Tutoriel Prise en main d'Unitex-CasSys pour l'annotation de corpus

Denis Maurel
Université de Tours

La correction complète de ce tutoriel est disponible sur :

https://tln.lifat.univ-tours.fr/medias/fichier/correction-tutoriel-priseenmain-unitex-annotationcorpus-denis-maurel_1603455974191-zip?ID_FICHE=334600&INLINE=FALSE

Ce tutoriel suppose de connaitre Unitex et la création de graphe, par exemple en ayant fait le tutoriel *Prise en main d'Unitex pour l'annotation de corpus* disponible sur :

https://tln.lifat.univ-tours.fr/version-francaise/ressources/tutoriels-unitex.

A titre d'exemple réel, on peut télécharger la cascade CasEN sur la reconnaissance d'entités nommées, réalisée par le Lifat, disponible sur : https://tln.lifat.univ-tours.fr/tln/version-francaise/navigation/ressources/casen/

La première chose à faire est d'installer le logiciel libre Unitex. Il faut télécharger la version 3.2 ou une version ultérieure : https://unitexgramlab.org/. En cas de difficultés d'installation, le manuel est aussi accessible en ligne : https://unitexgramlab.org/releases/3.3/man/Unitex-GramLab-3.3-usermanual-fr.pdf. Il faut alors consulter les sections 1.3-1.7.

Préparation

Le plus simple est de télécharger le fichier de préparation[1] sur le site TLN :
https://tln.lifat.univ-tours.fr/medias/fichier/preparation-tutoriel-unitex-cassys-denis-maurel_1562936277186-zip?ID_FICHE=321996&INLINE=FALSE
et de le dézipper dans votre dossier personnel Unitex (les fichiers se placeront au bon endroit).

Puis de passer à la section 1, page 3.

En cas d'impossibilité, poursuivre ci-dessous.

Création de dossiers

Plaçons-nous dans notre dossier personnel Unitex, dans *French*.

[1] Sauf si vous l'avez déjà fait pour le Tutoriel Unitex, dans ce cas passez directement à la section 1, page 3.

1. Dans *CasSys* : Créons un premier dossier pour l'ensemble du tutoriel, nommé *Tutoriel_Unitex_CasSys*[2]. Créons, dans ce dossier, trois dossiers nommés *TexteBrut*, *TexteXML* et *Nombres*.
2. Dans *Corpus* : Créons un premier dossier pour la première partie du tutoriel, nommé *Tutoriel_Unitex_CasSys*. Créons, dans ce dossier, deux dossiers nommés *TexteBrut* et *TexteXML*. Puis, créons un second dossier nommé *80jours*, dans lequel nous faisons glisser le fichier *80jours.txt* présent dans la distribution d'Unitex.
3. Dans *Graph* : Créons un premier dossier pour l'ensemble du tutoriel, nommé *Tutoriel_Unitex_CasSys*. Créons, dans ce dossier, deux dossiers nommés *Texte* et *Nombres*. Dans chacun de ces deux dossiers, créons deux nouveaux dossiers nommés *Analyse* et *Synthese*.

Copie des textes à analyser

Le texte ci-dessous est à recopier via l'éditeur d'Unitex (menu *File Edition/New File*) et à enregistrer sous le nom *texteBrut.txt* dans le dossier *French\Corpus\Tutoriel_Unitex_CasSys\TexteBrut*[3].

Le maire de la ville de Prèdetours a organisé une grande cérémonie le mardi 11 novembre 2014 pour modifier le nom de la rue de Prèdetours en rue du 11 novembre 1918.
Ce mardi 11 novembre prouve que Prèdetours n'a pas oublié ses anciens combattants.
D'ailleurs il n'est pas impossible qu'une Maison communale du 11 novembre voit le jour pour célébrer le 11 novembre 2018...

Puis utiliser le menu *File/Save As…* pour l'enregistrer sous le nom, *texteXML.txt*, dans le dossier *French\Corpus\Tutoriel_Unitex_CasSys\TexteXML*. Compléter alors le texte pour obtenir :

<tei>
<title>La Gazette de l'hôpital Bretonneau</title>
<date>Publié le mercredi 19 novembre 2014</date>
<body>
* Le maire de la ville de Prèdetours a organisé une grande cérémonie le mardi 11 novembre 2014 pour modifier le nom de la rue de Prèdetours en rue du 11 novembre 1918. *
* Ce mardi 11 novembre prouve que Prèdetours n'a pas oublié ses anciens combattants. *
* D'ailleurs il n'est pas impossible qu'une Maison communale du 11 novembre voit le jour pour célébrer le 11 novembre 2018... *
</body>
</tei>

Enregistrer le fichier puis fermer la fenêtre.

[2] Remarque importante : comme Unitex est un logiciel multiplateformes, les noms de dossiers et de fichiers utilisés doivent ne comporter ni diacritiques, ni espaces.
[3] Si il apparait, ne pas tenir compte de l'avis *This is not necessarily the text being processed by Unitex*, cliquer sur *OK*.

1 Texte brut à analyser en entités nommées

Dans cet exercice, le but est la création d'un fichier où seront balisés les dates, les villes, les adresses et les batiments du fichier *TexteBrut.txt*.

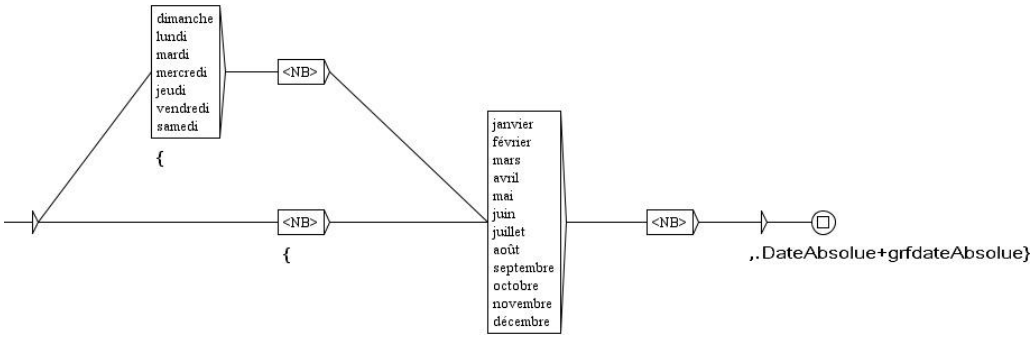
1.1 Cascade d'analyse

Commençons par ouvrir, dans le menu *Text* (et non *File Edition* comme précédemment), le fichier *French\Corpus\Tutoriel_Unitex_CasSys\TexteBrut\texteBrut.txt*, en répondant *No* à la question *Do you want to preprocess the text*[4].

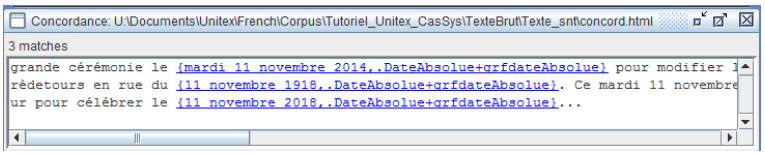
Créons un premier graphe qui sera enregistré (comme les suivants) dans le dossier *French\Graphs\Tutoriel_Unitex_CasSys\Texte\Analyse*.

1.1.1 Graphe *dateAbsolue.grf*

Ce premier graphe va reconnaitre les *dates absolues*, c'est-à-dire les dates contenant une année[5]. Pour simplifier les numéros de jours et d'années ne sont pas détaillés, mais simplement repérés par le code *<NB>* qui désigne une séquence de chiffres[6]. La date reconnue sera placée à l'intérieur des symboles *{* et *,.DateAbsolue}*, afin de constituer une étiquette lexicale[7], qui simule une entrée de dictionnaire de catégorie *DateAbsolue*. On ajoutera un trait pour le débogage, avec le nom du graphe, sous la forme *+grfdateAbsolue*.



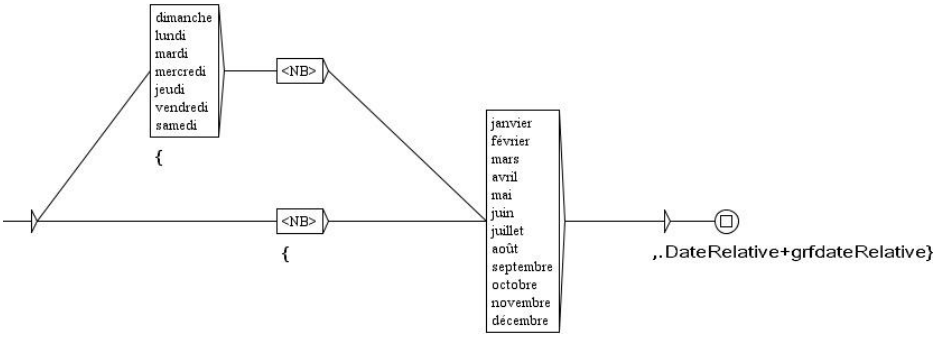
Enregistrons et compilons[8] (bouton *Compile*) ce graphe. Utilisons le menu *Text/Locate Pattern…* en choisissant ce graphe et en cochant l'option *Merge with input text*. Cliquons sur le bouton *Search*, puis sur *OK*. Règlons la taille du contexte : *Left: 20 chars* et *Right: 255 chars* ; puis l'ordre : *Sort according to:Text Order*. Cliquons sur le bouton *Build concordance*.



1.1.2 Graphe *dateRelative.grf*

Créons un deuxième graphe pour reconnaitre les *dates relatives*, c'est-à-dire les dates sans mention de l'année. Ouvrons le graphe *dateAbsolue.grf* et enregistrons-le sous le nom *dateRelative.grf*. Il nous reste à supprimer l'année et à modifier la sortie.

[4] Contrairement au premier tutoriel, nous n'utiliserons pas de dictionnaires.
[5] Cette définition et la suivante sont empruntées à la campagne d'évaluation *Ester*.
[6] Pour la liste des symboles spéciaux, voir le manuel, section 4.3.1.
[7] Voir le manuel, section 2.5.4.
[8] Pour créer une cascade il est obligatoire de compiler les graphes que nous souhaitons y placer.



Enregistrons et compilons ce graphe.

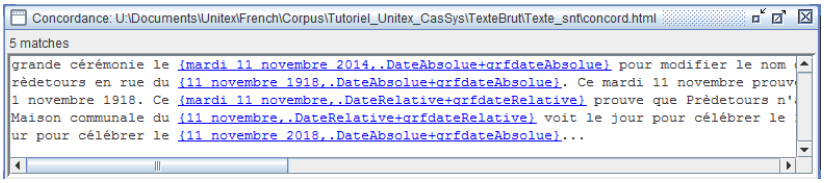
1.1.3 Cascade *analyse.csc*

Nous avons donc deux graphes que nous allons passer successivement sur le texte, grâce à une première cascade, que nous appellerons *cascade d'analyse*. Ouvrons le menu *Text/Apply CasSys Cascade…* et cliquons sur le bouton *New*. Plaçons-nous dans le dossier *French\Graphs\Tutoriel_Unitex_CasSys\Texte\Analyse*. Avec la souris, faisons glisser le graphe *dateAbsolue.fst2*[9], puis le graphe *dateRelative.fst2*, dans la partie droite de la fenêtre.

