Phonochrome

Manuel d'utilisation

[redacted]

Lignes censurées = identifiants github/serveur, à demander aux créateur si besoin.

Note :
Tous les logiciels utilisés sont distribués librement et réutilisables/modifiables à volonté.

Table des matières

I. Logiciels requis & installation :

Programmes requis :

- php5+, python2 & python3+
- Un système Unix (Linux de préférence, Programme testé et développé sous Ubuntu 16.04 LTS)
- OpenFST (dépendance de Phonetisaurus)
- Phonetisaurus (Logiciel de correspondance Graphème-Phonème)
- GoogleNgramLibrary (pour pouvoir faire les calculs de n-gram pour les couleurs et créer les modèles de langue pour Phonetisaurus.)
- Modules Python3 « colormaths » & « webcolors » pour permettre de passer du format d'une couleur à une autre, exemple RGB → LAB

Note : de manière générale nous vous recommandons d'installer un gestionnaire de paquet tel que Synaptic sous Debian/Ubuntu permettant d'installer plus facilement vos programmes plutôt que de le faire en ligne de commande.

Installation du toolkit « OpenFST » : (dépendance de Phonetisaurus)

Description : OpenFST est une librairie de manipulation de transducteurs à états finis permettant de créer des automates fonctionnant de manière statistiques avec un ensembles de valeurs permettant d'influencer le résultat en sortie. Ces technologies sont souvent utilisées en reconnaissance de parole, en synthèse vocale, en apprentissage automatique ou en reconnaissance optique de caractère par exemple.
Version utilisée dans Phonochrome : 1.6.1
Page de téléchargement :
http://www.openfst.org/twiki/bin/view/FST/WebHome
Note : Les instructions d'installation sont présentes dans le fichier README contenu dans l'archive lors du téléchargement du programme.
Ne pas oublier de réaliser l'étape « ./configure » avec les arguments suivant : « --enable-static --enable-shared --enable-far --enable-lookahead-fsts --enableconst-fsts --enable-pdt --enable-ngramfsts --enable-linear-fsts CC=gcc-4.9 »

Installation du toolkit « Phonetisaurus » (affichage du texte colorisé)

Description : Phonetisaurus est un programme de détection et d'alignement graphème à phonème (g2p) permettant d'entrainer un modèle de langue donnée comportant n'importe quel caractère afin de pouvoir en sortie modifier ou non les règles d'alignement et permettre par la suite de produire des « intuitions » de phonétisation pour des mots qui ne sont pas présent dans le modèle de langue de base.
Version utilisée dans Phonochrome : Phonetisaurus branch openfst 1.6.1
Page de téléchargement GitHub :
https://github.com/AdolfVonKleist/Phonetisaurus/tree/openfst-1.6.1

Installation du toolkit « GoogleNgramLibrary » (traitement préalable des corpus)

Description : GoogleNgramLibrary constitue un ensemble d'outils très performants permettant de manipuler des corpus de tailles et de symboles divers dans des opérations d'extractions de statistiques, de fusions de données statistiques, de fusions de corpus ou simplement de création de fichiers ngram. Le format privilégié de sortie de ce programme est le format « ARPA », à ne pas confondre avec le format « ARPABET ».

Installation des modules Python 3 Colormath et Webcolors (calcule des contrastes de couleurs) :
Pour cela vous devez préalablement avoir le programme « **pip3** » installé sur votre machine :

Colormaths : (calcul de la distance Delta E et conversion de couleurs) : http://pythoncolormath.readthedocs.io/en/latest/installation.html installation par pip : **« pip3 install colormath »**

Webcolors : (conversion de couleurs aux formats acceptés par HTML/CSS) https://pypi.python.org/pypi/webcolors Installation par pip : **« pip3 install webcolors »**

II. Phonochrome : l'interface graphique web

L'utilisation de Phonochrome depuis son interface web est plutôt aisée. Nous allons ainsi étudier le fonctionnement de chaque page composant la plateforme.

a) L'index :

L'interface propose de coloriser un texte que l'utilisateur aura préalablement rentré dans le formulaire prévu à cet effet.
Le traitement est effectué conjointement avec une lecture du fichier de ressource de couleur situé dans le dossier /config de Phonochrome et un script lançant Phonetisaurus avec comme argument chaque token du texte rentré par l'utilisateur.

