Unitex (Cont.)

Abdelhalim hafedh DAHOU

Les automates du texte

L’automate du texte permet d’exprimer toutes les interprétations lexicales possibles des mots. Ces différentes interprétations sont les différentes entrées présentes dans les dictionnaires du texte.

Cette présentation va explorer le détail de ***la construction*** des automates ainsi que les ***opérations*** qui peuvent leur être appliquées, en particulier ***la levée d’ambiguïtés*** au moyen du programme ELAG et aussi ***la normalisation*** et ***la conservation des bon chemins*** . Depuis la version 2.1, il est possible d’effectuer des recherches de motifs sur l’automate du texte.

Les automates du texte

Construction

Il est recommandé d'avoir découpé le texte en phrases et de lui avoir appliqué les dictionnaires.

- Si vous n'avez pas découpé le texte en phrases, le programme de construction découpera arbitrairement le texte en séquences de 2000 unités lexicales.
- Si vous n'avez pas appliqué les dictionnaires, les automates de phrase que vous obtiendrez ne seront constitués que d'un seul chemin ne comportant que des mots inconnus.

Les automates du texte

Normalisation de formes ambiguës

Lors de la construction de l'automate, il est possible d'effectuer une normalisation de

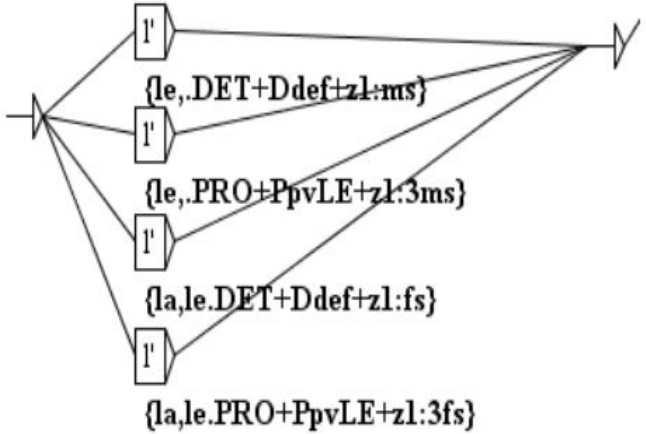
formes ambiguës en appliquant une grammaire de normalisation.

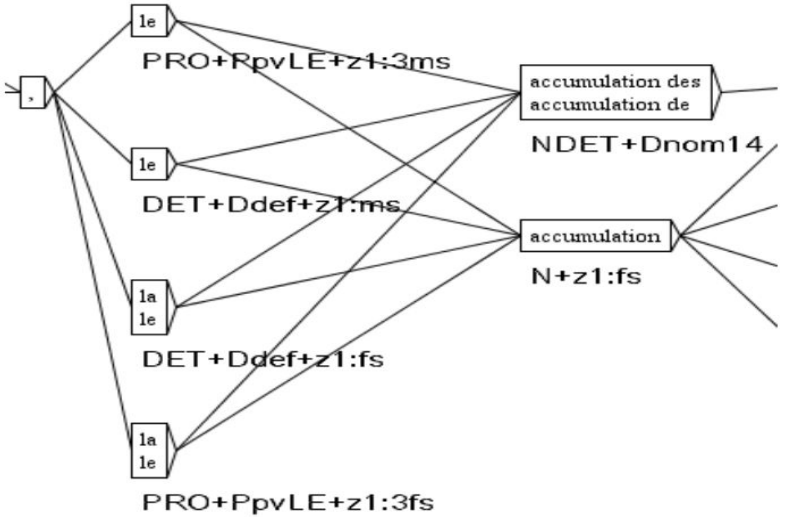
Si une séquence du texte est reconnue par la grammaire de normalisation, toutes les interprétations décrites par la grammaire sont insérées dans l'automate du texte.

Exemple : la séquence ***l'*** .