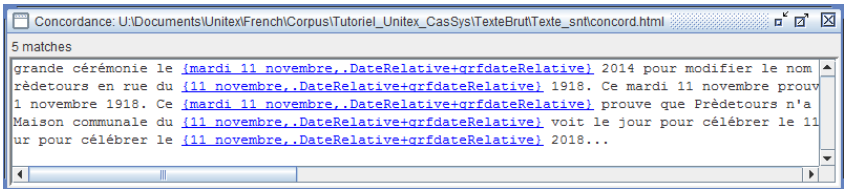
Voir la vidéo : https://tln.lifat.univ-tours.fr/medias/video/creationcascade_1580217726251-mp4?ID_FICHE=334606&INLINE=FALSE.

#	Disabled	Name	Merge	Replace	Until Fix Point	Generaliz...
1	☐	dateAbsolue.fst2	☑	☐	☐	☐
2	☐	dateRelative.fst2	☑	☐	☐	☐

Enregistrons cette cascade dans le dossier *French\CasSys\Tutoriel_Unitex_CasSys\TexteBrut* sous le nom *analyse.csc*. Fermons la fenêtre et cliquons sur le bouton *Launch*. Règlons la taille du contexte : *Left: 20 chars* et *Right: 255 chars* ; puis l'ordre : *Sort according to:Text Order*. Cliquons sur le bouton *Build concordance*.



Remarquons que l'ordre des graphes est important, car, si nous inversons (avec la souris) ces deux graphes, la concordance obtenue n'est plus la même.



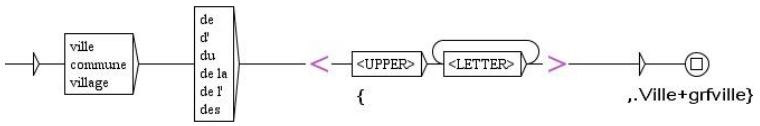
En effet, le graphe *dateRelative* a reconnu les cinq dates et donc, le graphe *dateAbsolue* n'avait plus rien à reconnaitre !

1.1.4 Graphe *ville.grf*

Le graphe suivant va reconnaitre les noms de ville, commune et village. Après l'un de ces trois mots, suivi de la préposition *de*, nous placerons un mot commençant par une majuscule, reconnu en utilisant le mode morphologique d'Unitex[10] et les codes *<UPPER>* et *<LETTER>*.

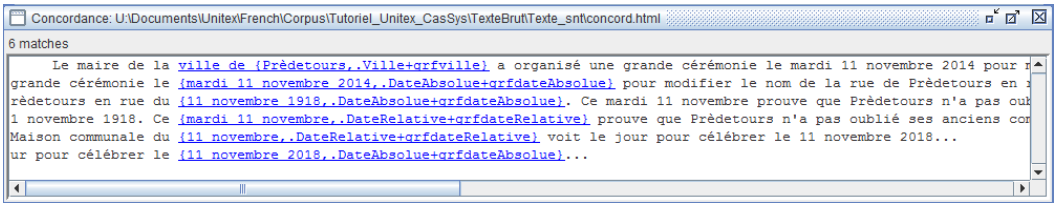
Voir la vidéo : https://tln.lifat.univ-tours.fr/medias/video/ville_1580217793422-mp4?ID_FICHE=334606&INLINE=FALSE.

[9] Les fichiers *.fst2* sont des graphes compilés.
[10] Voir le manuel, section 6.4.



Enregistrons et compilons ce graphe, puis ajoutons-le en dernier à la cascade *analyse.csc* (remise dans le bon ordre).

1	☐	dateAbsolue.fst2	☑	☐	☐	☐
2	☐	dateRelative.fst2	☑	☐	☐	☐
3	☐	ville.fst2	☑	☐	☐	☐

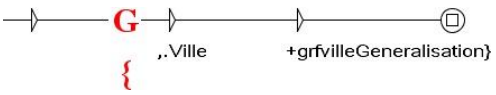


Nous pouvons remarquer sur cette concordance que *Prédetours* a été reconnu dans *ville de Prédetours*, mais, ni dans *rue de Prédetours*, ni dans *Prédetours n'a pas*. En effet, le premier terme est dans un contexte qui a permis sa reconnaissance, les deux autres non.

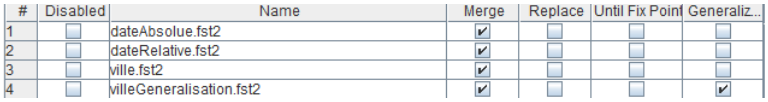
1.1.5 Graphe de généralisation d'étiquetage

Un graphe de généralisation d'étiquetage permet d'étiqueter des mots hors contexte s'ils ont déjà été étiquetés ailleurs dans le texte, grace à un contexte. Par exemple, ici, la reconnaissance de *Prédetours* par le contexte *ville de* entrainera celui des deux autres occurrences de ce mot par le graphe *villeGeneralisation.grf*. Pour créer un graphe générique simple[11], il faut sélectionner la boite qui contient le code à généraliser et cliquer, comme pour les variables, les contextes et le mode morphologique, sur le bouton *insert generic graph mark before the selected box*.

Voir la vidéo : https://tln.lifat.univ-tours.fr/medias/video/villegeneralisation_1580217856910-mp4?ID_FICHE=334606&INLINE=FALSE.

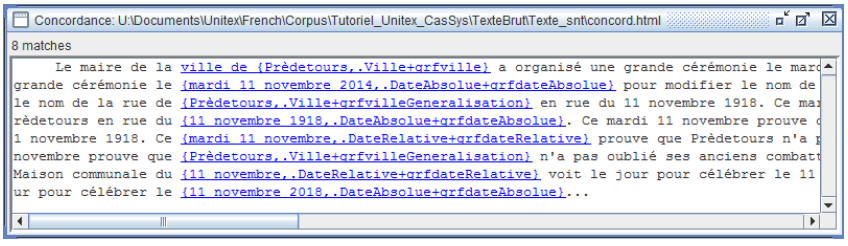


Enregistrons et compilons ce graphe, puis ajoutons-le en dernier à la cascade *analyse.csc*, en cochant, en plus de la case *Merge*, la case *Generaliz…*



#	Disabled	Name	Merge	Replace	Until Fix Point	Generaliz...
1	☐	dateAbsolue.fst2	☑	☐	☐	☐
2	☐	dateRelative.fst2	☑	☐	☐	☐
3	☐	ville.fst2	☑	☐	☐	☐
4	☐	villeGeneralisation.fst2	☑	☐	☐	☑

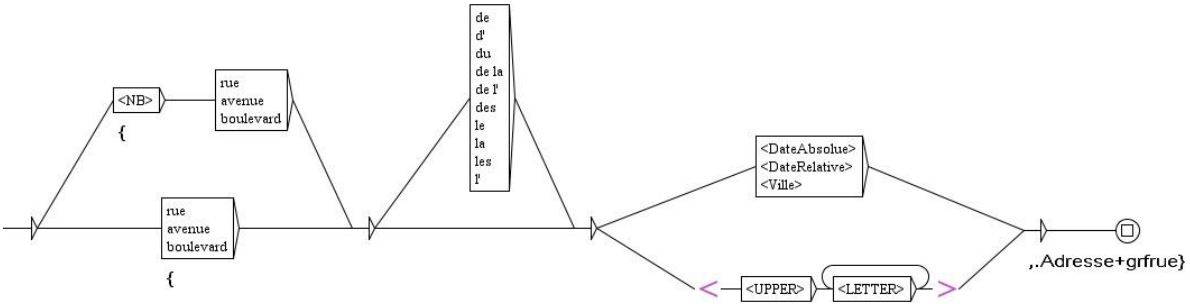
On obtient ce qu'on souhaitait : trois reconnaissances du mot *Prèdetours*.



1.1.6 Graphe *rue.grf*

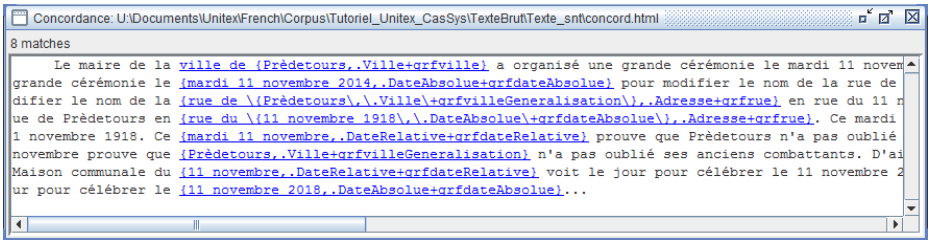
Créons maintenant un graphe pour reconnaître les noms de rue. Après les mots *rue*, *avenue*, *boulevard*…, éventuellement précédé d'un numéro et éventuellement suivi d'une préposition ou d'un déterminant, nous allons considérer deux possibilités : une date ou une ville, déjà reconnue par la cascade et donc désignée par les codes *<DateAbsolue>*, *<DateRelative>* ou *<Ville>* ; un mot commençant par une majuscule.

[11] Des graphes de généralisation d'étiquetage plus sophistiqués sont possibles aussi. Voir le manuel, section 12.3.



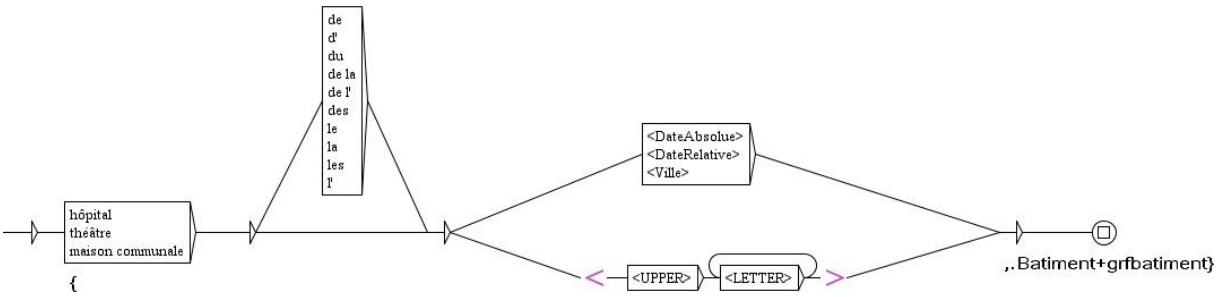
Enregistrons et compilons ce graphe, puis ajoutons-le en dernier à la cascade *analyse.csc*.

#	Disabled	Name	Merge	Replace	Until Fix Point	Generaliz...
1	☐	dateAbsolue.fst2	☑	☐	☐	☐
2	☐	dateRelative.fst2	☑	☐	☐	☐
3	☐	ville.fst2	☑	☐	☐	☐
4	☐	villeGeneralisation.fst2	☑	☐	☐	☑
5	☐	rue.fst2	☑	☐	☐	☐



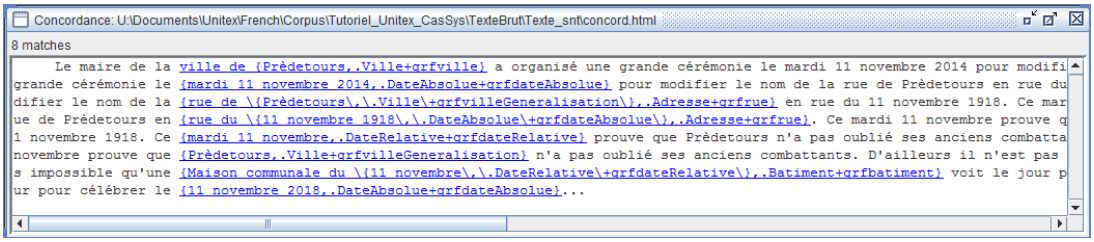
1.1.7 Graphe *batiment.grf*

Enfin, cliquons sur le menu *FSGraph/Save as…* et appelons ce graphe *batiment.grf*. Il nous suffit de modifier le début et la fin du graphe pour reconnaitre les batiments.