(code couleur provisoire utilisé sur la capture d'écran)

L'utilisateur a le choix entre plusieurs langues qui sont en fait les noms des dossiers présents dans le dossier /res de Phonochrome. Cette méthode permet de toujours garder l'outil à jour en affichant en temps réel les corpus présents dans le dossier de ressource. Il est à noter que pour que l'outil fonctionne il faut que les fichiers au format .fst (les modèles de langue) aient le même nom que le dossier dans lesquels ils se trouvent dans le dossier 'res'.

Le fichier-modèle de langue utilisé pour coloriser le texte est aussi indiqué en haut du document par une phrase telle que « modèle utilisé : english.fst ».

Lorsque l'utilisateur choisit l'option « coloriser le texte » chaque mot (ou token) du texte en entrée est envoyé comme paramètre dans une ligne de commande exécutant le programme « phoneticize.py » qui représente le cœur de Phonetisaurus : l'alignement graphème et phonème. Par la suite chaque token est enregistré dans l'ordre permettant ainsi de retranscrire la phrase d'entrée en phrase de sortie colorisée une fois le traitement achevé.

La commande de lancement de Phonetisaurus est la suivante : **python2 phoneticize.py -m modele.fst -w 'token';** où « modele.fst est un modèle de langue et où 'token' est une unité à traiter.

Ce script n'utilise pas la commande « **phonetisaurus-g2p** » car celle-ci n'aligne pas les graphies avec les phonèmes correspondants, ici la commande utilisé provient donc du dossier 'script' présent dans le répertoire d'installation de Phonetisaurus une fois celui-ci installé sur votre machine.

Concernant la « checkbox » présente en dessous du formulaire. Cette case indique l'option « Afficher les diphtongues en surlignage » permet de coloriser les diphtongues de deux couleurs : une pour les lettres et l'autre pour le surlignage.
Cette option est utile pour permettre de proposer une alternative à l'affichage de diphtongue par deux couleurs différentes pour les mêmes caractères. En effet compte tenu de la méthode utilisée ici l'affichage peut sembler un peu « bugué » sur de longs texte et d'autant plus sur des navigateurs moins récents.

Exemple en affichage normal (où nous pouvons voir le léger décalage du caractère « g »:

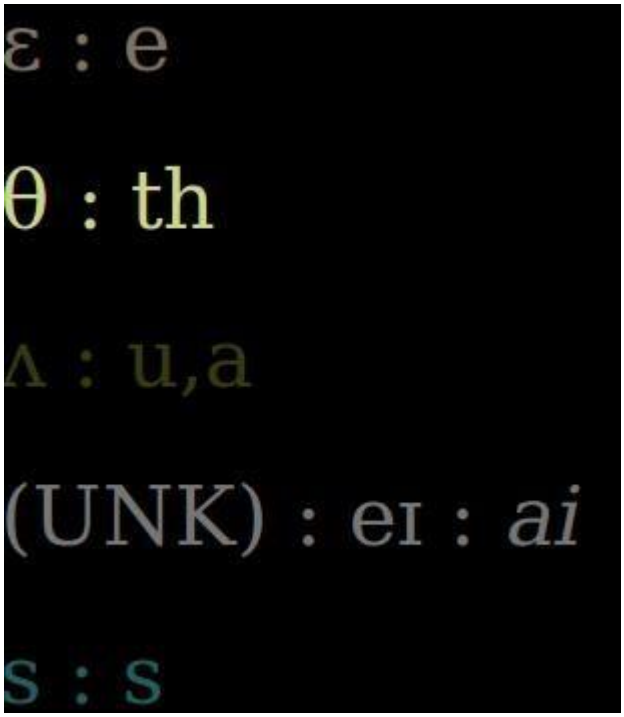
(code couleur provisoire)

Exemple en affichage « surlignage » :

(code couleur provisoire)

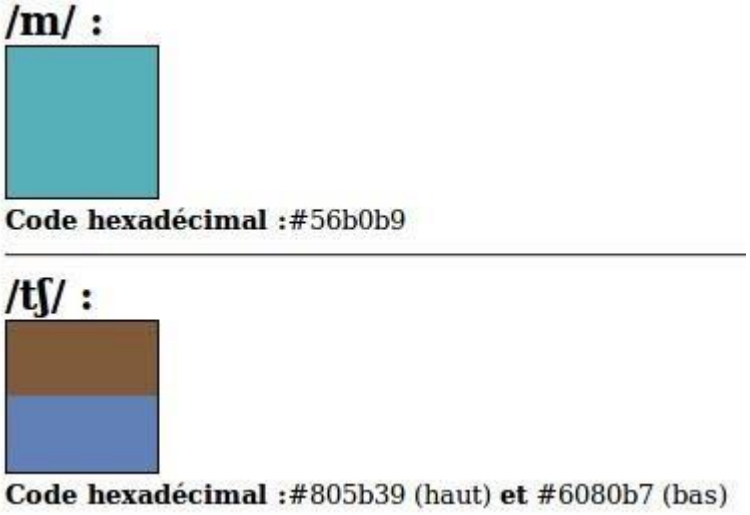
le traitement d'un texte provoque aussi l'apparition d'un bouton redirigeant vers l'affichage des phonèmes et graphies d'un texte.

