Documentation de la plateforme PFC (version 2.0)

20 Février 2009

Julien Eychenne
Simon Fraser University
jeychenne@gmail.com

La plateforme PFC est un outil dont l'objectif est d'offrir une interface unique pour le traitement et l'analyse des corpus PFC[1] : il permet de travailler sur un nombre illimité d'enquêtes, est totalement intégré dans Praat, est extensible (à l'aide de scripts Praat ou de greffons Praat/Python), supporte la gestion de métadonnées et fonctionne sous Windows, Mac OS X et Linux. L'intégration avancée avec Praat permet des va-et-vient aisés entre les résultats et le signal. Cela permet de vérifier ou d'extraire des portions du signal sonore, mais aussi d'apporter des modifications dans les TextGrids très rapidement. Le support des scripts et greffons permet par ailleurs d'étendre les fonctionnalités très simplement. La version 2.0 apporte de nombreuses améliorations dont notamment :

- une meilleure intégration dans Praat
- un support de l'encodage UTF-16 (nouvel encodage de Praat)
- un onglet Texte pour la recherche dans la tire 1
- une distinction entre fichiers physiques et fichiers logiques (via les alias)
- un support initial de l'étiquetage grammatical pour la liaison
- de meilleures performances

Par ailleurs, les performances ont été légèrement améliorées et divers bugs ont été corrigés. La plateforme est dorénavant distribuée comme une extension (plug-in) praat : de ce fait, l'installation est un peu plus compliqué sous Windows puisqu'il est dorénavant nécessaire d'installer le langage Python et ses dépendances.

1. Avant de commencer…

1.1. Pré-requis techniques

D'un point de vue matériel, il n'y a pas d'exigence particulière, bien qu'il soit recommandé de disposer d'une machine relativement récente. La vitesse d'exécution du programme dépend du volume de données, de la quantité de mémoire vive et de la vitesse du processeur.

D'un point de vue logiciel, la version de Praat installée doit être égale ou supérieure à 5.0.13 : de manière générale, il est recommandé d'utiliser la version la plus récente, car la plateforme tend à tirer profit des innovations qui apparaissent dans Praat. La version de Python doit être la version 2.5 ou 2.6. Les utilisateurs de Windows auront par ailleurs besoin des programmes praatcon.exe et sendpraat.exe (voir section Installation ci-dessous).

Il est recommandé de placer toutes vos enquêtes dans un seul dossier. Notez que tous **vos fichiers TextGrid doivent être encodés en Unicode** (UTF-16). Si ce n'est pas le cas, dans la fenêtre « Praat Objects », cliquez sur « Modify » puis « Convert to Unicode » et sauvegardez le fichier.

1.2. La notion de corpus

1 Je tiens à remercier Annelise Coquillon, Jacques Durand, Dominique Nouveau, Jean-Michel Tarrier, Gabor Turcsan et Sylvain Detey pour leurs tests et/ou commentaires qui m'ont permis d'améliorer l'outil et de clarifier la documentation.

Pour bien appréhender le fonctionnement de l'outil, il est nécessaire de comprendre qu'il ne fonctionne pas sur les TextGrids individuels : l'unité de base au sens de la plateforme est le *corpus*. Un corpus est une collection d'une ou plusieurs enquêtes PFC. Une enquête peut appartenir à plusieurs corpus et à partir de la version 2.0 il n'est pas nécessaire que toutes les enquêtes soient dans le même dossier (bien que ce soit fortement recommandé). La plateforme ne peut charger qu'un corpus à la fois : au démarrage, elle construit une représentation arborescente de ce corpus et la parcourt à chaque requête.

Par défaut, le programme considère que votre corpus est un dossier nommé « Corpus » (attention à la majuscule) situé dans un dossier nommé « PFC », lui-même situé dans votre dossier personnel. Ainsi, pour un utilisateur « Julien », le corpus par défaut sera :

`C:\Documents and Settings\Julien\PFC\Corpus`	(sous Windows)
`/Users/Julien/PFC/Corpus`	(sous Mac OS X)
`/home/julien/PFC/Corpus`	(sous Linux)

Ces valeurs peuvent être modifiées (cf. section Utilisation).

2. Installation

Etant donné le nombre de versions de Python et de wxPython (sa bibliothèque graphique), quelques remarques générales s'imposent. La plateforme PFC est prévue pour fonctionner avec les versions 2.5 et 2.6. Le langage python a récemment rendu publique sa version 3.0, qui introduit des changements majeurs dans le langage et qui par conséquent requiert des modifications significatives pour que les programmes puissent fonctionner avec cette nouvelle version. La version 2.6 de Python est destinée à faciliter la transition de 2.5 à 3.0. Sauf cas particulier, nous recommandons fortement l'installation de la version 2.6.

La bibliothèque wxPython possède son propre numéro de version, indépendant de python. A l'heure actuelle, il s'agit de la version 2.8.x.x (où les « x » représentent des numéros de version mineurs). Cette version existe en plusieurs « saveurs », en fonction des différentes versions de python, ce qui peut provoquer une certaine confusion pour le profane. Quel que soit votre système d'exploitation, il faut télécharger wxPython 2.8.x.x compilé pour python 2.6 en version Unicode (et non Ansii). Il en va de même pour PyWin32 sous Windows.

2.1. Installation sous Windows

La plateforme PFC requiert l'installation préalable de l'interpréteur Python et de ses dépendances. Python peut être téléchargé à partir de http://www.python.org/download/. Il faut choisir la version « Python 2.6.1 Windows installer ». Une fois Python installé, il faut installer l'extension PyWin32 que l'on peut récupérer à partir de http://sourceforge.net/projects/pywin32/. Là encore, on veillera à télécharger la version qui correspond à Python 2.6. Enfin, il faut installer wxPython (une bibliothèque offrant une interface graphique pour Python) à partir de :

http://www.wxpython.org/download.php#binaries

Il faut télécharger la version win32-unicode pour Python 2.6.

Une fois Python et ses dépendances installés, créez un dossier « Praat » à la racine du disque C (chemin : `C:\Praat`), dans lequel il faudra télécharger Praat et les programmes `pratcon.exe` et `sendpraat.exe`. Ces programmes peuvent être obtenus aux adresses suivantes :

- Praat et Praatcon : http://www.fon.hum.uva.nl/praat/download_win.html
- sendpraat : http://www.fon.hum.uva.nl/praat/sendpraat.exe

Praatcon et sendpraat sont des outils en ligne de commande qui permettent à la plateforme PFC de communiquer avec Praat.

Au démarrage, le programme vérifiera la présence de Praat (et de Praatcon) et proposera soit de préciser le dossier où ils se trouvent s'ils ne sont pas dans `C:\Praat`, soit de les télécharger et de les installer pour vous. L'outil `sendpraat.exe` doit en revanche être téléchargé manuellement.

2.2. Installation sous Mac OS X

La plateforme PFC est prévue pour fonctionner sous Mac OS X 10.3 (ou « Panther »), 10.4 (ou « Tiger ») et 10.5 (ou « Leopard »). Il faut tout d'abord installer Praat dans le dossier Applications. Il faut ensuite installer Python et wxPython que l'on récupèrera aux adresses suivantes :

- Python : http://python.org/ftp/python/2.6.1/python-2.6.1-macosx2008-12-06.dmg
- wxPython : http://www.wxpython.org/download.php#binaries

Pour ce dernier, il faut installer la version « osx-unicode » pour Python 2.6. Récupérez ensuite sendpraat à partir de http://www.fon.hum.uva.nl/praat/sendpraat.html. Téléchargez `sendpraat_ppc` si votre mac est un mac PPC (G4 ou G5) et `sendpraat_intel` s'il s'agit d'un mac Intel (les plus récents). Renommez le fichier en « `sendpraat` » sans extension et placez-le sur le bureau. Ouvrez ensuite l'application terminal (dans « Applications » > « Utilitaires ») et tapez les deux commandes suivantes (chacune suivie de la touche entrée) :

```
chmod +x ~/Desktop/sendpraat
sudo mv ~/Desktop/sendpraat /usr/local/bin
```

La deuxième commande vous demandera votre mot de passe utilisateur. Vous pouvez vérifier que sendpraat a bien été installé en tapant la commande « `which sendpraat` » qui retournera « `/usr/local/bin/sendpraat` ».