Enregistrons et compilons ce graphe, puis ajoutons-le en dernier à la cascade *analyse.csc*.

#	Disabled	Name	Merge	Replace	Until Fix Point	Generaliz...
1	☐	dateAbsolue.fst2	☑	☐	☐	☐
2	☐	dateRelative.fst2	☑	☐	☐	☐
3	☐	ville.fst2	☑	☐	☐	☐
4	☐	villeGeneralisation.fst2	☑	☐	☐	☑
5	☐	rue.fst2	☑	☐	☐	☐
6	☐	batiment.fst2	☑	☐	☐	☐

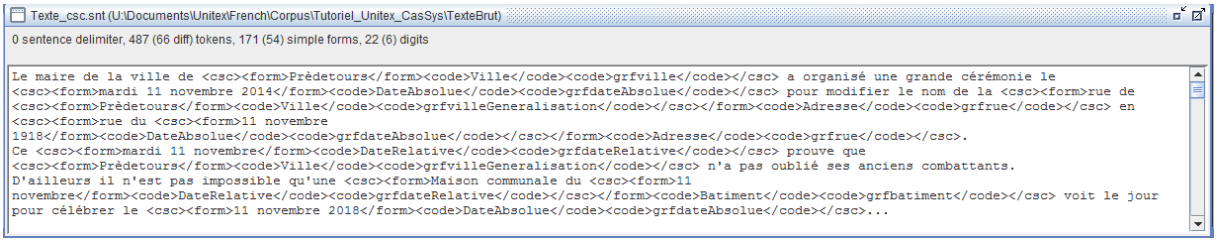


Le passage de la cascade sur le fichier *texteBrut.txt* génère (entre autres) un fichier XML, *texteBrut_csc.txt*, où les accolades sont remplacées par les balises *<csc>…</csc>*, la forme reconnue par *<form>…</form>* et le code et les traits par *<code>…</code>*[12].

[12] Voir le manuel, section 12.4.3.

1.2 Cascade de synthèse

Ouvrons dans le menu *Text* le fichier *French\Corpus\Tutoriel_Unitex_CasSys\TexteBrut\texteBrut_csc.txt*, en répondant *No* à la question *Do you want to preprocess the text*[13].



La cascade de synthèse sera passée sur ce fichier et permettra de créer un fichier balisé au format XML de notre choix. Créons un premier graphe qui sera enregistré (comme les suivants) dans le dossier *French\Graphs\Tutoriel_Unitex_CasSys\Texte\Synthese*.

1.2.1 Graphe *balisage.grf*

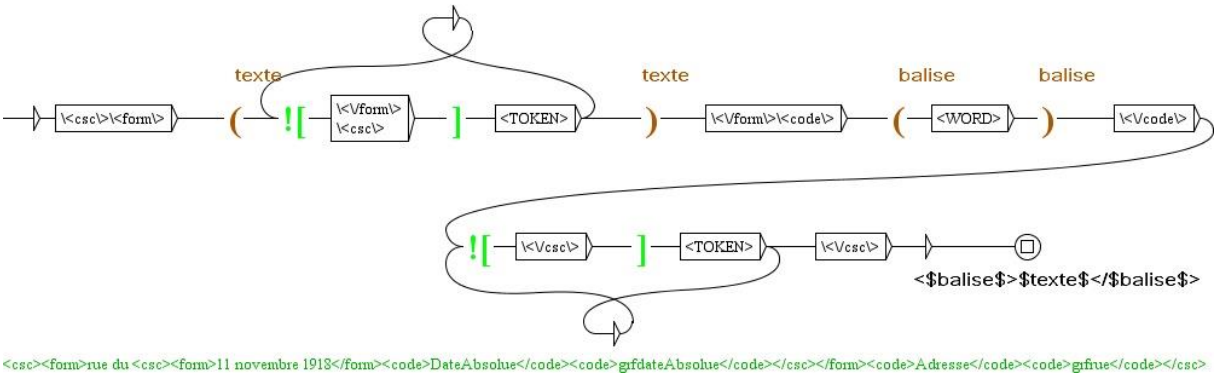
Le graphe *balisage* sera passé en mode *Replace* et aura pour but de :

1. supprimer les balises *csc*, *form* et *code* ;
2. sauvegarder le texte dans une variable[14] *$texte$* et la catégorie dans une variable *$balise$* ;
3. remplacer l'ensemble par le texte balisé : *<$balise$>$texte$</$balise$>*.

Pour récupérer le texte, nous utilisons une boucle qui reconnait tous les éléments (balise *<TOKEN>*) dont le contexte droit[15] n'est, ni une balise *</form>*, ni une balise *<csc>*, qui débuterait une cascade imbriquée dans celle que nous analysons. Puis nous reconnaissons la catégorie (supposée être une séquence de lettres) par le code *<WORD>* et nous cherchons la balise *</csc>* de fin de cascade.

Une remarque importante : comme Unitex utilise des chevrons pour ses codes, nous devons, pour reconnaitre un chevron dans un texte le *protéger* (c'est-à-dire le faire précéder) par un antislash[16]. Par exemple, la balise *<csc>* sera écrite dans une boite Unitex \<csc\>. De même pour le slash qui sert, dans Unitex, à marquer le début des sorties. Par exemple, la balise *</csc>* sera écrite dans une boite Unitex \<\/csc\>.

Pour nous aider à bien écrire ce graphe complexe, nous allons insérer un exemple dans une boite de commentaire[17], c'est-à-dire une boite non reliée aux autres et qui commence par un slash.



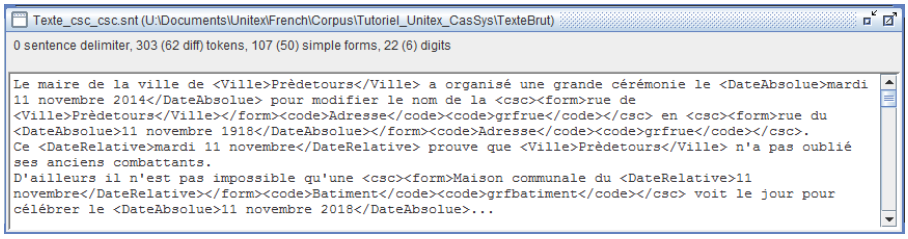
Voir la vidéo : https://tln.lifat.univ-tours.fr/medias/video/balisage_1580217918886-mp4?ID_FICHE=334606&INLINE=FALSE.

[13] La synthèse ne nécessite pas de dictionnaires, car il s'agit juste d'une modification des balises XML.
[14] Une variable dans Unitex mémorise la partie du texte reconnue par les boites entre parenthèses. Ces parenthèses se place comme les contextes et le mode morphologique, par le bouton *surround box selection with an input variable*. Le nom de la variable est libre. En sortie, ce nom est encadré par un caractère *$*. Voir le manuel, section 5.2.5.
[15] Voir le manuel, section 6.3.1.
[16] Voir le manuel, section 5.2.7. Ceci ne concerne pas les sorties pour lesquelles aucune protection n'est nécessaire.
[17] Voir le manuel, section 5.2.1.

Enregistrons et compilons ce graphe, puis créons une nouvelle cascade, que nous appellerons *synthese.csc*, et ajoutons-le en première position, en cochant le mode *Replace*.

#	Disabled	Name	Merge	Replace	Until Fix Point	Generaliz...
1	☐	balisage.fst2	☐	☑	☐	☐

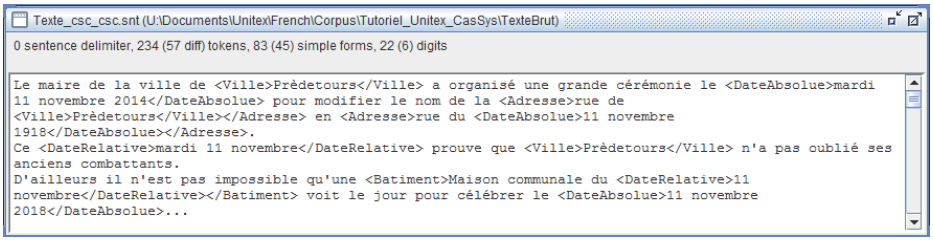
Lançons cette cascade. Nous obtenons la création d'un nouveau fichier XML, *texteBrut_csc_csc.txt*.[18]



La transformation du balisage des éléments non imbriqués est réussie. Par exemple, la première date devient *<DateAbsolue>mardi 11 novembre 2014</DateAbsolue>*. On remarque que, pour les éléments imbriqués, la partie intérieure est bien transformée. Il suffit donc de relancer une deuxième fois le graphe. Mais, s'il y avait trois niveaux d'imbrication, il faudrait le lancer trois fois… Nous allons donc cocher la case *Until Fix Point* qui relance le graphe jusqu'à ce que le texte ne soit plus modifié[19].

#	Disabled	Name	Merge	Replace	Until Fix Point	Generaliz...
1	☐	balisage.fst2	☐	☑	☑	☐

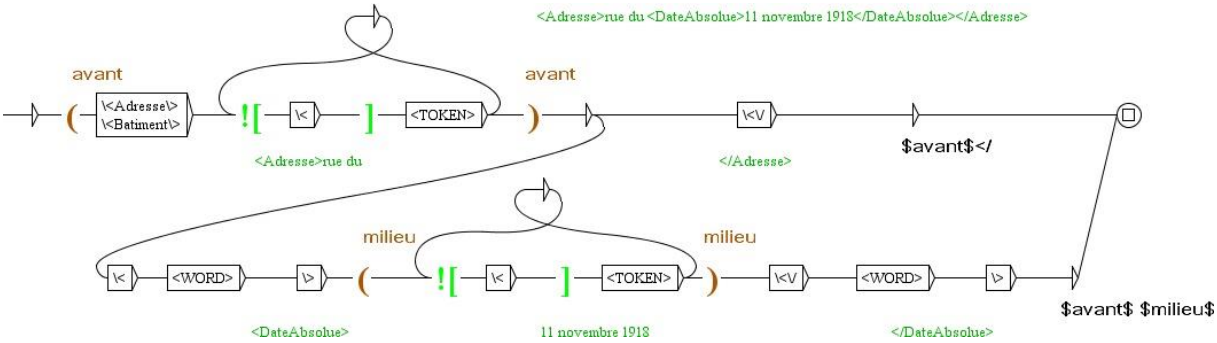
En lançant à nouveau cette cascade, le fichier *texteBrut_csc_csc.txt* devient correct, avec, par exemple : *<Adresse>rue du <DateAbsolue>11 novembre 1918</DateAbsolue></Adresse>*.



1.2.2 Graphe *suppressionInterne.grf*

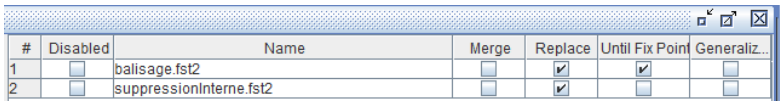
Bien sûr, la pertinence de baliser une date ou une ville dans une adresse n'est pas évidente ! Indispensable pour reconnaître l'adresse, nous souhaiterions maintenant supprimer ce balisage. De même pour les noms de batiment.