b) affichage des graphies et phonèmes d'un texte :



Cette page permet simplement d'afficher les différents phonèmes et graphies correspondants au texte analysés par phonochrome depuis la page d'index. Ceci est réalisé par un script python qui prend en entrée le fichier de résultat colorisé par phonochrome et stocké dans un fichier texte.

c) affichage des ressources :



Cette partie du site permet simplement une lecture plus confortable du fichier de ressource de couleur présent dans le dossier /config du programme. Les diphtongues y sont indiquées par des carrés de couleur double.

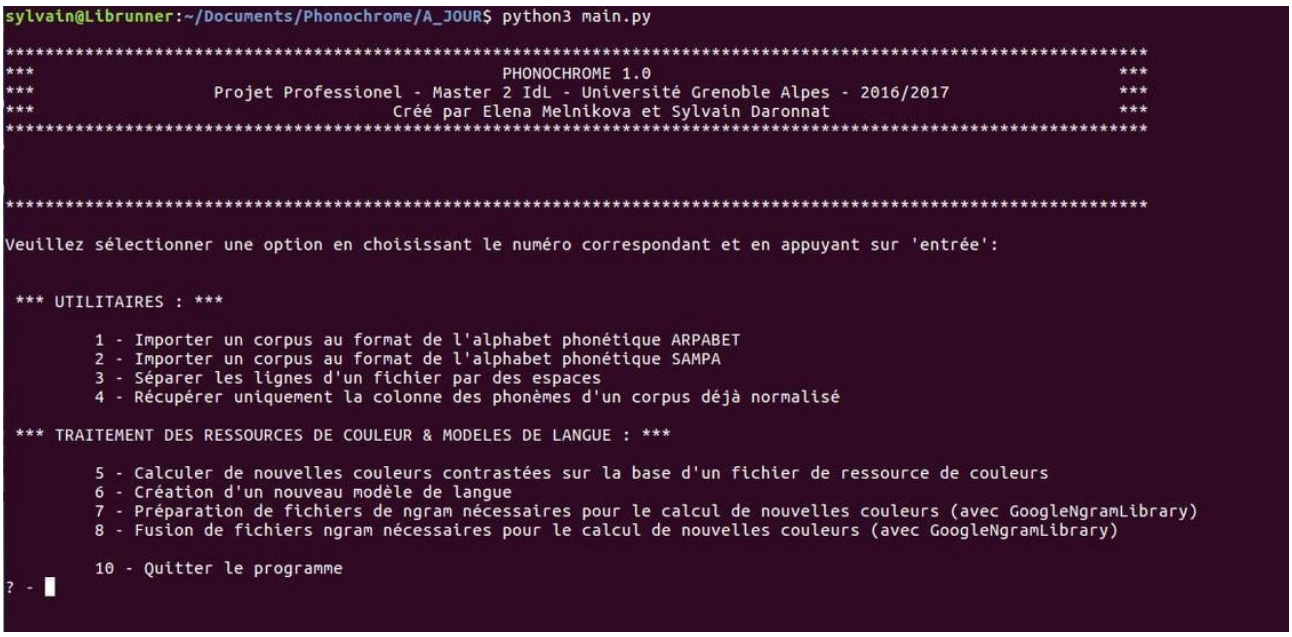
d) à propos

Cette page donne quelques informations sur l'outil, ses implications, son emplacement sur github…

III. Phonochrome : l'interface en ligne de commande

Pour tout le traitement préalable des corpus et des fichiers de ressources de couleurs et de bigram, Phonochrome propose une interface en ligne de commande écrite en Python3 qui réunit différents utilitaires créés spécialement pour Phonochrome ainsi que des traitements provenant tout droit de Phonetisaurus et de GoogleNgram library (voir partie 1 pour leur installation). Dans ce document nous allons expliquer quelles actions sont lancées par ce script et comment y parvenir sans obligatoire passer par cette interface en ligne de commande.
Conseil : utilisez au maximum le format UTF8 sans bom lors de la manipulations de fichiers avec Phonochrome, en effet l'UTF8 avec prise en charge du BOM introduit souvent des caractères invisibles néfastes au bon fonctionnement de Phonochrome.

