche de la definition de "proteine" et de "100g de pomme de terre"

I - Simplification des phrase complexe en phrase simple

I.2 - "Traduction" des verbes compliqué en verbe simple grace au cnrtl ou autre moteur de synonyme:

- import dans neo4j ?
- parcours de graph -> trouver le synonyme connu le plus proche ?

II - conversion de phrase simple en requête RDF ou en requête neo4j ?

I.1 découper en mot et phrase

## Lien en vrac:

- <https://crisco2.unicaen.fr/des/synonymes/contenir>
- <https://www.atilf.fr/ressources/tlfi/>
- <http://www.nooj4nlp.org/index.html>
- <https://www.modyco.fr/fr/base-documentaire/ressources/jean-dubois.html> => dictionnaires francais pour machines
- [talep.lif.univ-mrs.fr/FondamenTAL/](http://talep.lif.univ-mrs.fr/FondamenTAL/) => même dictionnaire
- <http://rali.iro.umontreal.ca/rali/?q=fr/versions-informatisees-lvf-dem>
- <https://dumps.wikimedia.org/frwiktionary/>
- <http://infolingua.univ-mlv.fr/DonneesLinguistiques/Dictionnaires/telechargement.html>

Errol

# Neo4j

## Recherche:

```
MATCH (v1:verbe {label:'réfréner'}), (v2:verbe {label:'contenir'})  
RETURN v1,v2
```

## Creer un index

```
CREATE INDEX FOR (verbe:label) ON (verbe.label)
```

## Creer un noeud

```
CREATE (contenir:verbe {label:"contenir"})  
return contenir
```

## Faire un relation plus un nouveau noeud

```
MATCH (contenir:verbe {label:"contenir"})  
CREATE (contenir)-[synonyme:SYNONYME {poid:3}]->(refrener:verbe {label:"réfréner"})  
RETURN contenir,synonyme,refrener
```

## Faire une relation entre deux noeuds existants

```
MATCH (u:User {username:'admin'}), (r:Role {name:'ROLE_WEB_USER'})  
CREATE (u)-[:HAS_ROLE]->(r)
```



Errol

# api

<https://www.wikidata.org/w/api.php?action=wbsearchentities&language=fr&search=pr%C3%A9sident&type=property&format=json>

# Todo

1. correcteur orthographique
2. traitement récursif des triplets (ex: Quelle est la taille du premier-ministre de la france ?)
3. améliorer la tokenisation sur les mots "courants" (special rules spacy)
4. vérifier le vrai sens (président de l'angleterre => ca donne le roi, donc verifier après coup que c'est bien un président et pas un roi)
5. compréhension des verbes ?
6. gérer les mots type 'français' 'anglais' synonyme de "de la France" "de l'Angleterre"
7. gérer le cas du premier ministre anglais (qui n'a pas de relation dans wikidata)
8. gérer le cas où la recherche est nulle (la chine =>  
<https://www.wikidata.org/wiki/Q113473434>) (peut être un soucis du point 3)
9. réfléchir à l'adapter pour un usage actionnel 'allume la lampe du salon'
10. gérer les synonymes

Qui est le président du pays qui a bâti la tour eiffel ?

Quand a été bâtie la tour eiffel

comment se nomme le président de la france ?

Quel est la fonction d'Emmanuel macron ? (ca fonctionne déjà :p)

Quelle est la couleur des yeux de Xi Jinping ? => ('couleur des yeux', 'xi Jinping')

# train tts

## script to divide audio

```
srt_file=$1
audio_file=$2

#rm -Rf wav_data.csv

mkdir -p wav

function convert_to_second {
    IFS=: read -r h m s <<<"$1"

    echo $(((h * 60 + m) * 60 + s))
}

function cut_part_from_file {
    FROM=$1
    TO=$2
    INPUT=$3
    OUTPUT=$4

    #LENGTH=$(( $TO - $FROM ))

    ffmpeg -ss $FROM -to $TO -i $INPUT -ar 22050 $OUTPUT -hide_banner -loglevel error
}

stringContain() { case $2 in *$1*) return 0;; *) return 1;; esac ;}

is_line_after_time="false"
counter=0
```

```

IFS=$'\n';
for line in $(cat $srt_file); do
    echo $line
    if [ ! -z "$line" ]; then

        if [[ $is_line_after_time == "true" ]]; then
            is_line_after_time="false"
            echo "$counter|$line" >> _data.csv
        fi

        if [[ $line =~ "-->" ]]; then

            echo $line

            let "counter+=1"
            is_line_after_time="true" # true in bash
            start_time=$(echo $line | awk -F' --> ' '{print $1}' | sed 's/,./g')
            end_time=$(echo $line | awk -F' --> ' '{print $2}' | sed 's/,./g')

            echo $start_time " to " $end_time

            #start_time_in_s=$(convert_to_second $start_time)
            #end_time_in_s=$(convert_to_second $end_time)

            #echo $start_time_in_s " to " $end_time_in_s

            cut_part_from_file $start_time $end_time $audio_file "wav/$counter.wav"
        fi

    fi
done

```

# train

```

python3.10 -m venv env-piper
source env-piper/bin/activate.fish
pip install wheel setuptools
git clone https://github.com/rhasspy/piper.git

```

```
cd piper/src/python/  
pip install -e .  
./build_monotonic_align.sh  
pip install torchmetrics==0.11.4
```

ajouter ligne 232 ".local/lib/python3.10/site-packages/torch/utils/data/dataloader.py"

```
num_workers = 60
```

```
python3 -m piper_train.preprocess --language fr --sample-rate 22050 --dataset-format ljspeech --single-speaker  
--input-dir /home/tjiho/info/ia/input/ --output-dir /home/tjiho/info/ia/output/
```

```
export 'PYTORCH_CUDA_ALLOC_CONF=max_split_size_mb:256'
```

```
python3 -m piper_train \  
    --dataset-dir /home/tjiho/info/ia/output/ \  
    --accelerator 'gpu' \  
    --devices 1 \  
    --batch-size 32 \  
    --validation-split 0.0 \  
    --num-test-examples 0 \  
    --max_epochs 5000 \  
    --checkpoint-epochs 1 \  
    --precision 32 \  
    --resume_from_checkpoint /home/tjiho/info/ia/base-siwis/epoch=3304-  
step=2050940.ckpt
```

```
python3 -m piper_train.export_onnx ~/output/lightning_logs/version_0/checkpoints/epoch\=1314-  
step\=63120.ckpt ~/output/model.onnx
```

# tts v2 amd

```
pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/rocm6.1  
HSA_OVERRIDE_GFX_VERSION=11.0.0 python ...
```