Le graphe *suppressionInterne.grf* commence par placer dans la variable *$avant$* la balise et ce qui la suit jusqu'à l'ouverture d'une autre balise. Si cette balise est une balise fermante, il n'y a pas d'imbrication et on recopie le tout (c'est-à-dire *$avant$</*). Sinon, si cette balise est une balise ouvrante, il y a imbrication et on mémorise juste le texte dans une variable *$milieu$*, puis on réécrit simplement *$avant$ $milieu$*.



Enregistrons et compilons ce graphe, puis ajoutons-le en dernier à la cascade *synthese.csc*.

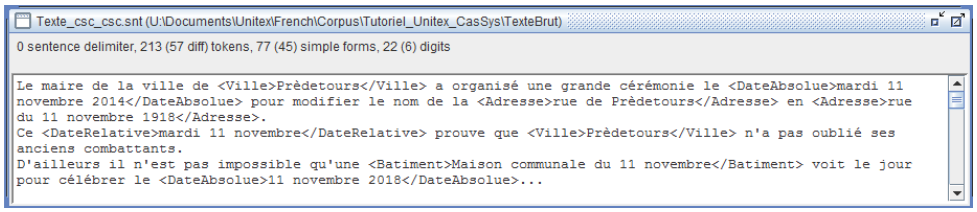
[18] Lorsqu'il y a des suppressions (mode *Replace*) le résultat est difficile à visualiser dans une concordance. Nous afficherons ici systématiquement le fichier résultant, suffixé par *_csc_csc.txt*.
[19] Voir le manuel, section 12.2.2.



#	Disabled	Name	Merge	Replace	Until Fix Point	Generaliz...
1	☐	balisage.fst2	☐	☑	☑	☐
2	☐	suppressionInterne.fst2	☐	☑	☐	☐

1.2.3 Résultat final

En lançant cette cascade, le fichier *texteBrut_csc_csc.txt* devient tel que nous le souhaitons, avec, par exemple : *<Adresse>rue du 11 novembre 1918</Adresse>*.



Cependant une difficulté peut surgir, traitée en annexe (section 4, page 20), pour les utilisateurs avancés.

2 Texte XML à analyser en entités nommées

Dans cet exercice, le but est le même que pour l'exercice précédent, mais le fichier est un fichier XML.

Une remarque : ce fichier ne doit pas avoir l'extension *.xml*, sinon Unitex supprime les balises pour analyser le texte. Il doit être renommé[20] si c'est le cas.

Nous allons utiliser les mêmes cascades, complétées par quelques graphes supplémentaires. Copions les deux cascades (*analyse.csc* et *synthese.csc*) du dossier *French\CasSys\Tutoriel_Unitex_CasSys\TexteBrut* dans le dossier *French\CasSys\Tutoriel_Unitex_CasSys\TexteXML*.

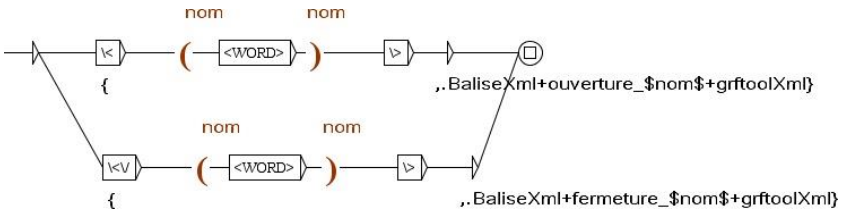
2.1 Cascade d'analyse

Pour lancer l'analyse, commençons par ouvrir, toujours dans le menu *Text*, le fichier *French\Corpus\Tutoriel_Unitex_CasSys\TexteXML\texteXML.txt*, en répondant *No* à la question *Do you want to preprocess the text*.

Créons un premier graphe qui sera enregistré (comme les suivants) dans le dossier *French\Graphs\Tutoriel_Unitex_CasSys\Texte\Analyse*.

2.1.1 Repérer les balises XML : graphe *toolXml.grf*

Pour simplifier, nous ne traitons ici que des balises <nom>…</nom>. Un graphe qui reconnait toutes les balises XML est disponible à l'intérieur de la cascade CasEN[21]. Le nom de la balise sera supposé être une séquence de lettres, reconnue par le code Unitex *<WORD>*. La catégorie de toutes les balises sera notée *baliseXml*. Cependant, pour garder le nom de la balise, nous allons utiliser une variable nom et indiquer aussi dans un trait s'il s'agit d'une balise ouvrante ou fermante. Ce trait sera donc *ouverture_nom* ou *fermeture_nom*.



Enregistrons et compilons ce graphe, puis ajoutons-le en première position à la cascade *analyse.csc*.

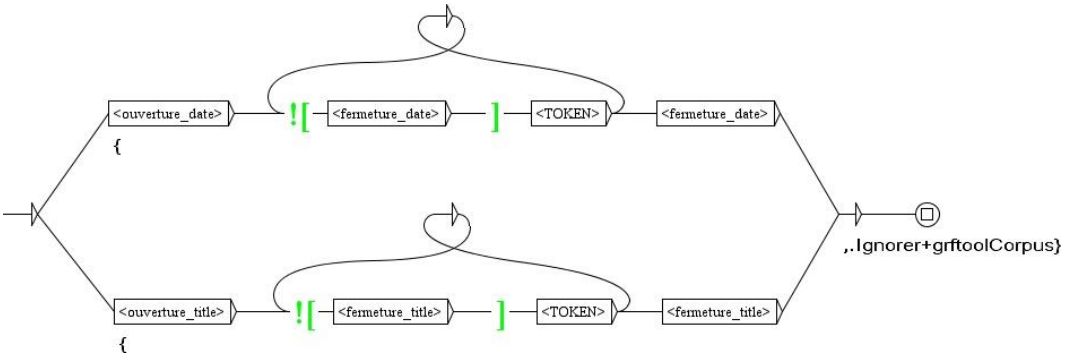
#	Disabled	Name	Merge	Replace	Until Fix Point	Generaliz...
1	☐	toolXml.fst2	☑	☐	☐	☐
2	☐	dateAbsolue.fst2	☑	☐	☐	☐
3	☐	dateRelative.fst2	☑	☐	☐	☐
4	☐	ville.fst2	☑	☐	☐	☐
5	☐	villeGeneralisation.fst2	☑	☐	☐	☑
6	☐	rue.fst2	☑	☐	☐	☐
7	☐	batiment.fst2	☑	☐	☐	☐

[20] C'est-à-dire qu'un fichier *Texte.xml* doit être renommé *Texte.txt* ou *Texte.xml.txt* avant utilisation sous Unitex.

[21] https://tln.lifat.univ-tours.fr/tln/version-francaise/navigation/ressources/casen/

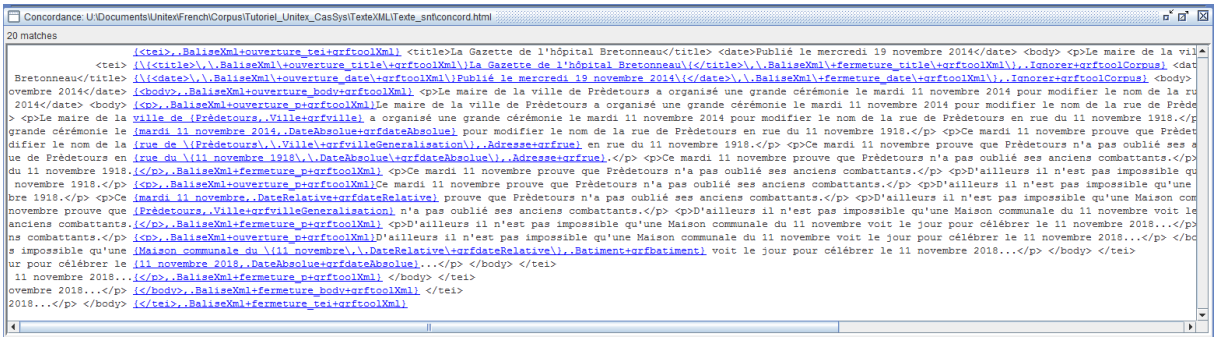
2.1.2 Cacher l'entête : graphe *toolCorpus.grf*

Dans l'exemple précédent, le batiment *hôpital Bretonneau* et la date *mercredi 19 novembre 2014* est reconnue dans l'entête alors qu'elle n'est pas dans le texte. Pour éviter cela, nous allons cacher le contenu des balises de l'entête, en les catégorisant *Ignorer*. En effet, les contenus de catégorie *Ignorer* ne seront pas traités par les graphes qui suivent.



Enregistrons et compilons ce graphe[22], puis ajoutons-le en deuxième position à la cascade *analyse.csc*.

#	Disabled	Name	Merge	Replace	Until Fix Point	Generaliz...
1	☐	toolXml.fst2	☑	☐	☐	☐
2	☐	toolCorpus.fst2	☑	☐	☐	☐
3	☐	dateAbsolue.fst2	☑	☐	☐	☐
4	☐	dateRelative.fst2	☑	☐	☐	☐
5	☐	ville.fst2	☑	☐	☐	☐
6	☐	villeGeneralisation.fst2	☑	☐	☐	☑
7	☐	rue.fst2	☑	☐	☐	☐
8	☐	batiment.fst2	☑	☐	☐	☐

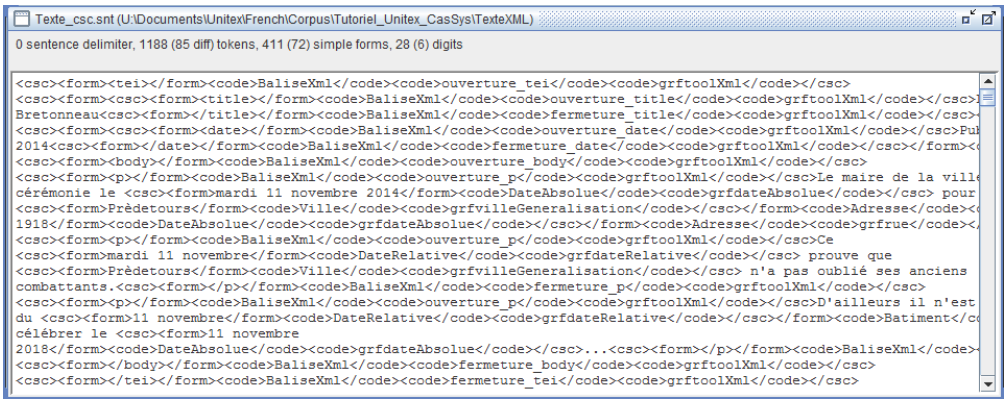


Remarque : nous avons utilisé le code Unitex *<ouverture_date>*, ce qui ne pose pas de problème, car il n'y a pas d'ambiguïté. Si le trait avait été nommé *ouverture*, le code *<ouverture>* aurait fonctionné aussi, mais aurait été ambigu avec le lexème *ouverture* (une *ouverture*, des *ouvertures*). Pour éviter cette ambiguïté, il faudrait utiliser le code *<BaliseXml+ouverture>*.

2.2 Cascade de synthèse

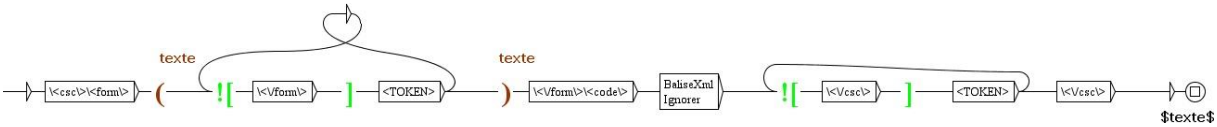
Ouvrons dans le menu *Text* le fichier *French\Corpus\Tutoriel_Unitex_CasSys\TexteXML\texteXML_csc.txt*, en répondant *No* à la question *Do you want to preprocess the text*.