Une fois Python installé, décompressez l'archive plugin_Plateforme_PFC.zip, ceci créera un dossier plugin_Plateforme_PFC.

2.3. Installation sous Linux

Vous devez installer Python 2.5 ou 2.6 et wxPython (consultez l'aide de votre distribution).

Il est nécessaire de compiler le fichier sendpraat à partir du fichier source (récupérable sur le site de praat). Pour ce faire, il faut disposer d'un compilateur GCC et des fichiers headers de X11 ou X.org. Sur Ubuntu (version 8.10), la commande pour compiler sendpraat est :

```
gcc –o sendpraat –L/usr/share sendpraat.c –lX11
```

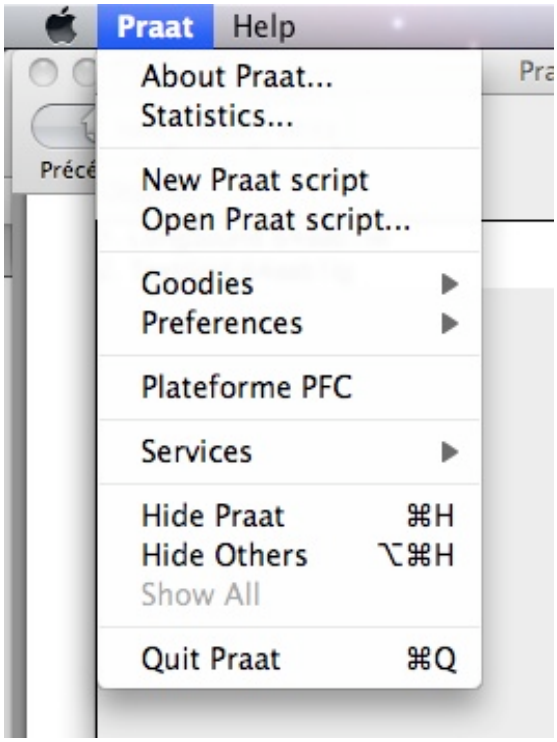
Par défaut, la plateforme suppose que Praat est dans `/usr/bin/` et que sendpraat est dans `/usr/local/bin/`. En cas de problème, contactez l'auteur en indiquant la distribution que vous utilisez (par exemple Ubuntu 8.10).

Déplacez ensuite le dossier `plugin_Plateforme_PFC` dans le dossier `~/.praat-dir`.

3. Prise en main et utilisation

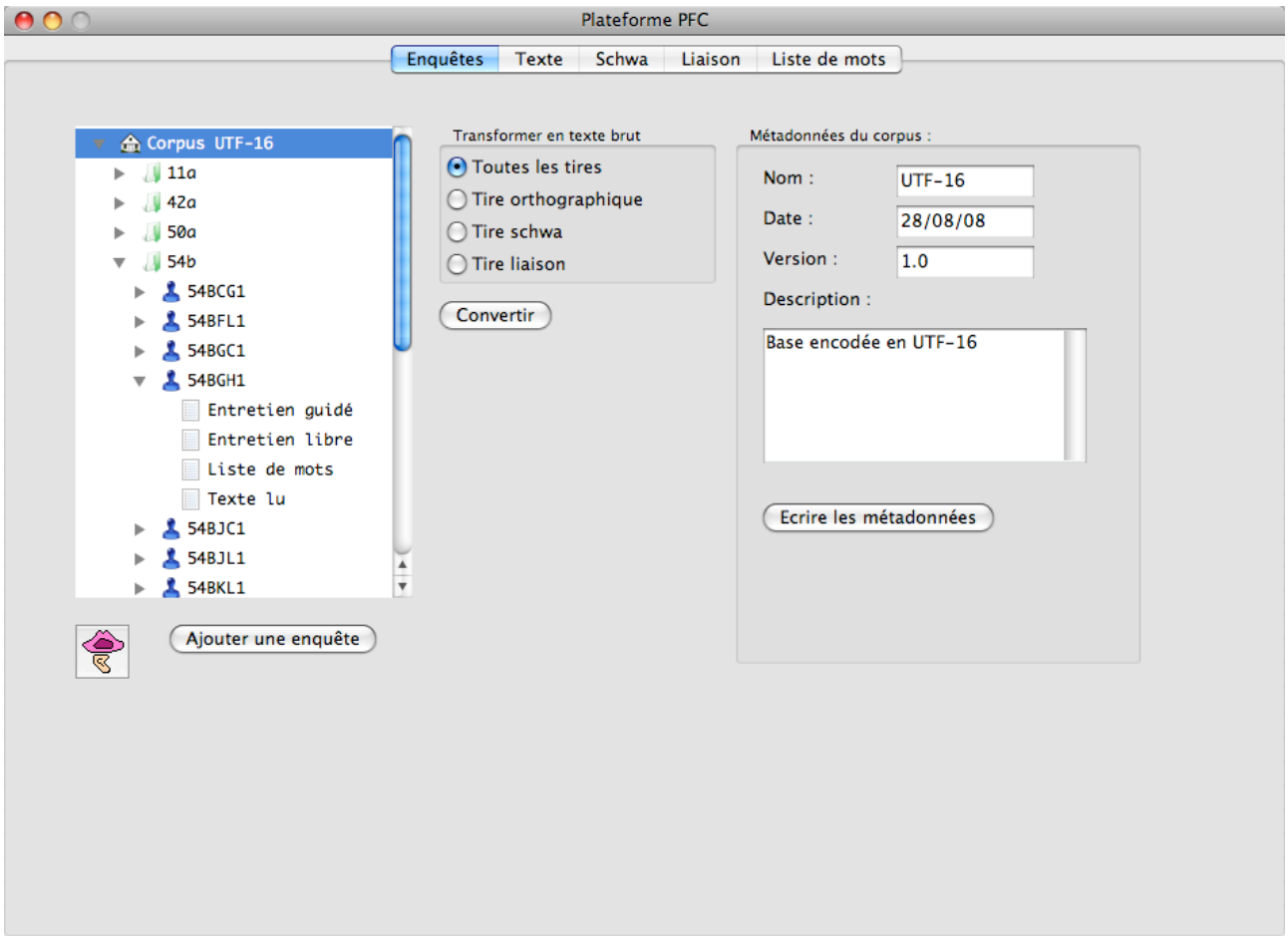
Nous avons vu plus haut que l'unité de base de la plateforme PFC est le *corpus*, c'est-à-dire est un ensemble d'une ou plusieurs enquêtes. Un corpus est en réalité un dossier contenant des enquêtes ou des liens vers des enquêtes. Vous pouvez avoir autant de corpus que vous le souhaitez, et une enquête peut appartenir à plusieurs corpus. A partir de la version 2.0, il est recommandé de stocker toutes vos enquêtes dans un même dossier : nous appellerons ce dossier le « dossier physique ».