Les automates du texte

Normalisation de formes ambiguës





Les automates du texte

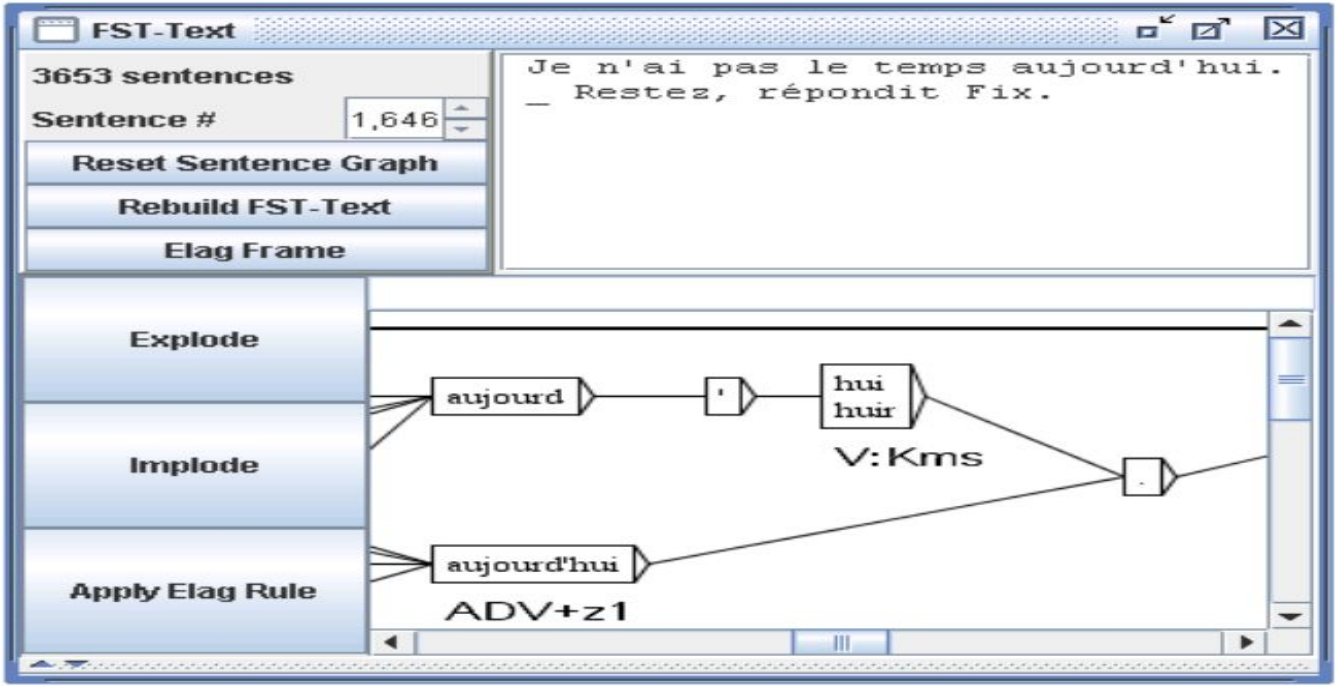
Conservation des meilleurs chemins

Il peut arriver qu'un mot inconnu vienne parasiter l'automate du texte en étant concurrent avec une séquence complètement étiquetée.

Il est possible de supprimer ces chemins parasites. a l'aide de l'option "Clean Text FST". Cette option indique au programme de construction de l'automate qu'il doit nettoyer chaque automate de phrase.

Les automates du texte

Conservation des meilleurs chemins



Les automates du texte

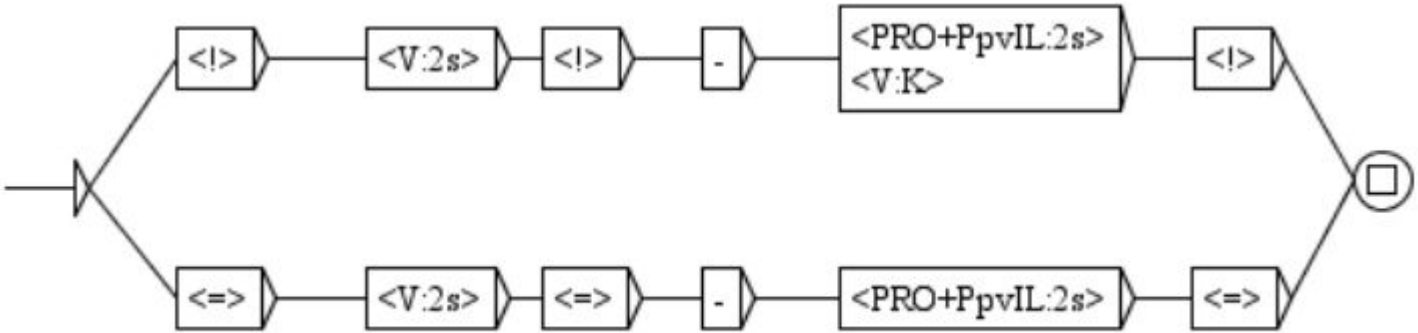
Levée d'ambiguïtés lexicales avec ELAG

Le programme ELAG permet d'appliquer des grammaires de levée d'ambiguïtés sur l'automate du texte. Les grammaires manipulées par ELAG ont une syntaxe particulière. Elles comportent deux parties, que nous appellerons partie si et alors.

La signification d'une grammaire est que dans l'automate du texte, si l'on trouve une séquence reconnue par la partie ***si*** alors elle doit aussi être reconnue par la partie ***alors*** de la grammaire, faute de quoi elle sera retirée de l'automate du texte.