[22] Si le texte que nous voulons cacher était trop long, ce graphe ne conviendrait pas. Il faudrait utiliser un graphe qui le cache petit à petit, avec l'option *Until Fix Point* cochée. Un exemple d'un tel graphe se trouve dans la cascade CasEN.



2.2.1 Graphe *suppressionCodes.grf*

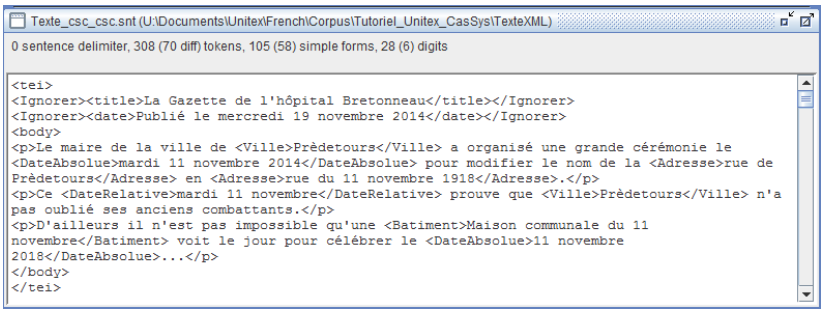
Créons un nouveau graphe qui sera enregistré (comme les suivants) dans le dossier *French\Graphs\Tutoriel_Unitex_CasSys\Texte\Synthese*. Ce graphe va supprimer les codes *BaliseXml* et *Ignorer* en récupérant le texte dans une variable *$texte$*.



Enregistrons et compilons ce graphe, puis ajoutons-le, en mode *Replace*, en première position à la cascade *analyse.csc*.

#	Disabled	Name	Merge	Replace	Until Fix Point	Generaliz...
1	☐	suppressionCodes.fst2	☐	☑	☐	☐
2	☐	balisage.fst2	☐	☑	☑	☐
3	☐	suppressionInterne.fst2	☐	☑	☐	☐

Si ce graphe n'est passé qu'une fois, il ne traite que les balises XML imbriquées dans les balises *Ignorer* :

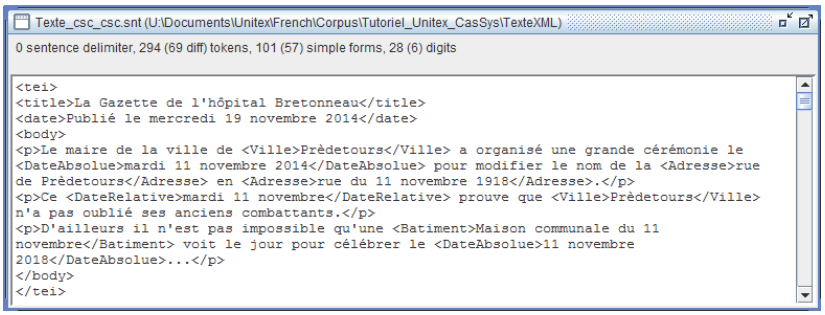


Pour que la suppression soit complète, il faut cocher la case *Until Fix Point* :

#	Disabled	Name	Merge	Replace	Until Fix Point	Generaliz...
1	☐	suppressionCodes.fst2	☐	☑	☑	☐
2	☐	balisage.fst2	☐	☑	☑	☐
3	☐	suppressionInterne.fst2	☐	☑	☐	☐

2.2.2 Résultat final

La deuxième cascade génère le fichier *texteBrut_csc_csc.txt* qui correspond à notre souhait.



3 Nombres et mesures

Le but de cet exercice est d'annoter les nombres et les mesures dans le roman de Jules Verne distribué avec Unitex ("Le tour du monde en 80 jours"). Nous allons réutiliser, parfois en les modifiant, les graphes du tutoriel *Prise en main d'Unitex pour l'annotation de corpus*, disponibles sur https://tln.univ-tours.fr/tln/version-francaise/navigation/ressources/tutoriels-unitex/. Mais, si vous avez téléchargé et dézippé le fichier de préparation dans votre dossier personnel Unitex, les graphes que nous allons réutiliser ont déjà été copiés au bon endroit, à savoir dans le dossier *French\Graphs\Tutoriel_Unitex_CasSys\Nombres\Analyse*.

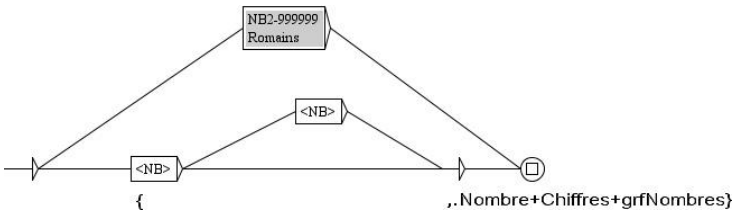
3.1 Cascade d'analyse

Commençons tout d'abord par ouvrir, dans le menu *Text*, le fichier intitulé *French\Corpus\80jours\80jours.txt*, en répondant *No* à la question *Do you want to preprocess the text*.

Créons un premier graphe qui sera enregistré (comme les suivants) dans le dossier intitulé *French\Graphs\Tutoriel_Unitex_CasSys\Nombres\Analyse*.

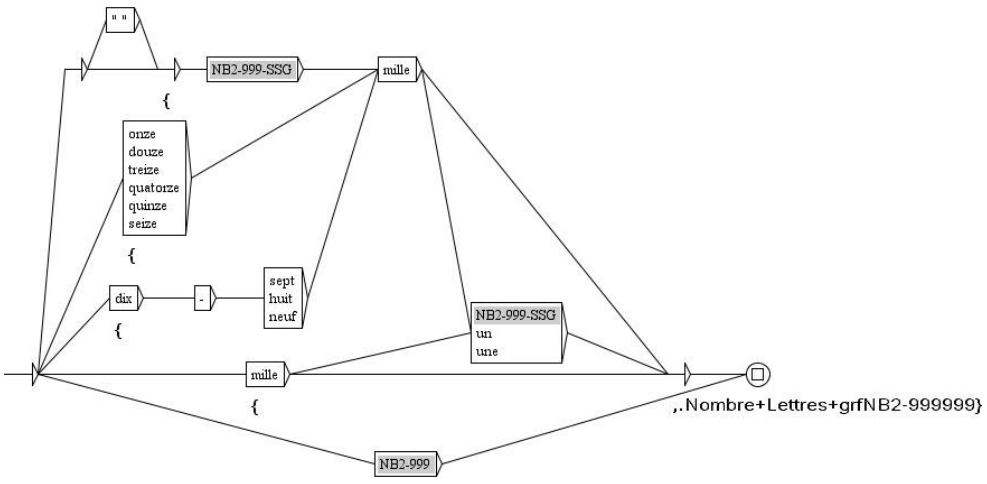
3.1.1 Graphe *Nombres.grf*

Le premier graphe de la cascade sera chargé de reconnaître et d'annoter les nombres entiers inférieurs au million, qu'ils soient écrits en chiffres arabes, en chiffres romains ou en toutes lettres. Rappelons que le symbole *<NB>* désigne une séquence de chiffres. Dans ce fichier, l'espace joue le role de séparateur de milliers.

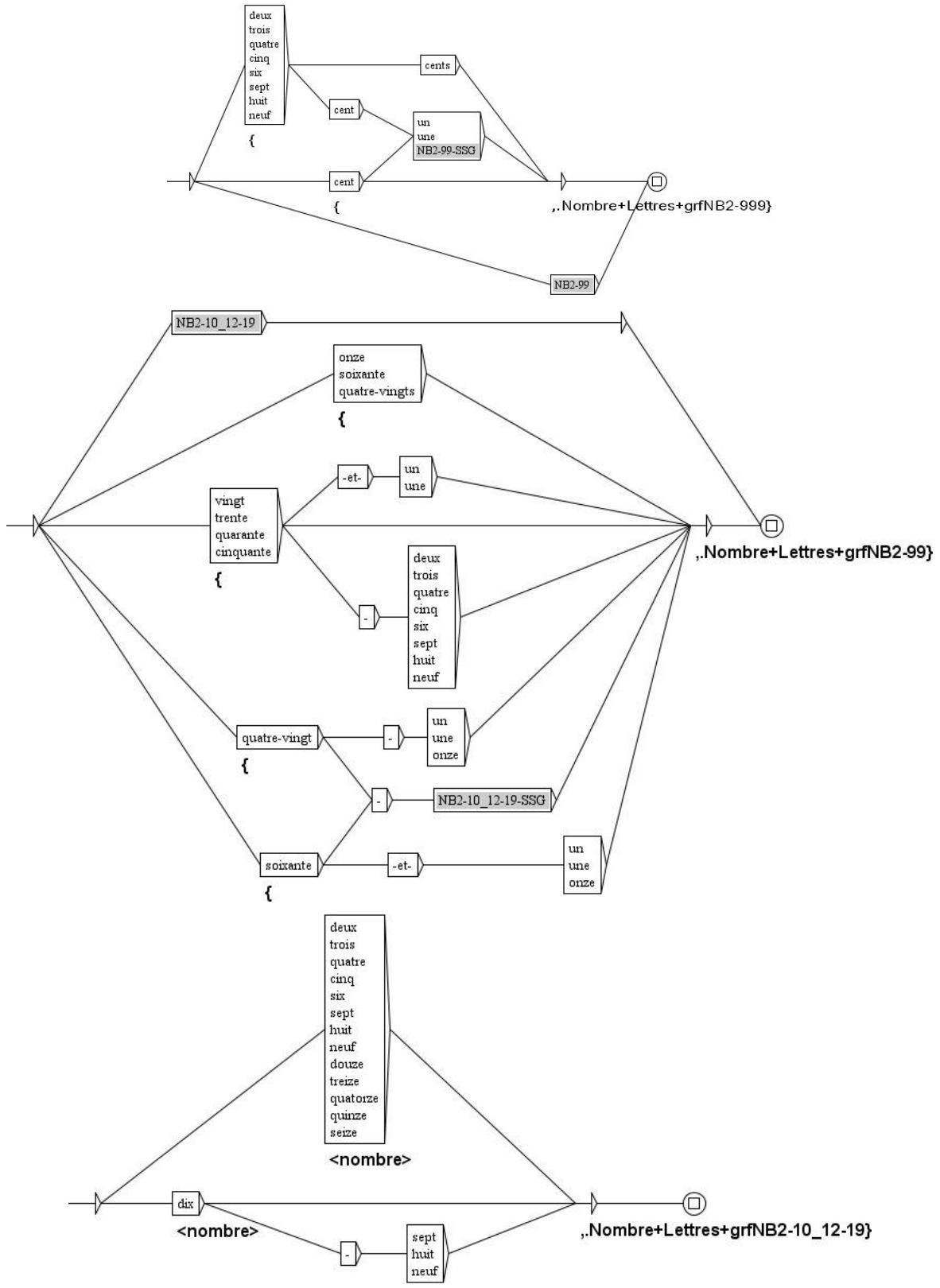


Les sous-graphes *NB2-999999* et *Romains* sont ceux construits dans le premier tutoriel. Nous allons les modifier rapidement en remplaçant les ouvertures de balise par *{* et les fermetures par *,.Nombre}*[23]. Dans la suite, les graphes suffixés *-SSG* sont les mêmes que les graphes non suffixés, mais sans sorties.