Pour lancer la plateforme PFC, ouvrez Praat et dans le menu Praat, cliquez sur le bouton « Plateforme PFC » comme dans la capture d'écran ci-dessous :



Au démarrage, le programme recherchera le corpus par défaut : s'il n'existe pas, il proposera une boîte de dialogue vous permettant de sélectionner le corpus à charger. Il est possible de charger votre dossier physique, mais il est recommandé de créer pour commencer un corpus vide : il suffit de créer un dossier vide et d'indiquer ce dossier à la plateforme : elle s'ouvrira alors avec un corpus vide (voir capture d'écran ci-après). Dans la partie gauche, cliquez sur le bouton « Ajouter une enquête » et sélectionnez l'enquête que vous souhaitez ajouter. Répétez l'opération pour toutes les enquêtes que vous voulez ajouter à ce corpus[2], puis redémarrez la plateforme. Si tout s'est bien passé, une fenêtre comme celle qui suit apparaîtra :

[2] Pour toute enquête que vous ajoutez à l'aide de ce bouton, la plateforme créera un fichier « alias »



Comme on le voit sur cette capture d'écran, la fenêtre principale est constituée de cinq « onglets » respectivement intitulés « Enquêtes », « Texte », « Schwa », « Liaison » et « Liste de mots », onglets que nous aborderons successivement. L'onglet par défaut est l'onglet « Enquêtes ». Les utilisateurs de Windows remarqueront qu'au lancement, une console noire apparaît en fond : elle est utilisée pour communiquer avec Praat et ne doit jamais être fermée car cela provoquerait l'arrêt du programme.

3.1. L'onglet « Enquêtes »

La partie gauche de l'onglet offre une vue arborescente du corpus : il est organisé en dossiers « enquêtes » et sous-dossiers « locuteurs » de la même manière que les enquêtes sur votre disque dur. En revanche, les dossiers locuteurs ne montrent pas tous les fichiers (sons et TextGrids) présents, mais plutôt les tâches : ainsi, dans la fenêtre ci-dessus, au lieu de montrer les fichiers `54bgh1tg.TextGrid` et `54bgh1tw.wav`, l'outil présente une tâche « Texte lu ». Il est possible de double-cliquer dessus ou de cliquer sur l'icône Praat (en bas à gauche) pour ouvrir les fichiers son et TextGrid correspondant à cette tâche directement dans Praat.

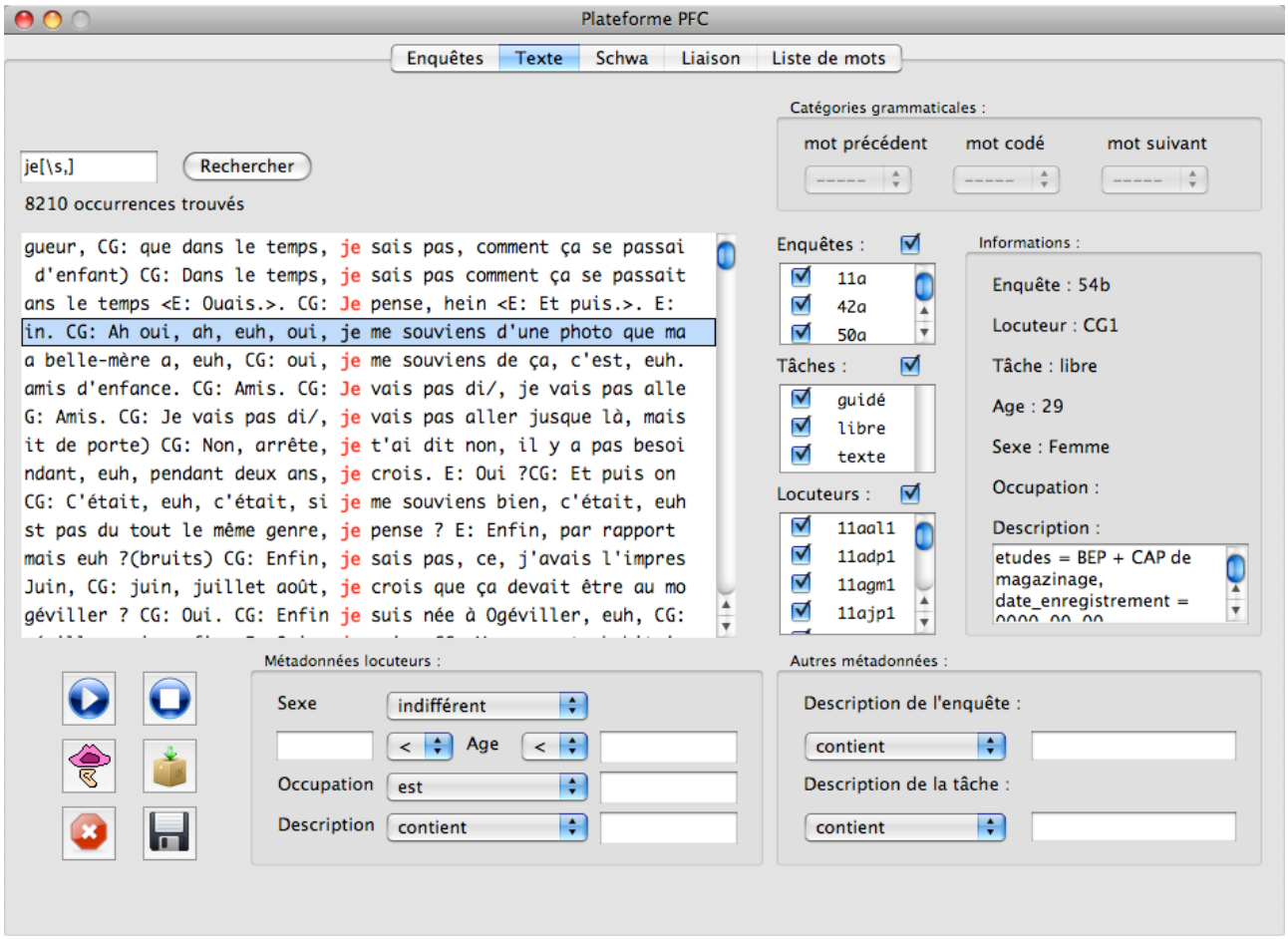
La partie centrale de l'onglet (« Transformer en texte brut ») permet de transformer un TextGrid en texte simple qui peut être lu dans n'importe quel éditeur, traitement de texte (par exemple Word) ou concordancier . Pour l'utiliser, sélectionnez une tâche dans la partie gauche, puis choisissez la tire que vous souhaitez convertir (toutes par défaut), et enfin cliquez sur le bouton « convertir ». Ceci ouvrira une nouvelle fenêtre contenant le texte converti : vous pouvez l'enregistrer dans un fichier texte (format TXT) en cliquant dans le menu sous `Fichier > Enregistrer sous...`

Enfin, la partie droite de l'onglet permet d'afficher et de modifier les métadonnées. Les métadonnées sont des « données sur les données ». La plateforme PFC supporte 4 niveaux de métadonnées : le

corpus, l'enquête, le locuteur et la tâche. Chaque niveau possède un ensemble de métadonnées spécifiques, et tous possèdent un champ libre « Description » qui permet à l'utilisateur d'intégrer ses propres commentaires et notes de travail. On peut par exemple cataloguer telle ou telle enquête, comme « Nord » ou « Canada », tel ou tel locuteur comme « conservateur » ou « innovateur », etc. Un point particulièrement intéressant est qu'il est possible de saisir des caractères : on peut donc utiliser les symboles de l'API et stocker des commentaires du type « opposition e/ɛ absente ». Les métadonnées[3] peuvent ensuite être utilisées dans les autres onglets pour restreindre les requêtes.