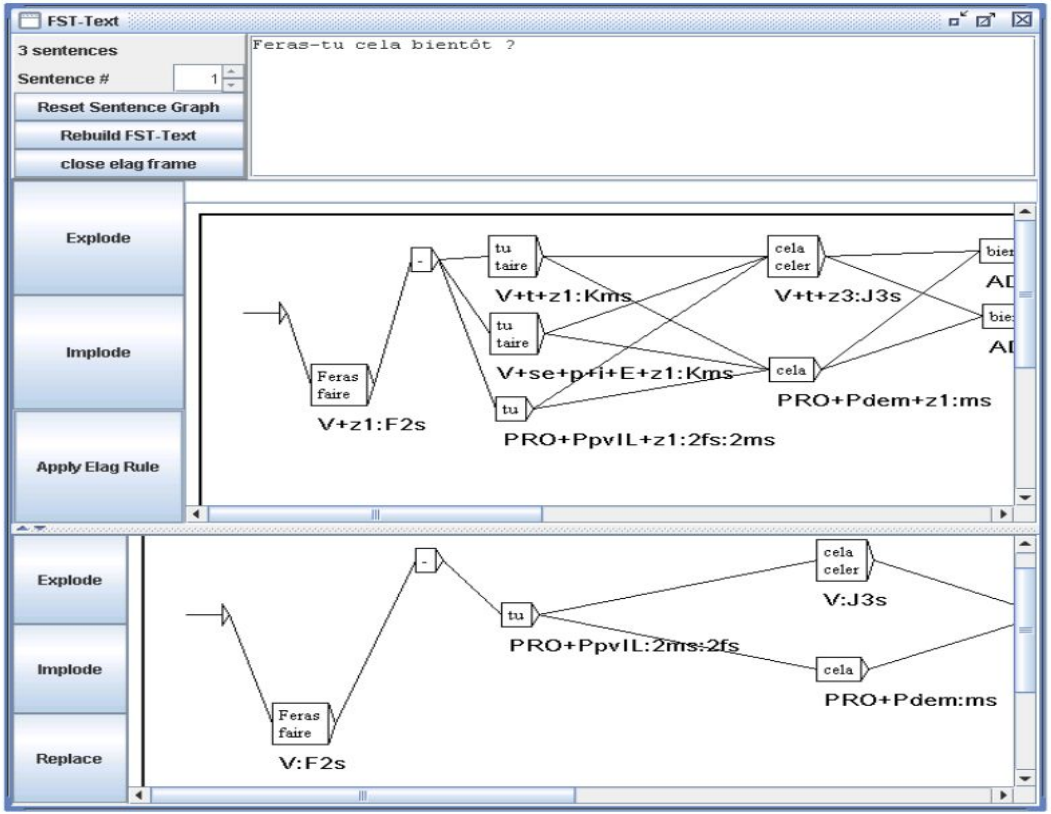
Les automates du texte

Levée d'ambiguïtés lexicales avec ELAG



Les automates du texte

Levée d'ambiguïtés lexicales avec ELAG



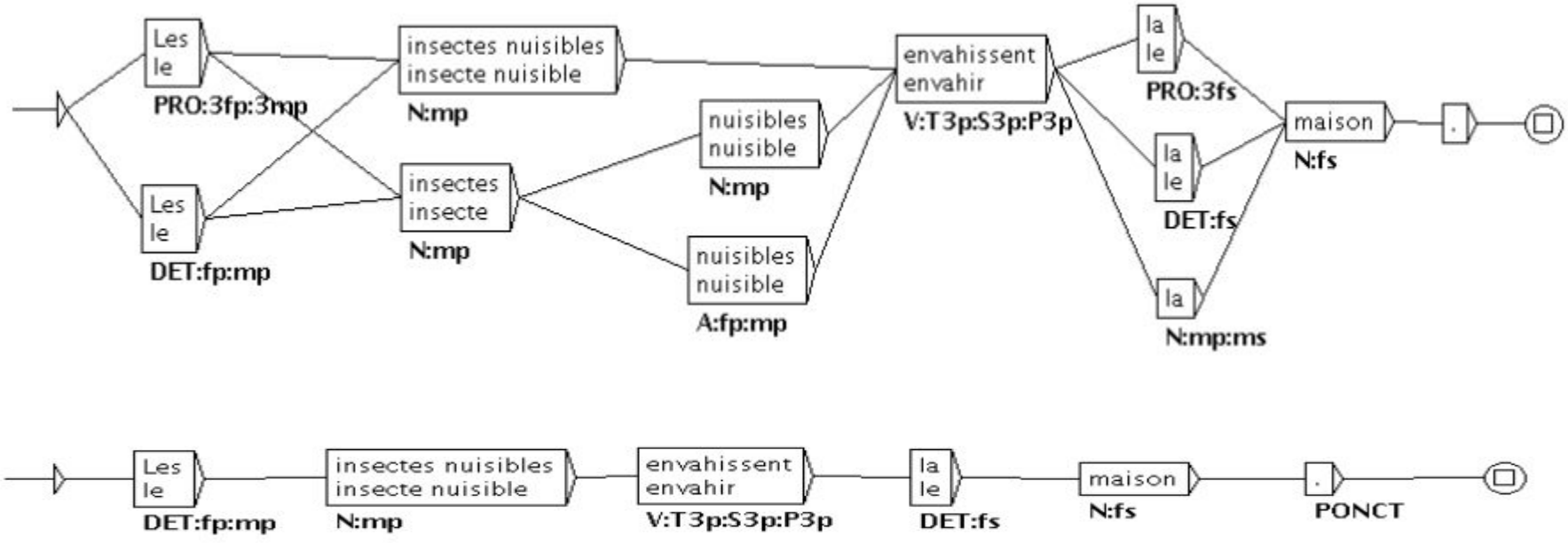
Les automates du texte

Linéarisation de l'automate du texte avec le taggeur

Par défaut, l'automate du texte contient de nombreux chemins étiquetés en raison de l'ambiguïté lexicale. Le processus de linéarisation consiste à choisir un chemin unique, une séquence d'étiquettes avec une étiquette par token et de supprimer les autres. Le résultat est un automate du texte avec un seul chemin.

Les automates du texte

Linéarisation de l'automate du texte avec le taggeur



Les automates du texte

Affichage de la Table

Les automates de phrases peuvent être affichées sous forme de tableau. Il devrait être considéré comme une vue approximative et compacte des informations contenues dans l'automate.

Les automates du texte

Affichage de la Table

Automaton | Table

Filter grammatical/semantic codes

☐ Always show POS category, regardless filtering | Export all text as POS list | ☐ Export DELAF style

Filter Tags

◉ All ○ Only POS category ○ Use filter:

Form	POS sequence #1	POS sequence #2	POS sequence #3	POS sequence #4
Anglais	{Anglais,.N+z1:ms:mp}	{Anglais,anglais.A+z1:ms:mp}	{Anglais,anglais.N+z1:ms}	{Anglais,angler.V+z3:I1s:I2s}
,	,			
à	{à,.PREP+z1}			
à coup sûr	{à coup sûr,.ADV+z1}			
coup	{coup,.N+z1:ms}			
sûr	{sûr,.A+z1:ms}	{sûr,.ADV+z1}		
,	,			
Phileas	Phileas			
Fogg	Fogg			
n'	{ne,.XI+z1}			
était	{était,être.V+z1:I3s}			
peut-être	{peut-être,.ADV+z1}			
peut-être pas	{peut-être pas,.ADV+z1}			
pas	{pas,.ADV+z1}	{pas,.N+z1:ms:mp}		
Londonner	Londonner			
.	.			

Les automates de sequence

La construction de grammaires locales peut être un long processus durant lequel le linguiste répète de nombreuses fois les mêmes opérations. La finalité du programme ***Seq2Grf*** est de produire rapidement et automatiquement des grammaires locales.