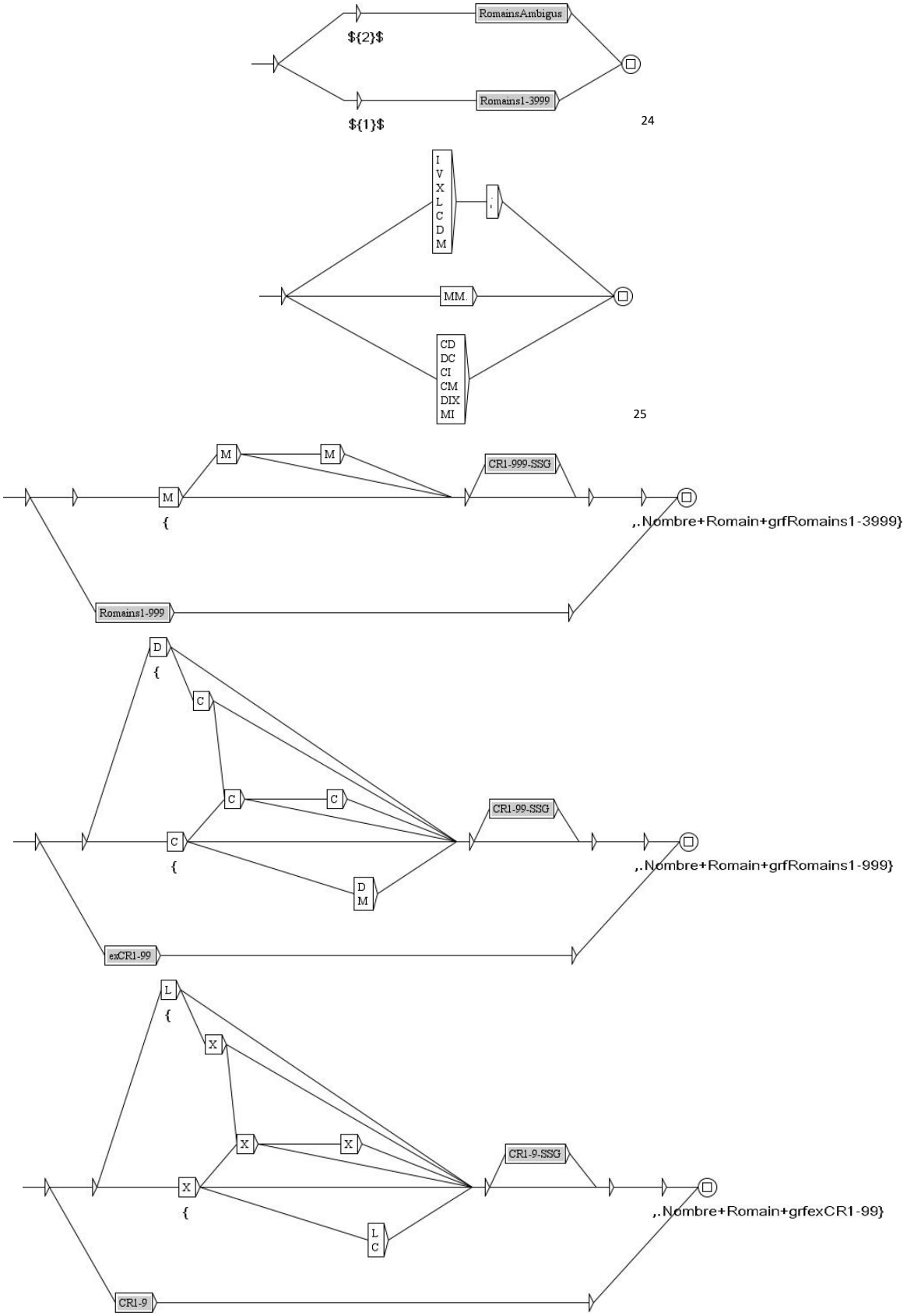
3.1.1.1 Sous-graphes des nombres écrits en toutes lettres



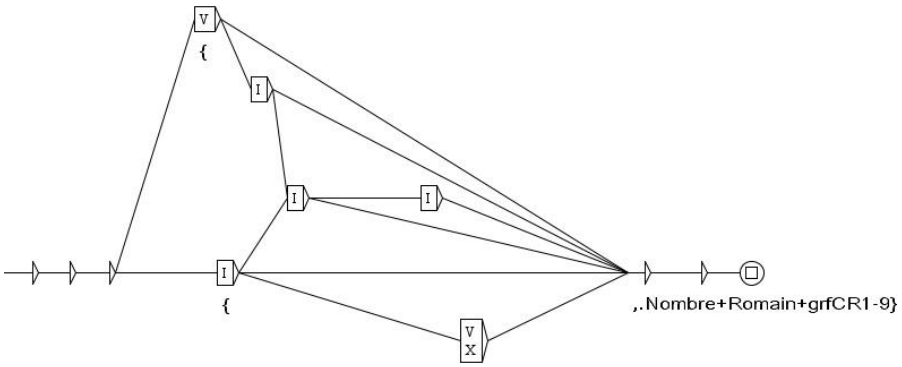
[23] Comme d'habitude, nous ajouterons des traits à ces catégories.


deux
trois
quatre
cinq
six
sept
huit
neuf
{
cents
cent
un
une
NB2-99-SSG
cent
{
,.Nombre+Lettres+grfNB2-999}
NB2-99
NB2-10_12-19
onze
soixante
quatre-vingts
{
-et-
un
une
vingt
trente
quarante
cinquante
{
,.Nombre+Lettres+grfNB2-99}
-
deux
trois
quatre
cinq
six
sept
huit
neuf
quatre-vingt
-
un
une
onze
{
-
NB2-10_12-19-SSG
soixante
-et-
un
une
onze
{
deux
trois
quatre
cinq
six
sept
huit
neuf
douze
treize
quatorze
quinze
seize
<nombre>
dix
<nombre>
-
sept
huit
neuf
,.Nombre+Lettres+grfNB2-10_12-19}

3.1.1.2 Sous-graphes des nombres écrits en chiffres romains



[24] Sous-graphe *Romains.grf*.
[25] Sous-graphe *RomainsAmbigus.grf*.



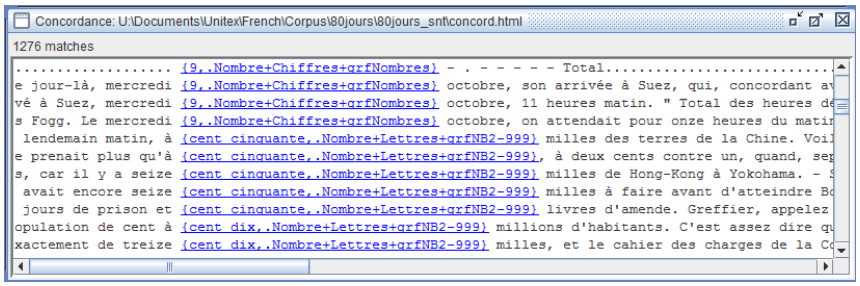
3.1.2 Création de la cascade d'analyse

Compilons le graphe *Nombres.grf*.

Ouvrons le menu *Text/Apply CasSys Cascade…* et cliquons sur le bouton *New*. Plaçons-nous dans le dossier *French\Graphs\Tutoriel_Unitex_CasSys\Nombres\Analyse*. Avec la souris, faisons glisser le graphe *Nombre.fst2* dans la partie droite de la fenêtre.

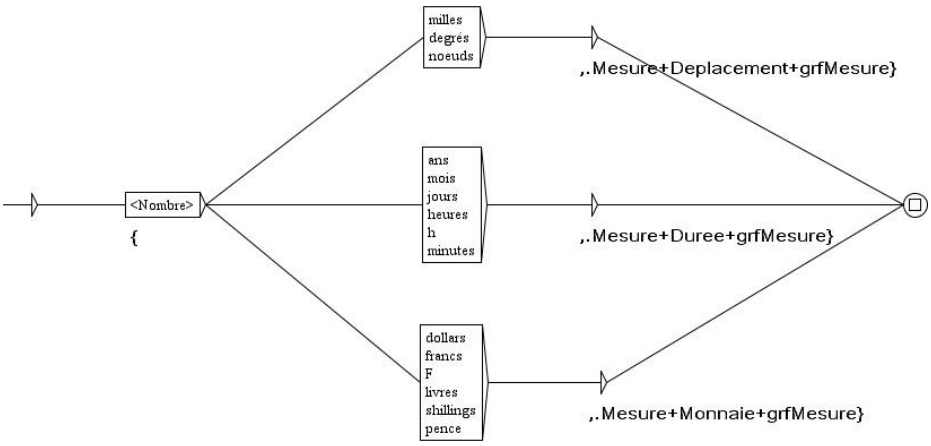
#	Disabled	Name	Merge	Replace	Until Fix Point	Generaliz...
1	☐	Nombres.fst2	☑	☐	☐	☐

Enregistrons cette cascade dans le dossier *French\CasSys\Tutoriel_Unitex_CasSys\Nombres* sous le nom *analyse.csc*. Fermons la fenêtre et cliquons sur le bouton *Launch*. Règlons la taille du contexte : *Left: 20 chars* et *Right: 255 chars*. Cliquons sur le bouton *Build concordance*.



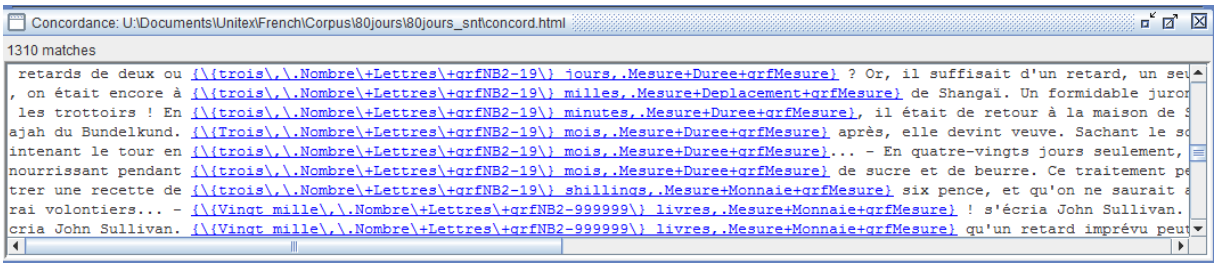
3.1.3 Graphe *mesure.grf*

Ce graphe, pour simplifier, sera *ad hoc* par rapport au roman. Un graphe des mesures est disponible dans la distribution de la cascade CasEN. Trois catégories de mesure apparaissent : les déplacements (*milles*, *degrés* et *noeuds*), la durée (*ans*, *mois*, *jours*, *heures*, *h*, *minutes*) et les monnaies (*dollars*, *francs*, *F*, *livres*, *shillings*, *pence*).



Enregistrons et compilons ce graphe, puis ajoutons-le en dernier à la cascade *analyse.csc*.