3.2. L'onglet « Texte »

L'onglet texte est tout simplement un concordancier pour la tire 1. On peut chercher une chaîne de caractères quelconque[4] et il est possible d'utiliser la syntaxe des expressions régulières. L'onglet se présente comme dans la fenêtre ci-dessous :



L'onglet se décompose en plusieurs parties : en haut à gauche se trouve un champ de saisie qui permet d'entrer le motif à rechercher : dans notre exemple, nous recherchons le mot *je* suivi d'un espace ou d'une virgule, ce qui correspond au motif « `je[\s,]` ». Pour exécuter la requête, il suffit d'appuyer sur la touche « Entrée » ou sur le bouton « Rechercher ». Lorsque la requête est exécutée, le programme affiche toutes les occurrences du motif pour les critères sélectionnés dans la liste déroulante sous le bouton de recherche : le motif est présenté en rouge gras avec ses cotextes

[3] Les métadonnées concernant le corpus et les enquêtes sont stockées dans des fichiers à la racine des dossiers corpus et enquêtes correspondants. Les métadonnées concernant le locuteur et les tâches sont stockées dans un sous-dossier « data » contenu dans le dossier locuteur. Les métadonnées sont gérées automatiquement par la plateforme, et l'utilisateur n'a normalement pas à s'en préoccuper. Le format de stockage est XML et l'encodage UTF-8.

[4] Notez que la plateforme n'est pas sensible à la casse.

gauche et droit. On peut naviguer dans les résultats à l'aide de la molette de la souris. Chaque fois qu'un motif est sélectionné, un certain de métadonnées sont affichées dans le cadre « Informations » de la partie droite. Ceci permet de savoir quel est le locuteur qui a réalisé l'occurrence.

Entre la liste de résultats et le cadre « Informations » se trouvent 3 listes déroulantes, à savoir « Enquêtes », « Tâches » et « Locuteurs ». Pour sélectionner ou désélectionner une enquête, une tâche ou un locuteur spécifique, il suffit de cocher/décocher la petite case qui le précède. Pour sélectionner/désélectionner toutes les enquêtes (ou tous les locuteurs), on utilisera les cases qui suivent les labels « Enquêtes » et « Locuteurs » respectivement. La liste des locuteurs contient tous les locuteurs des enquêtes cochées : elle est mise à jour chaque fois qu'une enquête est cochée ou décochée. On peut donc avoir un contrôle très fin sur les recherches que l'on veut effectuer, et l'on peut par exemple aisément grouper « libre » et « guidé » (en décochant « texte ») pour comparer les résultats de la conversation à ceux de la lecture de texte.

Dans la partie inférieure de l'onglet, se trouvent deux cadres relatifs aux métadonnées : il permettent de saisir des critères de recherche de métadonnées afin de filtrer les enquêtes, locuteurs et tâches. On peut filtrer les requêtes en fonction des critères « Sexe » (homme ou femme), « Age » (valeur numérique), « Profession » (champ libre), mais également « Description » (champ libre). Ce dernier s'avère particulièrement flexible puisqu'on peut y saisir (et y chercher) des chaînes de texte arbitraires[5]. Par exemple, si l'on étiquette les enquêtes du sud de la France comme « Midi », on peut effectuer une requête pour tous les locuteurs de toutes les enquêtes SAUF ceux qui appartiennent à des enquêtes étiquetées « Midi ». Dans les champs « Profession » et « Description », il est possible de chercher plusieurs valeurs, en les séparant par un point-virgule (sans espaces). Ceci correspond à l'opérateur logique OU, c'est-à-dire qu'il renvoie tous les enquêtes/locuteurs/tâches (en fonction de la requête) qui contiennent l'une des valeurs saisies. Notez enfin que pour réinitialiser les critères de recherche, il suffit de cliquer sur le dernier bouton, en bas à gauche.

Pour écouter une occurrence, il suffit de la sélectionner (il apparaît alors en bleu) et de cliquer sur le bouton « jouer » (le premier des 6 boutons en bas à gauche). On peut également double-cliquer dessus. Dans les deux cas, la plateforme jouera l'intervalle Praat dans lequel le codage se situe. On peut l'interrompre en appuyant sur le bouton « stop » à côté du bouton « jouer ». Lorsqu'un codage est sélectionné, il est également possible de l'ouvrir dans Praat : il suffit pour ce faire de cliquer sur le troisième bouton avec l'icône de Praat. Ceci aura pour effet de sélectionner les fichiers son et TextGrid du locuteur et de les ouvrir directement sur l'intervalle dans lequel le codage se trouve. Cela permet entre autres de vérifier la cohérence des codages à grande échelle et éventuellement de les modifier de manière particulièrement efficace[6]. Le quatrième bouton permet d'enregistrer la sélection au format Collection de Praat : il s'agit d'un format binaire qui fusionne le son et la transcription dans un seul fichier. Le cinquième bouton permet de réinitialiser tous les champs de recherche dans les cadres de métadonnées. Enfin, le sixième bouton permet d'exporter la liste d'occurrences au format CSV (encodage UTF-16, champs séparés par un point-virgule). Ces fichiers peuvent être importés dans un tableur de type Excel[7].

Lorsque vous utilisez des critères de sélection dans les métadonnées, les locuteurs pour lesquels il n'existe pas de métadonnées pour l'un de vos critères de recherche seront ignorés. Les locuteurs ignorés sont consignés dans un fichier nommé `missing_metadata.txt` dans votre dossier « Préférences » (voir §3.7). Ce fichier est automatiquement mis à jour (ou effacé) à chaque requête.

---

[5] Les caractères « < » et « > » sont illicites. De même, « ; » est utilisé pour séparer des valeurs.

[6] Notez toutesfois que si vous modifiez des fichiers TextGrid, il vous faudra relancer la plateforme pour que vos modifications soient prises en compte.

[7] Il semble qu'à l'heure actuelle Excel ne gère pas (ou mal) l'encodage UTF-16. On peut utiliser le tableur de la suite OpenOffice.org ou bien utiliser un éditeur de texte élaboré pour changer l'encodage du fichier CSV puis l'importer dans Excel.

En dernier lieu, il faut signaler que le cadre en haut à droite permet d'effectuer des recherches par catégories grammaticales. A l'heure actuelle, l'étiquetage grammatical[8] ne fonctionne que pour la liaison (et pour un nombre limité d'enquêtes). Il est pour l'instant désactivé et non documenté.

3.2 L'onglet « Schwa »

L'onglet Schwa est rigoureusement identique à l'onglet Texte. Du point de vue des possibilités de requêtes, le moteur de recherche inclut quelques facilités pour permettre des recherches ne se limitant pas à un seul codage. On peut utiliser le caractère « * » pour remplacer n'importe quel chiffre. Ainsi, on pourrait chercher le code « *412 » qui renverrait les codes « 0412 », « 1412 » et « 2412 » (non triés). L'étoile peut être utilisée dans n'importe quelle position du code. Pour grouper plusieurs chiffres pour une position, on les place entre crochets : par exemple, le motif « 01[23]2 » renvoie tous les monosyllabes sans réalisation vocalique, précédés d'une consonne ou d'une frontière intonative (mais pas d'une voyelle) et suivis d'une consonne. De même, on peut grouper les chiffres 3 et 4 en quatrième position (frontières forte et faible), par exemple 141[34] soit tous les mots qui ont une réalisation vocalique (1) en fin de polysyllabe (4) après une suite VC (1) et devant frontière forte (3) ou faible (4). Etant donné qu'il est fréquemment nécessaire de grouper les frontières droites forte et faible (XXX3 et XXX4), on peut également utiliser, pour la position 4 uniquement, le caractère « # » à la place de [34] : ainsi, le motif « 141# » n'est rien d'autre qu'un raccourci pour « 141[34] ». Le dernier caractère spécial que reconnaît le moteur de recherche schwa est le « X » majuscule qui correspond à tous les caractères alphabétiques sauf « e ». Ceci permet de tester très facilement les corrélations graphie/phonie : on sait que certaines variétés (méridionales) conservent une opposition phonologique de type /mɛr/ ~ /mɛrə/ (*mer* ~ *mère*) qui n'existe plus dans la majorité des usages (/mEr/ dans les deux cas). Pour tester ce type de corrélation, on analysera d'abord les motifs du type « e*4** », à savoir la lettre graphique « e » (par exemple *maire1412 de*), puis les motifs du type « X*4** » (cf. *vingt1412-six*). Signalons que la taille du motif n'est pas limitée, et l'on pourrait par exemple chercher « che****mise****s » pour étudier les diverses réalisations du mot *chemise*. Comme dans l'onglet « Texte », la recherche n'est pas sensible à la casse et un motif tel que « je1132 » renverra aussi bien « je1132 » que « Je1132 ».