Tout d'abord, afin de lancer le programme, veuillez tapez « Python3 main.py » dans une fenêtre de commande ouverte dans le répertoire où est installé Phonochrome. (vous y verrez uniquement des dossier et un script python 'main.py')



Les utilitaires (4 premières options du script) :

Ces programmes sont tous situés dans le dossier « tools » de Phonochrome, ils consistent un ensemble de script python3 aillant pour but de faciliter le traitement d'un corpus en entrée. Ils ne sont absolument pas nécessaire au bon fonctionnement de Phonochrome.
En effet pour que le programme fonctionne au niveau de la création d'un modèle de langue ou d'un modèle de ngram pour la recherche de contraste de couleurs (que nous aborderons plus tard) il faut que les corpus en entrée suivent le même schéma :
mot \t p h o n e m e
(où \t représente une tabulation entre le mot et sa transcription phonémique).

1. Conversion ARPA→ API

Ce script permet, à l'aide d'un fichier ressource présent dans le dossier /config, de transposer un corpus en minuscule et en API depuis l'alphabet phonétique Arpabet.

2. Conversion SAMPA → API

Effectue le même traitement mais depuis l'alphabet SAMPA adapté aux normes de lexique.org

3. Séparation des caractères

permet de séparer les caractères d'un fichier par des espaces, ce qui est particulièrement utile pour normaliser un corpus en entrée.

4. Récupération des phonèmes

Ce script permet de récupérer la « colonne » de droite d'un corpus déjà normalisé, cette « colonne » suivant notre modèle de constitution de corpus, regroupe l'ensemble des phonèmes d'un texte et sa récupération est cruciale vis à vis de la création du fichier de ressource ngram pour le calcul d'un nouveau set de couleur.

Traitement des ressources de couleurs et modèles de langue :

5. Calcul d'un nouveau jeu de couleur contrasté

Situé dans le dossier '/calcul_contrast' ce script permet de calculer un nouveau fichier de ressource de couleur sur la base d'un ancien fichier du même type. (le premier fichier de ressource a avoir servi s'étant grandement inspiré de Kinéphone dans le choix de ses couleurs).

Lors de son fonctionnement il prend en entrée un fichier de ressource de couleur au format .txt ainsi qu'un autre fichier de ressource de ngram (fichier dont nous verrons la création plus tard).
Sa sortie est un fichier de ressource de couleurs contrastées nouvellement calculé par le programme, qui est prêt à être intégré à Phonochrome une fois renommé en « ressource_couleur.txt » et inséré dans le dossier « config » de l'outil Phonochrome.

Ce script aillant était conçu pour Phonochrome, son utilisation est plutôt simple et sa chaine de traitement ne nécessite que deux modules complémentaires (colormath et webcolors). Pour plus de détails veuillez vous reporter au script en lui même et à ses commentaires.

6. Création d'un modèle de langue

Cette étape permet la création d'un modèle de langue et utilise des commandes issues du toolkit 'Google Ngram Library' et de Phonetisaurus (qui sont donc requis pour cette étape du traitement). L'entrée de cette option nécessite un corpus phonétisé selon le modèle standard de Phonochrome, à savoir : mot1 \tabulation p h o n e m e s. Ce fichier devra se trouver dans le dossier 'import_corpus' de Phonochrome. Sa sortie est un modèle de langue prêt à être utilisé par le programme. Pour cela veuillez ensuite l'insérer dans un dossier portant le même nom que le fichier (par exemple /test/test.fst si le fichier s'appelle 'test.fst') dans le dossier '/res' de Phonochrome.
Vous pourrez ensuite utiliser ce nouveau modèle directement depuis l'interface graphique de Phonochrome en le sélectionnant parmi les langues disponibles (mises à jour dynamiquement).

Voici en détail les commandes pouvant tout à fait être utilisées indépendamment de l'interface en ligne de commande Phonochrome.

1) alignement basique : ici on aligne les graphies et les phonèmes du corpus d'entrainement avec phonetisaurus :

phonetisaurus-align --input=test.train --ofile=test.corpus –seq1_del=false où test.train est le fichier

d'entrée et test.corpus le fichier de sortie.

2) avec Google Ngram on créé un fichier de ngram à partir du fichierde base estimate-ngram -

o 8 -t test .corpus -wl test.arpa où test.corpus est le fichier d'entrée et test.arpa le fichier de sortie.