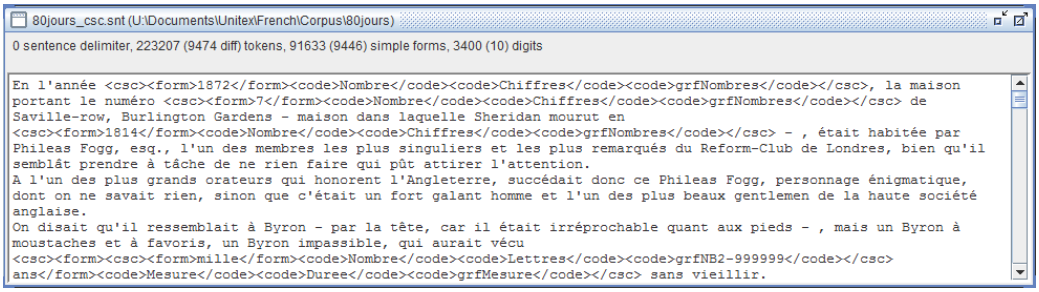
#	Disabled	Name	Merge	Replace	Until Fix Point	Generaliz...
1	☐	Nombres.fst2	☑	☐	☐	☐
2	☐	Mesure.fst2	☑	☐	☐	☐



3.2 Cascade de synthèse

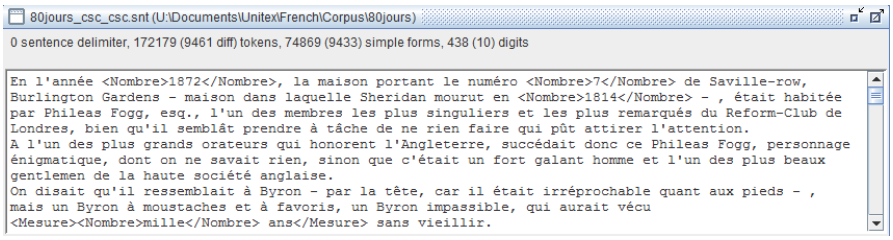
3.2.1 Création de la cascade

Ouvrons dans le menu *Text* le fichier *French\Corpus\80jours\80jours_csc.txt*, en répondant *No* à la question *Do you want to preprocess the text*.



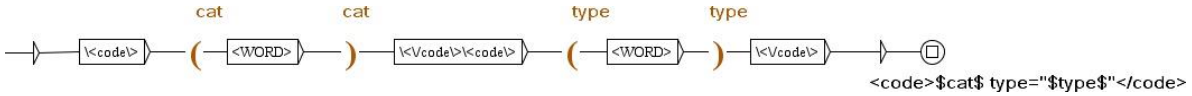
Dans le dossier *French\Graphs\Tutoriel_Unitex_CasSys\Texte\Synthese*, copions les graphes *balisage.grf* et *balisage.fst2*. Collons-les dans le dossier *French\Graphs\Tutoriel_Unitex_CasSys\Nombres\Synthese*. Ouvrons le menu *Text/Apply CasSys Cascade…* et cliquons sur le bouton *New*. Plaçons-nous dans le dossier *French\Graphs\Tutoriel_Unitex_CasSys\Nombres\Synthese*. Avec la souris, faisons glisser le graphe *balisage.fst2* dans la partie droite de la fenêtre et cochons *Replace* et *Unitil Fix Point* (car il y a des balises imbriquées).

Enregistrons cette cascade dans le dossier *French\CasSys\Tutoriel_Unitex_CasSys\Nombres* sous le nom *synthese.csc*. Fermons la fenêtre et cliquons sur le bouton *Launch*. Inutile de lancer la concordance, regardons plutôt le fichier *French\Corpus\80jours\80jours_csc_csc.txt*.



3.2.2 Le graphe *type.grf*

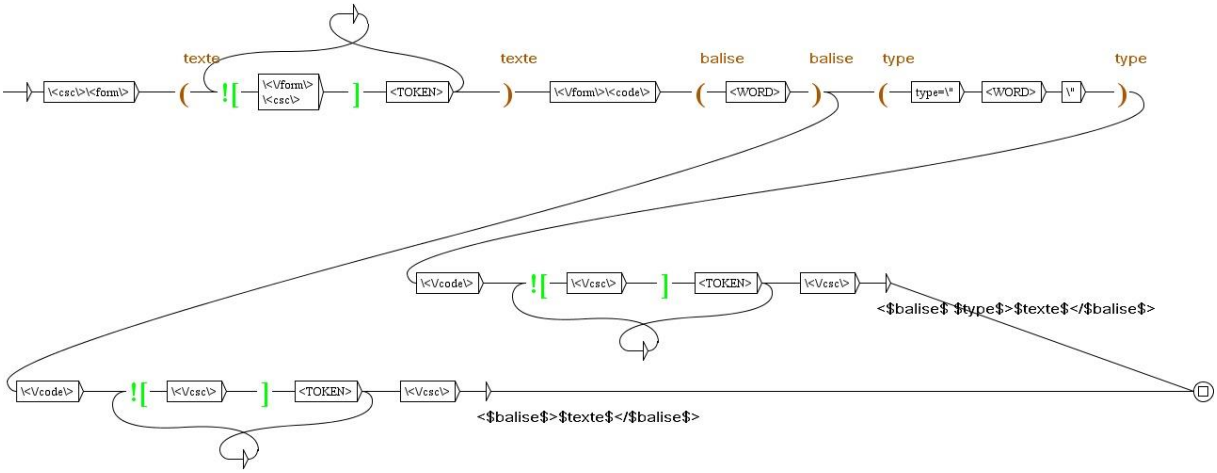
L'inconvénient de cette synthèse est que nous perdons toute l'information sur les catégories de nombre (lettres, chiffres, chiffres romains) et de mesure (déplacement, durée et monnaie). Pour éviter cela, nous allons créer un graphe type.grf qui va transformer le trait placé dans Unitex en un type XML. Par exemple, nous allons transformer *<code>Mesure</code><code>Monnaie</code>* en *<code>Mesure type="Monnaie"</code>*.



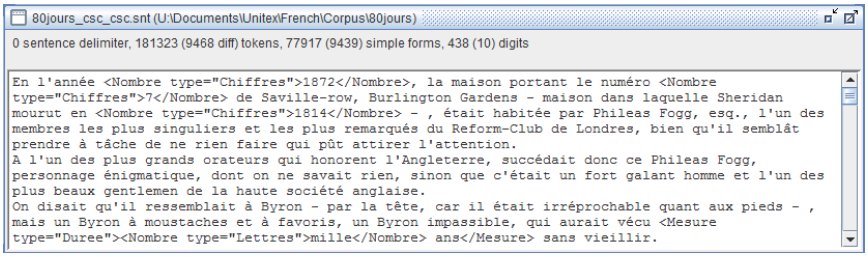
Enregistrons et compilons ce graphe, puis ajoutons-le en première position à la cascade *synthese.csc*, en mode Replace.

3.2.3 Le nouveau graphe *balisage.grf*.

Il nous faut alors compléter le graphe *balisage.grf* par l'éventuelle présence d'un type. Remarquons que, dans une boite, les guillemets doivent être protégés par un antislash : \"[26].



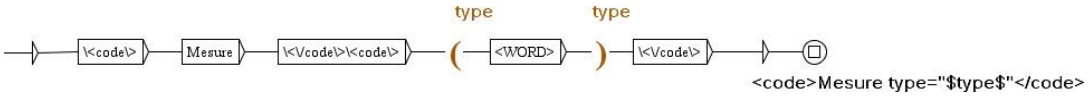
Enregistrons et compilons ce graphe.



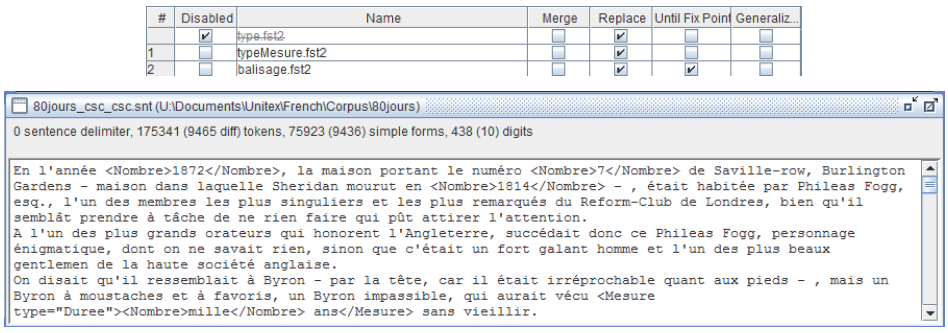
Un graphe plus élégant est présenté en annexe (section 0, page 25) pour les utilisateurs avancés.

3.2.4 Des types à placer uniquement sur les mesures

Montrons combien la cascade de synthèse permet un résultat "sur mesure", avec deux autres présentations des (mêmes) résultats : imaginons que nous ne souhaitions pas placer de type sur la balise Nombre, mais seulement sur la balise Mesure. Une solution simple serait d'écrire un graphe *typeMesure.grf* en remplaçant le code *<WORD>* par le mot *Mesure* et, donc, en supprimant la variable *cat*.



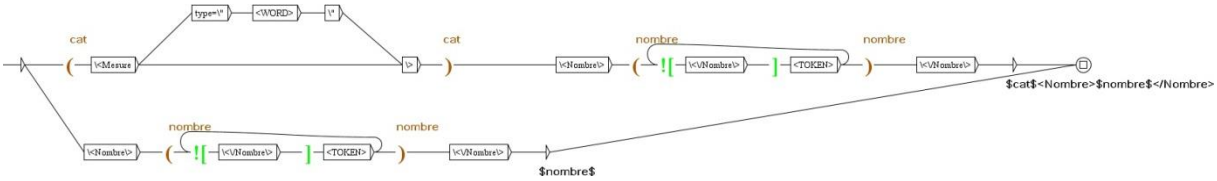
Enregistrons et compilons ce graphe, puis ajoutons-le en deuxième position à la cascade *synthese.csc*, en mode *Replace*. Sur la ligne du graphe *type.grf*, cochons la case *Disabled*, qui nous permet de choisir les graphes actifs ou non dans la cascade.



[26] Cette remarque ne concerne pas les sorties pour lesquelles aucune protection n'est nécessaire.

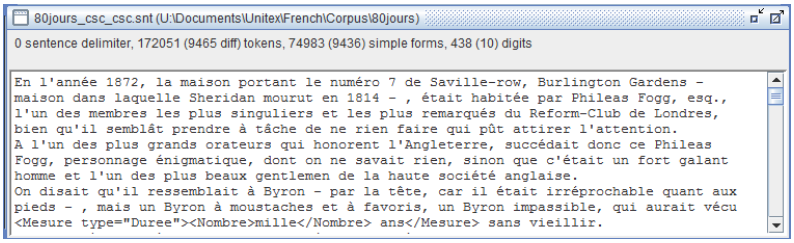
3.2.5 Suppression du balisage des nombres hors mesure

Ajoutons une contrainte : ne baliser les nombres que lorsqu'ils font partie d'une mesure. Le graphe *baliseNombre.grf* ne recopie les balises *<Nombres>* que si elles sont précédées d'une balise *<Mesure>*.



Enregistrons et compilons ce graphe, puis ajoutons-le en dernier à la cascade *synthese.csc*.

#	Disabled	Name	Merge	Replace	Until Fix Point	Generaliz...
	✔	~~type.fst2~~		✔		
1		typeMesure.fst2		✔		
2		balisage.fst2		✔	✔	
3		baliseNombre.fst2		✔		



Un graphe plus élégant est présenté en annexe (section 5.3, page 25) pour les utilisateurs avancés.

4 Annexe : création de scripts

L'interface d'Unitex est un outil de développement de graphes et de cascades de graphes. Une fois le travail de création réalisé et testé, il est possible de créer des scripts Unitex qui vont permettre de passer des cascades sur un corpus de texte[27]. Ces scripts Unitex sont lancés par une ligne de commande qui consulte un fichier compressé qui contient l'ensemble des ressources : scripts Unitex, graphes, cascades et dictionnaires.