3.3 L'onglet « Liaison »

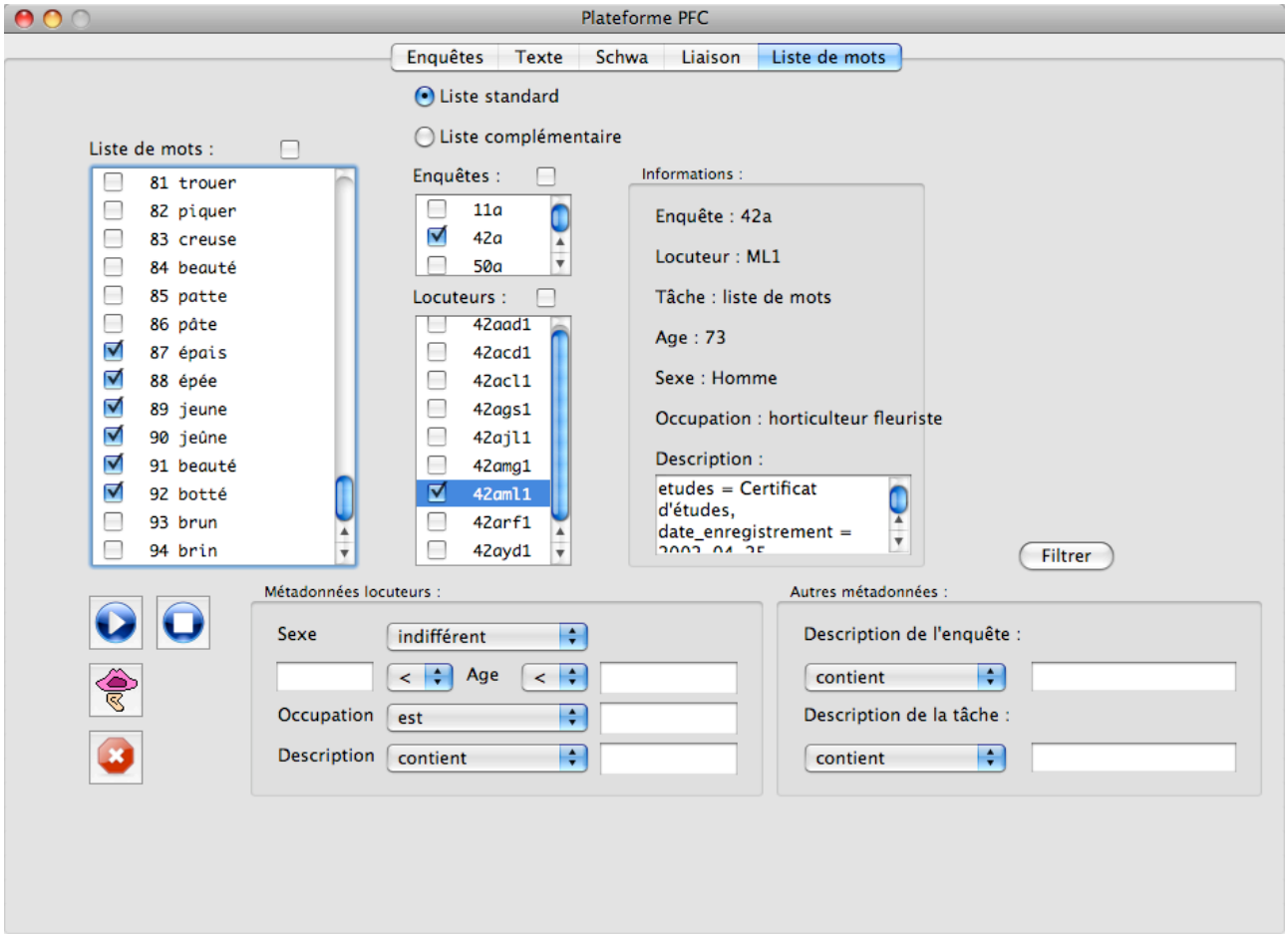
L'onglet « Liaison » est identique aux onglets « Texte » et « Schwa », à ceci près qu'il fonctionne avec des codages liaison au lieu de codages schwa (il analyse donc la tire 3 et non la tire 2). Nous renvoyons par conséquent aux explications de l'onglet « Texte » pour les aspects communs.

Le moteur de recherche n'offre que 2 caractères spéciaux : l'étoile « * », qui permet de rechercher n'importe quel chiffre (position 1 ou 2 du code) et le « C » majuscule, qui correspond à n'importe quelle consonne de liaison. Par exemple, le motif « *3C » renverra tous les mots, monosyllabiques ou polysyllabiques (*), présentant une liaison non enchaînée (3) avec n'importe quelle consonne (C).

3.4 L'onglet « Liste de mots »

Cet onglet se présente comme dans la fenêtre ci-dessous :

[8] L'étiquetage grammatical utilise le TreeTagger (http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/).



La liste déroulante la plus à gauche présente la liste de mots (standard ou complémentaire), alors que les listes à sa droite présentent les enquêtes disponibles et les locuteurs choisis. La sélection s'effectue comme pour les onglets schwa et liaison, et l'on utilisera les cases qui suivent les labels des listes pour sélectionner/désélectionner tous les items d'une liste donnée. Une fois les locuteurs et les mots sélectionnés, il suffit de cliquer sur le bouton « Jouer » le premier, en bas à gauche, pour jouer les mots les uns à la suite des autres. On peut sélectionner plusieurs mots pour un seul locuteur, un seul mot pour plusieurs locuteurs, ou même plusieurs mots pour plusieurs locuteurs, auquel cas le programme joue tous les mots du premier locuteur, puis passe au locuteur suivant, et ainsi de suite. Cette fonctionnalité permet notamment de comparer les paires minimales de façon aisée. On peut également ouvrir un mot dans Praat en utilisant le bouton Praat : si plusieurs mots et/ou locuteurs sont sélectionnés, c'est le premier mot pour le premier locuteur qui sera sélectionné.

La plateforme génère automatiquement la liste de mots en analysant le TextGrid « liste de mots » du premier locuteur de la première enquête. On notera par ailleurs qu'il n'est pas nécessaire que la liste d'un locuteur contienne exactement 94 intervalles[9] : le programme ne conserve que les intervalles commençant par un nombre, et élimine les intervalles vides ou ne contenant que du texte. S'il y a une liste complémentaire, elle est également chargée et l'utilisateur peut naviguer entre les deux. Il n'est cependant pas possible d'inclure des enquêtes ayant des listes complémentaires différentes dans le même corpus : seul la première liste complémentaire sera chargée.