Pour un document donné (TEILite ou des fichiers au format txt ou SNT) ce programme construit un unique automate qui reconnaît toutes les séquences contenues dans le document.

Les automates de sequence

La construction de grammaires locales peut être un long processus durant lequel le linguiste répète de nombreuses fois les mêmes opérations. La finalité du programme ***Seq2Grf*** est de produire rapidement et automatiquement des grammaires locales.

Pour un document donné (TEILite ou des fichiers au format txt ou SNT) ce programme construit un unique automate qui reconnaît toutes les séquences contenues dans le document.

Les automates de sequence

Le corpus de séquences est stocké dans un seul fichier qui peut avoir l'un des formats suivants :

- **Fichiers texte brut**, dans lequel les séquences sont délimitées par des fins de lignes
- **Fichiers SNT déjà prétraités** : les séquences sont délimitées par {STOP}
- **Fichiers TEILite** dont les séquences sont délimitées par un tag xml de la forme :

<seg type="sequence">example</seg>

Les automates de sequence

Vous pouvez définir des options supplémentaires pour produire un graphe qui permet une reconnaissance approximative : vous pouvez fixer le nombre de jokers à utiliser pour produire de nouvelles séquences dérivées des séquences du corpus original, et choisir le joker approprié.

- ***insertion*** : pour chaque séquence, ajouter à l'automate toutes les séquences où <TOKEN> a été inséré entre deux mots de la séquence originale.

- ***remplacement*** : pour chaque séquence, ajouter à l'automate toutes les séquences où *i* tokens ont été remplacés par <TOKEN>

- ***suppression*** : pour chaque séquence, ajouter à l'automate toutes les séquences où un token a été supprimé