3) enfin on compile le tout dans un fichier .fst de modèle de langue quiest prêt à être utilisé par Phonochrome

phonetisaurus-arpa2wfst –lm=test.arpa –ofile=test.fst où test.arpa est le fichier d'entrée et

où test.fst le fichier de sortie.

7. préparation d'un fichier au format CNTS (pour le fusionner ultérieurement)

Cette étape permet, en utilisant exclusivement Google Ngram Library, de constituer un premier fichier qui sera nécessaire à l'option 8 : la création d'un fichier de ngram au format ARPA pour le calcul de contraste de couleurs.
Ces commandes prennent en entrée un corpus qui, pour que le programme fonctionne comme prévu, doit consister uniquement en une liste de phonèmes (voir utilitaire permettant de récupérer uniquement la colonne droite d'un corpus).
En sortie ce programme produit un fichier au format CNTS nécessaire dans la création d'un nouveau fichier de ngram.

Ce script utilise les lignes de commandes suivantes :

1) création de la table de symboles (répertories les différentssymboles que le programme va utiliser:

ngramsymbols < taiwanese_pinyin.api > taiwanese_pinyin.syms où taiwanese_pinyin.api est

le fichier d'entrée et taiwanese_pinyin.syms le fichier de sortie

2) compilation des symboles :
farcompilestrings -unknown_symbol="<unk>"
-symbols=taiwanese_pinyin.syms -keep_symbols=1 taiwanese_pinyin.api > taiwanese_pinyin.far où

taiwanese_pinyin.syms et taiwanese_pinyin.api sont les fichiers d'entrée et

taiwanese_pinyin.far le fichier de sortie. **3) estimation des ngram du corpus :**

ngramcount -order=5 taiwanese_pinyin.far >taiwanese_pinyin.cnts où taiwanese_pinyin.far

est le fichier d'entrée et taiwanese_pinyin.cnts le fichier de sortie

8. Fusion des fichiers CNTS vers un fichier de statistiques NGRAM au format ARPA

Cette étape se sert des résultats du 7ème script pour fusionner des fichiers
CNTS et créer un fichier de ressource ngram au format ARPA (à ne pas confondre avec ARPAbet en phonétique) qui servira à calculer des contrastes de couleurs suivant les bigram présents dans ce fichier. Il utilise lui aussi Google Ngram Library.

l'entrée se compose de deux fichiers au format CNTS la sortie est un fichier
de ngram au format ARPA.

Explications des commandes :

1) on fusionne les deux fichiers de corpus CNTS ngrammerge fichier1.cnts fichier2.cnts >

fichier1_fichier2.merged où les entrée sont fichier1.cnts et fichier2.cnts et la sortie le

fichier1_fichier2.merged.

Attention, la commande ne permet que de mélanger deux fichiers à la fois, pour créer un fichier de ngram avec plus de 2 corpus il faut alors mélanger le dernier fichier fusionné avec un autre corpus et répéter la manipulation jusqu'au traitement de tous les fichiers.

Illustration :

fichier à mélanger : fichier1.cnts fichier2.cnts fichier3.cnts fichier4.cnts

il faut donc faire : ngrammerge fichier1.cnts fichier2.cnts > fusion1.merged
ngrammerge fusion1.merged. fichier3.cnts > fusion2.merged ngrammerge
fusion2.merged. fichier4.cnts > fusion3.merged

Ici le fichier 'fusion3.merged ' contient toutes les informations normalisés (car exprimées en pourcentages) des fichiers 1, 2, 3 et 4.

2) dès que tous les fichiers sont fusionnés, on peut alors créer lemodèle de langue correspondant : ngrammake fusion3.merged > fusion.mod où fusion3.merged est le fichier d'entrée et fusion.mod le fichier de sortie.

3) On peut donc enfin créer le fichier de ngram au format ARPA :

ngramprint --ARPA fusion.mod > fusion.ARPA où fusion.mod est le fichier d'entrée et

fusion.ARPA le fichier de sortie.

Exemple de contenu d'un fichier ngram au format ARPA :

\data\

ngram 1=384 ngram 2=15225
ngram
3=186744 ngram
4=561757 ngram
5=869708

\1-grams:
-99 <s> -0.8381258
-0.8483457</s>
-1.515879 t -2.321678

-2.070425 r	-2.456514
-2.148135 ɪ	-2.698511
-1.744444 p	-2.360304
-1.695535 ʌ	-2.452186

…