3.5 Tâches complémentaires

La plateforme gère les tâches complémentaires et celles-ci sont automatiquement chargées si elles

9 Rappelons néanmoins que ceci est requis par les conventions du projet.

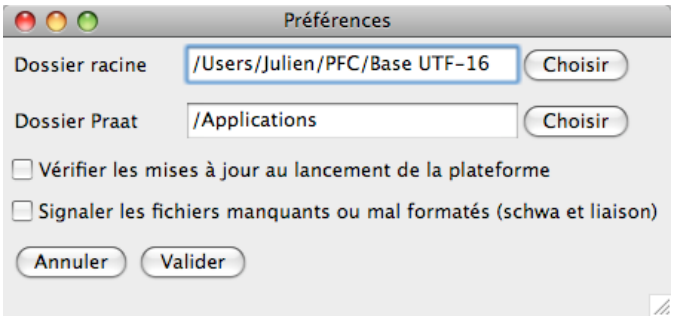
sont détectées. Toutefois, pour qu'elles soient détectées, elles doivent obéir au format suivant : pour signaler qu'un fichier est une tâche complémentaire, on ajoutera un « x » à la fin de la base du nom de fichier. Ainsi, `11atg1mgx.TextGrid` désignerait le fichier TextGrid de la liste de mot (« m ») complémentaire (« x ») du locuteur 11atg1. Le fichier WAV correspondant serait `11atg1mwx.wav`. Les tâches qui ne respectent pas ce format ne seront pas reconnues.

3.6 Sauvegarde de l'état du corpus

Lorsque l'on travaille sur un corpus volumineux, le temps de chargement des enquêtes peut être assez long. La plateforme offre donc une option de sauvegarde de l'état du corpus (dans le menu `Fichier > Enregistrer l'état du corpus`) : cette fonctionnalité crée un fichier `corpus.data` à la racine du corpus, qui est une copie exacte du corpus tel qu'il est stocké en mémoire. Au redémarrage, la plateforme n'a plus besoin de réanalyser tous les fichiers et se contente de charger le fichier `corpus.data`. Ceci accélère sensiblement le chargement des enquêtes. En revanche, si vous effectuez des modifications (ex : correction de codage, ajout de métadonnées), il faut ré-enregistrer l'état du corpus car l'opération n'est jamais automatique. Si vous souhaitez supprimer un état précédemment enregistré, il suffit de supprimer le dossier corpus

3.7 Réglage des préférences

Si pour une raison ou pour une autre vous souhaitez utiliser des réglages non standards, vous pouvez ajuster les paramètres dans le menu `Édition > Préférences...` (ou `Python > Preferences` sous Mac). La fenêtre des préférences se présente comme suit sous Windows :



Le « dossier racine » est le dossier du corpus par défaut. Ce chemin peut contenir des espaces et des caractères accentués sous Mac. Le dossier « Praat » indique le chemin vers Praat (et vers Praatcon et sendpraat sous Windows). Il est **essentiel** que ce chemin ne contienne pas d'espace.

Le bouton « vérifier les mises à jour au lancement de la plateforme » est particulièrement explicite : si cette option est activée, le programme vérifiera à chaque lancement du programme si une version plus récente est disponible en ligne et vous avertira le cas échéant. Vous pouvez également choisir de vérifier manuellement par le menu `Aide > Vérifier les mises à jour`.

La dernière option permet de choisir si l'on veut que la plateforme signale les fichiers problématiques. Cette option est activée par défaut pour attirer l'attention de l'utilisateur sur les éventuels problèmes, mais il est recommandé de la désactiver lorsqu'on sait que certains fichiers sont manquants (par exemple, une enquête n'a pas d'entretien guidé) et que l'on veut malgré tout utiliser l'outil.

Pour information, les préférences sont stockées dans un fichier `preferences.xml`, lui-même situé dans un dossier « Préférences » dont le chemin est, en fonction de la plateforme et selon l'utilisateur :

```
C:\Documents and Settings\Julien\PFCPrefs        (Windows)
/Users/Julien/Library/Preferences/PFCPrefs       (Mac OS X)
/home/julien/.PFCPrefs                           (Linux)
```

Sous Windows, il s'agit d'un dossier caché : s'il n'est pas visible, allez dans `Panneau de configuration > Options des dossiers >` Onglet `Affichage` > cochez `Afficher les fichiers et dossiers cachés`.

Après toute modification des préférences, il est recommandé de relancer la plateforme PFC.

3.8. Import de métadonnées

Il est possible d'importer les métadonnées disponibles sur le site PFC dans votre corpus. Ces métadonnées sont générées par Atanas Tchobanov. Les métadonnées doivent être stockées dans un fichier CSV encodé en UTF-8 (contacter Julien Eychenne si besoin). Pour importer ces métadonnées, il suffit d'aller dans le menu `Fichier > Import > Métadonnées…` et de sélectionner le fichier CSV. Il est nécessaire de relancer la plateforme pour que les changements soient pris en compte.

3.9. Archiver un corpus

La fonction « `Archiver un corpus…` » (accessible depuis le menu `Fichier`) permet de créer une archive ZIP de tous les fichiers TextGrid et métadonnées contenues dans un dossier, que la plateforme vous demandera de choisir. La structure arborescente du dossier en dossiers et sous-dossiers est également préservée. Cette fonction permet donc de faire des sauvegardes régulières des données textuelles des enquêtes.

4. Le menu « Scripts »

La plateforme PFC offre un menu « Scripts » extensible qui permet d'étendre la plateforme de manière très simple à l'aide de scripts Praat et éventuellement Python. La première sous-section décrit ces fonctionnalités du point de vue de l'utilisateur, les deux sous-sections suivantes, plus techniques, abordent l'écriture de scripts et greffons pour les utilisateurs qui souhaitent étendre la plateforme.

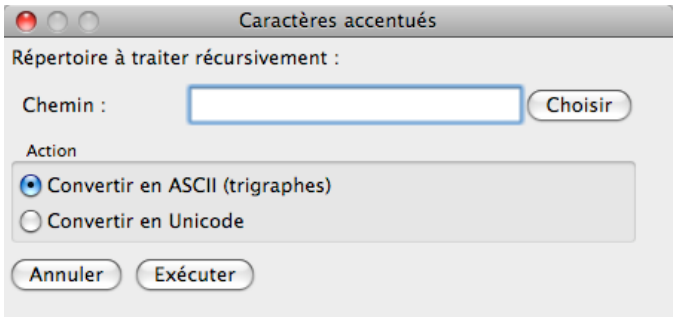
4.1 Généralité sur les extensions

Les scripts offrent des fonctionnalités additionnelles et permettent de traiter un corpus ou sous-corpus en une seule opération. Pour l'heure, trois extensions sont fournies en standard : « Caractères accentués », « Vérifier l'alignement des frontières » et « Compresser les fichiers son »[10]. Les extensions peuvent être des scripts Praat ou des greffons (ou *plugins*), ces derniers étant distribués au format ZIP. Pour importer un script ou un greffon dans votre bibliothèque, allez dans le menu `Fichier > Import` et sélectionnez le script ou le greffon que vous souhaitez importer, puis validez. Au prochain démarrage, celui-ci apparaîtra dans le menu « Scripts ». Notez qu'un script doit impérativement porter l'extension « `.praat` » et un greffon l'extension ZIP (voir § 5.2 et 5.3 pour les détails techniques).

Le script « Caractères accentués » permet de convertir les caractères accentués en trigraphes spécifiques à Praat (encodage ASCII) ou en Unicode (UTF-16). La plateforme PFC suppose en

10 Cette extension convertit les fichiers son au format FLAC (Free Lossless Audio Codec), un format audio compact sans perte d'information.

effet que vos TextGrids sont génériques, ceci afin d'éviter les problèmes de conversion entre Mac et PC par exemple. Pour généricer une enquête, au premier lancement de la plateforme, cliquez sur « Annuler » lors de la sélection du dossier, puis allez dans le menu `Scripts > Caractères accentués...` Un formulaire ressemblant à celui-ci apparaîtra :



En cliquant sur le bouton « Choisir », on sélectionne le dossier que l'on souhaite traiter (tous les sous-dossiers sont également parcourus). On choisit ensuite l'opération que l'on souhaite accomplir (pour supprimer les accents, il s'agit de « Convertir en ASCII (trigraphes) ») et l'on clique sur « Exécuter » pour lancer le script. Une fois exécuté, un message apparaît pour notifier l'utilisateur de la réussite de l'opération, après quoi le formulaire du script se referme.

Le greffon « Vérifier l'alignement des frontières » fonctionne sur le même principe et permet de vérifier que les frontières sont bien alignées sur toutes les tires (dans les TextGrids autres que les listes de mots). De manière plus générale, tous les scripts demandent normalement la sélection d'un dossier et la saisie éventuelle d'un certain nombre de paramètres.

4.2 Ecriture de scripts (Praat)

Cette section aborde l'écriture de scripts pour la plateforme PFC d'un point de vue technique : elle suppose une certaine familiarité avec le langage de scripts de Praat (voir l'aide en ligne de Praat à la section « Scripting tutorial » pour une introduction).

Les scripts PFC sont en réalité des scripts Praat qui suivent un certain nombre de règles d'écriture. Avant de les exposer, il est important de souligner que tous les scripts Praat ne peuvent être intégrés à la plateforme. Praat est en lui-même extensible, et dans la plupart des cas on préfèrera intégrer un script Praat dans Praat lui-même plutôt que dans la plateforme PFC (par exemple, un script qui copierait une partie d'une tire dans une autre tire). La plateforme PFC permet simplement de simplifier l'écriture de scripts destinés à traiter un corpus, une enquête ou un locuteur : le client (le script) n'a qu'à implémenter le code pour le traitement d'un seul fichier (TextGrid ou Wav) ou d'un couple TextGrid/Wav, et la plateforme prend en charge l'application du script à tous les fichiers contenus dans le répertoire sélectionné par l'utilisateur.

Un script doit se trouver soit dans le dossier `scripts` de la plateforme (par exemple : `C:\Program Files\Plateforme PFC\scripts`) ou dans le sous-dossier `scripts` du dossier « préférences » de l'utilisateur. Ces dossiers, s'ils existent, seront analysés à chaque démarrage. Mais pour qu'il soit reconnu comme valide, un script doit contenir certains commentaires spéciaux. Nous donnons ici un exemple avec le contenu du script `gennat.praat` (qui correspond à l'entrée « Caractères accentués... ») :

```
# Généricier ou nativiser les TextGrids d'une enquête

# Les lignes suivantes sont des commentaires spéciaux
# $INCLUDE = True
# $LABEL = "Caractères accentués..."
```

```
# $EXTENSION = "TextGrid"
# $SHORT_DESCRIPTION = "Généricizer ou nativiser un fichier"
# $ACCESS_KEY = G

form Genericize/Nativize
      comment Répertoire à traiter récursivement
            sentence textgrid /home/julien/Desktop/Gennat
      choice Action: 1
            button Genericize
            button Nativize
endform


Read from file... 'textgrid$'
if action = 1
      Genericize
else
      Nativize
endif

Write to text file... 'textgrid$'
```

Les commentaires spéciaux sont de la forme `$VARIABLE = valeur` : ils sont ignorés par Praat mais sont interprétés par la plateforme PFC. Les deux variables obligatoires sont `$INCLUDE` et `$LABEL` : la variable `$INCLUDE`, si elle a la valeur « True », indique que le fichier doit être inclus dans le menu « Scripts » ; la variable `$LABEL` contient le texte sous lequel le script apparaît dans le menu (en l'occurrence, « Caractères accentués... »). La variable `$EXTENSION` définit le type de fichier auquel le script s'applique : chaque fois qu'un fichier de ce type sera rencontré, ce fichier sera passé en premier argument au script. Cette variable peut prendre les valeurs « TextGrid », « wav » ou « TextGrid|wav »[11] : l'extension TextGrid signifie que chaque fois qu'un fichier TextGrid est trouvé, le script est appelé avec en premier argument le fichier TextGrid (les autres arguments étant ceux saisis dans le formulaire) ; l'extension wav fait de même mais avec les fichiers WAV ; enfin, dans le cas de l'extension TextGrid|wav, l'outil cherche les fichiers TextGrid et, pour tout TextGrid trouvé, exécute le script avec en premier argument le fichier TextGrid trouvé (p.ex. `85agm1lg.TextGrid`), en deuxième argument le fichier WAV associé (ici, `85agm1lw.wav`), et les autres arguments saisis dans le formulaire. Pour que ces fichiers puissent être passés en paramètres au script, il faut qu'ils soient les premiers arguments de votre script : le Textgrid doit impérativement être encodé en tant que variable « `sentence textgrid` » et le fichier WAV en tant que « `sentence wav` ». Si le script reçoit un fichier TextGrid et un fichier WAV (soit `$EXTENSION = TextGrid|wav`), le fichier TextGrid doit précéder le fichier wav. Ainsi, lorsqu'un script Praat est importé, le formulaire du script est analysé et le ou les champs correspondant au(x) fichier(s) sont remplacés par un sélecteur de dossier, et les autres arguments (y compris les commentaires) sont reproduits tels quels.

Les autres variables disponibles sont `$SHORT_DESCRIPTION` qui permet de donner une description courte de la fonctionnalité du script et `$ACCESS_KEY` qui définit un raccourci clavier pour lancer le script (ici, il s'agit de la combinaison `control+G`). Il existe également une variable `$BUFFER` qui sera abordée à la sous-section suivante.

Notez enfin que si vous saisissez des caractères accentués dans le formulaire de votre script, vous devrez l'enregistrer au format UTF-16 (Unicode), sans quoi les accents ne s'afficheront pas correctement.

4.3 Écriture de greffons (Praat + Python)

11 La valeur par défaut est « TextGrid » si la variable n'est pas définie.

Bien que le langage de scripts de Praat permette de l'étendre de manière particulièrement intéressante, il est en réalité assez limité et il arrive que l'on souhaite disposer d'un véritable langage de programmation : pour cette raison, la plateforme PFC offre la possibilité d'écrire des greffons, et permet ainsi d'ajouter des fonctionnalités en utilisant le langage Python.

Un greffon est tout simplement un couple script Praat + script python portant la même base mais les extensions « `.praat` » et « `.py` » respectivement, placés dans un dossier portant le nom base + « .plugin »[12].

A titre d'exemple, nous allons créer un greffon (rudimentaire) qui affiche un message à l'utilisateur contenant tous les fichiers TextGrid présents dans un dossier (et ses sous-dossiers). Ce plugin s'appellera « test ». Nous commençons par créer un dossier appelé `test.plugin`. Dans ce dossier, nous créons ensuite un fichier nommé `test.praat` :

```
# test.praat : afficher les fichiers TextGrid
# $INCLUDE = True
# $LABEL = "Mon premier greffon..."
# $EXTENSION = "TextGrid"
# la ligne suivante associe le buffer à la variable temp$
# $BUFFER = temp$

form Genericize/Nativize
      comment Répertoire à traiter récursivement
            sentence textgrid /home/julien/Desktop/Gennat
endform

# la ligne suivante définit le buffer
temp$ = "buffer.txt"

var$ = textgrid$ + newline$
# écriture du nom de fichier dans le buffer
var$ >> 'temp$'
```

On voit ici apparaître la variable `$BUFFER` qui définit un fichier tampon qui sera utilisé pour communiquer avec le script Python[13]. Il est fortement recommandé de ne pas utiliser un chemin absolu pour le chemin du buffer, car cela nuit à la portabilité du script. De plus, dans le cas d'un greffon, le buffer sera automatiquement déplacé dans le dossier « préférences » et sera supprimé après exécution du script Python. Le formulaire du script Praat ne contient qu'un seul champ, à savoir le chemin du fichier TextGrid : ce champ apparaîtra comme un dossier à sélectionner lorsqu'il sera exécuté dans la plateforme PFC.

On crée maintenant un script `test.py` que l'on place également dans le dossier `test.plugin`, et qui sera associé au script `test.praat` : ce script doit impérativement implémenter une fonction `main()` : si cette fonction n'existe pas, une exception de type `PluginMainError` sera levée.

```
# test.py : script Python rudimentaire
# importation du module plugins
from gui import plugins
# importation du dialogue d'information
from gui.messages import info

# fonction principale qui doit être présente
```

12 Notez que les noms de fichiers ne peuvent contenir que des chiffres, des lettres (non accentuées) et le caractère de soulignement « _ ».

13 Le buffer peut également être utilisé dans les scripts Praat seuls, auquel cas l'utilisateur sera informé du fait que des données ont été écrites dans un fichier.

```
def main():
      # récupération du contenu du buffer
      buffer = plugins.getBufferLines()
      # affichage du contenu
      info("".join(buffer))
```

Le paquet `gui` contient les éléments relatifs à l'interface graphique (*Graphical User Interface*). Le module `gui.plugins` fournit un certain nombre de commodités, et permet notamment de récupérer le chemin ou le contenu du fichier tampon.

Pour que le greffon soit reconnu, il suffit de le placer dans le sous-dossier `scripts` du dossier « préférences ». Pour distribuer le greffon, on créera simplement une archive ZIP du dossier du greffon (dans notre cas, on zippera le dossier `test.plugin`). Le nom de l'archive n'a en lui-même pas d'importance, mais nous recommandons de le nommer selon la convention base + « .zip » (en l'occurrence, `test.zip`). Le greffon au format ZIP peut alors être importé via le menu `Fichier > Import > Greffon...` Le greffon deviendra visible après redémarrage de la plateforme (dans notre exemple, il portera le label « Mon premier greffon... »).

Ce greffon n'a en lui-même que peu d'intérêt, mais il montre la facilité avec laquelle on peut étendre la plateforme : on peut en effet accéder à toutes les bibliothèques python et même utiliser l'API de la plateforme. Pour un exemple plus réaliste, consultez le code source du greffon `check_boundaries` (fichiers `check_boundaries.praat` et `check_boundaries.py`).

Pour plus de détails, vous pouvez étudier le code source de la plateforme, et notamment le contenu des paquets `pfc` et `praat`. Notez d'ailleurs que si vous souhaitez développer des extensions et que vous fonctionnez sous Windows, il est nettement préférable d'installer Python normalement et d'utiliser la version source de la plateforme : cela vous permettra d'accéder à toutes les bibliothèques Python, et non pas uniquement à celles qui sont incluses dans la version « toute prête ». Sachez qu'il est également possible d'inclure, en plus des script Praat et Python, n'importe quel type de ressources dans le dossier du greffon (sous-dossiers, images, scripts, modules...). Pour les programmeurs Python, il est intéressant de savoir que tous les dossiers greffons sont ajoutés à la variable `sys.path` : ainsi, un module écrit pour un greffon sera accessible à partir de tous les autres.

5. Les alias

Les alias sont de petits fichiers texte qui pointent vers un autre fichier. Ils correspondent aux "raccourcis" sous Windows. La Plateforme PFC gère les alias pointant vers des enquêtes, des fichiers TextGrid ou des fichiers sonores (WAV ou FLAC).

5.1. Syntaxe des fichiers alias

Tous les fichiers alias sont de simples fichiers texte contenant au moins une ligne ayant la syntaxe suivante : `path=/chemin/vers/fichier`. Le mot `path` doit impérativement être en début de ligne, et `/chemin/vers/fichier` représente le chemin du fichier ou dossier vers lequel pointe l'alias. Les alias peuvent être créés à la main dans n'importe quel éditeur de texte du moment qu'ils respectent cette syntaxe.

5.2. Alias vers une enquête

Un alias d'enquête porte l'extension `SurveyAlias`. La base du nom de fichier doit obligatoirement être le code de l'enquête pointée par l'alias. Par exemple, un alias pointant vers l'enquête Vendée, dont le code est 85a, doit obligatoirement s'appeler `85a.SurveyAlias`.

Les alias d'enquête sont utiles pour créer différents sous-corpus sans avoir à dupliquer les données ou à les déplacer sans cesse. Il est recommandé de stocker toutes les enquêtes dans un même dossier physique, et de créer un corpus « logique » à l'aide de la plateforme. Pour ce faire, on créera un dossier vide (appelons-le `Corpus`) que l'on ouvrira avec la plateforme (à régler dans le menu préférences) ; il suffit ensuite d'ajouter des enquêtes au corpus à l'aide du bouton `Ajouter une enquêtes` dans l'onglet `Enquêtes`. La plateforme demande alors de sélectionner l'enquête que l'on veut ajouter et créera automatiquement un alias dans le dossier ""$Corpus"". Les changements seront pris en compte au redémarrage de la plateforme. On peut ainsi créer autant de corpus que nécessaire, et il est bien sûr possible (quoique déconseillé) de mélanger des enquêtes physiques et des alias d'enquête dans un même dossier.

5.3. Alias vers un fichier son

Un alias de fichier son porte l'extension `SoundAlias`. Il s'agit d'un fichier pointant vers un fichier WAV ou FLAC. Les alias vers des fichiers son ont plusieurs utilités. Lorsque plusieurs locuteurs partagent un même fichier son, on peut ne garder qu'une copie du fichier son pour un locuteur et créer un alias pour les autres locuteurs (au lieu d'avoir à conserver une copie pour chaque locuteur). Lorsqu'on travaille avec un grand nombre d'enquêtes, cela permet de faire des économies d'espace disque appréciables. Mais l'intérêt principal des alias de fichier son est que ces alias peuvent être vides : autrement dit, ils peuvent ne pointer vers aucun fichier son. En créant des fichiers alias vides en lieu et place des fichiers sonores, on peut créer des enquêtes où seules les données textuelles sont chargées. De telles enquêtes occupent très peu d'espace une fois compressées au format ZIP et peuvent par conséquent être échangées par courriel très facilement.

5.4. Alias vers un fichier TextGrid

Un alias de fichier TextGrid porte l'extension `TextAlias`. Ils présentent en pratique un intérêt limité et il est déconseillé de les utiliser.

6. Limitations et problèmes connus

A l'heure actuelle, les requêtes n'ont qu'un support limité des opérateurs booléens : en particulier, l'opérateur ET est absent pour les champs « description ».

La bibliothèque wxPython pour Mac OS X cause un bug d'affichage dans les onglets texte, schwa et liaison. Celui-ci disparaît si l'on sélectionne un autre onglet et que l'on revient sur le premier. Ce bug est spécifique à Mac OS X et semble être un bug de bas-niveau.

Il est important de noter que, pour des raisons d'implémentation, la plateforme considère comme liste complémentaire la première liste complémentaire qu'elle trouve. Par conséquent, si le corpus contient plusieurs listes complémentaires, la première sera utilisée pour toutes les enquêtes ayant une liste complémentaire. Si l'on souhaite travailler sur des listes de mots, il est donc recommandé dans ce cas de constituer des corpus différents pour chaque enquête.

Sous Linux, il n’est pas possible d’arrêter l’écoute d’un fichier avec le bouton stop des onglets Texte, Schwa et Liaison. Il s’agit d’un bug de bas-niveau. Toujours sous Linux, après le chargement des enquêtes, il est nécessaire d’appuyer sur le bouton « close » pour que la plateforme apparaisse.

7. Développements futurs

Les prochains développements porteront sur l’intégration de l’étiquetage grammatical (à l’aide du TreeTagger) dans la plateforme. Il s’agit d’abord de traiter un nombre maximal d’enquêtes, puis

d’étendre sa disponibilité aux tires orthographique et schwa. En parallèle, le code sera porté vers Python 3.0. La possibilité d’enrichir les métadonnées par un système de descripteurs (*tags*) est par ailleurs envisagée.

Pour toutes remarques, suggestions et/ou rapports de bugs, n'hésitez pas à contacter Julien Eychenne.