Plaçons-nous dans notre dossier personnel Unitex et créons un dossier *Scripts*. Et, dans celui-ci un dossier *Script_texte*.

Créons, dans le dossier *Script_texte*, deux dossiers, *Origine* et *Retours* pour y placer deux textes à analyser et leurs résultats (le texte brut et le texte XML). Copions, dans le dossier *Origine*, le fichier *texteBrut.txt* du dossier *French**Corpus**Tutoriel_Unitex_CasSys**TexteBrut*, puis ensuite le fichier *texteXML.txt* du dossier *French**Corpus**Tutoriel_Unitex_CasSys**TexteXML*.

4.1 Création du *package linguistique*

Créons, dans le dossier *Script_texte*, un dossier pour lancer les cascades d'analyse et de synthèse de la section 2, page 10 : appelons-le *Script_texte_lingpkg*. Dans ce dossier, créons un dossier *resource* et un dossier *script*.

4.1.1 Le dossier *resource*

Dans le dossier *resource*, nous allons reproduire notre répertoire personnel. Pour cela, créons un dossier *French*. Copions dans ce dossier les trois fichiers : *Alphabet.txt*, *Alphabet_sort.txt* et *Norm.txt* qui se trouvent dans le dossier *French* du répertoire personnel Unitex. Dans ce dossier *resource**French*, créons trois répertoires : *CasSys*, *Dela* et *Graphs*.

Créons, dans le dossier *resource**French**CasSys*, des dossiers identiques à ceux du dossier *French**CasSys* de notre répertoire personnel. C'est-à-dire un dossier *Tutoriel_Unitex_CasSys* qui contient un dossier *Texte*. Puis copions les deux fichiers (*analyse.csc* et *synthese.csc*) du dossier *French**CasSys**Tutoriel_Unitex_CasSys**Texte* dans le dossier *resource**French**CasSys**Tutoriel_Unitex_CasSys**Texte*[28].

Créons, dans le dossier *resource**French**Graphs*, des dossiers identiques à ceux du dossier *French**Graphs*[29] de notre répertoire personnel. C'est-à-dire un dossier *Tutoriel_Unitex_CasSys* qui contient un dossier *Texte*. Puis copions les deux dossiers (*Analyse* et *Synthese*) du dossier *French**Graphs**Tutoriel_Unitex_CasSys**Texte* dans le dossier *resource**French**Graphs**Tutoriel_Unitex_CasSys**Texte*.

Nous n'avons pas utilisé de dictionnaires, nous n'avons donc rien à copier dans le dossier *resource**French**Dela*[30].

4.1.2 Le dossier *script*

Le dossier *script* contient le script Unitex[31]. Pour écrire celui-ci, nous allons nous inspirer de la console d'Unitex.

Si Unitex est déjà ouvert, il faut le fermer et le rouvrir (pour nettoyer la console). Lançons nos deux cascades :

1. Ouvrons (*Text/Open*) le fichier *French**Corpus**Tutoriel_Unitex_CasSys**TexteBrut**texteBrut.txt*, en répondant *No* à la question *Do you want to preprocess the text*.
2. Ouvrons le menu *Text/Apply CasSys Cascade…*, puis double-cliquons sur les dossiers *Tutoriel_Unitex_CasSys* et *Texte* ; cliquons sur la cascade *analyse.csc*, puis sur le bouton *Launch*.
3. Ouvrons (*Text/Open*) le fichier *French**Corpus**Tutoriel_Unitex_CasSys**TexteBrut**texteBrut_csc.txt*, en répondant *No* à la question *Do you want to preprocess the text*.
4. Ouvrons le menu *Text/Apply CasSys Cascade…*, puis cliquons sur la cascade *synthese.csc*, puis sur le bouton *Launch*.
5. Ouvrons le menu *Info/Console*, puis copions-la (*CTRL-C*).

[27] Voir le manuel, chapitre 13.
[28] On pourrait aussi, sans modifications, recopier directement les deux fichiers en question dans le dossier *resource**French**CasSys*.
[29] Il est aussi possible de ne pas créer ces deux dossiers et de copier les deux dossiers (*Analyse* et *Synthese*) du dossier *French**Graphs**Tutoriel_Unitex_CasSys**Texte* directement dans le dossier *resource**French**Graphs*. Mais, dans ce cas, il faut modifier les deux cascades en supprimant sur chaque ligne *Tutoriel_Unitex_CasSys**Texte*\.
[30] Sa présence n'est donc pas obligatoire ici. Mais il faut y penser si on utilise des dictionnaires.
[31] Plusieurs scripts peuvent être placés ici. C'est la ligne de commande qui contiendra le nom du script à lancer (voir section 4.2.1, page 23).

6. Collons ces instructions dans un fichier *tutoriel_texte.uniscript*, placé dans le dossier *script*.

Ce fichier contient une série d'instructions du genre :

```
mkdir "C:\Documents\Unitex\French\Corpus\Tutoriel_Unitex_CasSys\TexteBrut\texteBrut_snt"

"C:\Program Files (x86)\Unitex-GramLab\App\UnitexToolLogger.exe" Normalize "C:\Documents\Unitex\French\Corpus\Tutoriel_Unitex_CasSys\TexteBrut\texteBrut.txt" "-rC:\Documents\Unitex\French\Norm.txt" "--output_offsets=C:\Documents\Unitex\French\Corpus\Tutoriel_Unitex_CasSys\TexteBrut\texteBrut_snt\normalize.out.offsets" -qutf8-no-bom

"C:\Program Files (x86)\Unitex-GramLab\App\UnitexToolLogger.exe" Tokenize "C:\Documents\Unitex\French\Corpus\Tutoriel_Unitex_CasSys\TexteBrut\texteBrut.snt" "-aC:\Documents\Unitex\French\Alphabet.txt" -qutf8-no-bom

"C:\Program Files (x86)\Unitex-GramLab\App\UnitexToolLogger.exe" Cassys "-aC:\Documents\Unitex\French\Alphabet.txt" "-tC:\Documents\Unitex\French\Corpus\Tutoriel_Unitex_CasSys\TexteBrut\texteBrut.snt" "-lC:\Documents\Unitex\French\CasSys\Tutoriel_Unitex_CasSys\TexteBrut\analyse.csc" -v "-rC:\Documents\Unitex\French\Graphs\" "--input_offsets=C:\Documents\Unitex\French\Corpus\Tutoriel_Unitex_CasSys\TexteBrut\texteBrut_snt\normalize.out.offsets" -qutf8-no-bom

mkdir "C:\Documents\Unitex\French\Corpus\Tutoriel_Unitex_CasSys\TexteBrut\texteBrut_csc_snt"

"C:\Program Files (x86)\Unitex-GramLab\App\UnitexToolLogger.exe" Normalize "C:\Documents\Unitex\French\Corpus\Tutoriel_Unitex_CasSys\TexteBrut\texteBrut_csc.txt" "-rC:\Documents\Unitex\French\Norm.txt" "--output_offsets=C:\Documents\Unitex\French\Corpus\Tutoriel_Unitex_CasSys\TexteBrut\texteBrut_csc_snt\normalize.out.offsets" -qutf8-no-bom

"C:\Program Files (x86)\Unitex-GramLab\App\UnitexToolLogger.exe" Tokenize "C:\Documents\Unitex\French\Corpus\Tutoriel_Unitex_CasSys\TexteBrut\texteBrut_csc.snt" "-aC:\Documents\Unitex\French\Alphabet.txt" -qutf8-no-bom

"C:\Program Files (x86)\Unitex-GramLab\App\UnitexToolLogger.exe" Cassys "-aC:\Documents\Unitex\French\Alphabet.txt" "-tC:\Documents\Unitex\French\Corpus\Tutoriel_Unitex_CasSys\TexteBrut\texteBrut_csc.snt" "-lC:\Documents\Unitex\French\CasSys\Tutoriel_Unitex_CasSys\TexteBrut\synthese.csc" -v "-rC:\Documents\Unitex\French\Graphs\" "--input_offsets=C:\Documents\Unitex\French\Corpus\Tutoriel_Unitex_CasSys\TexteBrut\texteBrut_csc_snt\normalize.out.offsets" -qutf8-no-bom
```

Commençons par une harmonisation entre les systèmes Windows, MacOs et Unix :

1. Supprimons partout les guillemets
2. Remplaçons partout les antislash (\) par des slash (/)[32]

Supprimons partout le programme *UnitexToolLogger* et le chemin qui y mène, car c'est la ligne de commande (voir section 4.2.1, page 23) qui contiendra le chemin en question (*C:/Program Files (x86)/Unitex-GramLab/App/UnitexToolLogger.exe*).

Ajoutons quelques lignes correspondant à la gestion en mémoire des fichiers :

1. Quatre lignes au début du script[33]

```
CURRENT_WORK_DIR = {CORPUS_WORK_DIR}/{UNIQUE_VALUE}

DuplicateFile -p {CURRENT_WORK_DIR}

DuplicateFile --make-dir {CURRENT_WORK_DIR}/texte_snt

DuplicateFile -i {INPUT_FILE_1} {CURRENT_WORK_DIR}/texte.txt
```

2. Deux lignes au milieu du script[34] (après le premier appel à Cassys)

```
DuplicateFile --make-dir {CURRENT_WORK_DIR}/texte_csc_snt

DuplicateFile -i {CURRENT_WORK_DIR}/texte_csc.txt {CURRENT_WORK_DIR}/texte_csc.snt
```

3. Deux lignes à la fin du script[35]

[32] Cela est dû à une inversion de ces symboles entre les systèmes Windows, d'une part, et MacOs ou Unix, d'autre part. Si vous n'utilisez pas Windows, vous pouvez donc ignorer cette ligne.
[33] Dans le script, qui peut traiter plusieurs fichiers successivement (ou en parallèle), le fichier traité aura pour nom générique *texte.txt*.
[34] On prépare là le passage de la deuxième cascade.
[35] Pour l'envoi du fichier résultat et le nettoyage de la mémoire.

```
DuplicateFile -i {CURRENT_WORK_DIR}/texte_csc_csc.txt {OUTPUT_FILE_1}

DuplicateFile --recursive-delete {CURRENT_WORK_DIR}
```

Puis supprimons les deux lignes commençant par *mkdir* (remplacée par *DuplicateFile --make-dir*) et la deuxième ligne *Normalize* (le fichier *texte_csc.txt* étant déduit du fichier *texte.txt*, il est déjà normalisé, il ne faut pas relancer la normalisation). Il faut cependant dupliquer le fichier *texte_csc.txt* en *texte_csc.snt*, ce que fait la deuxième ligne ajoutée au milieu.

Pour rendre ce script général, nous allons transformer cette série d'instructions de la manière suivante :

1. Remplaçons partout *texteBrut* par *texte*
2. Remplaçons partout le chemin menant à nos fichiers[36]
   C:/Documents/Unitex/French/Corpus/Tutoriel_Unitex_CasSys/texte
   par
   {CURRENT_WORK_DIR}
3. Remplaçons partout le chemin menant à notre dossier personnel Unitex
   C:/Documents/Unitex
   par
   {PACKAGE_DIR}

Puisqu'il n'y a pas de deuxième normalisation, le fichier d'offset transmis au deuxième lancement de *cassys* est le fichier d'offset transmis par le premier lancement de *cassys*.

4. Remplaçons, sur la deuxième ligne *cassys*
   --input_offsets={CURRENT_WORK_DIR}/texte_snt/normalize.out.offsets
   par :
   --input_offsets={CURRENT_WORK_DIR}/texte_csc_txt_offsets.txt

Nous obtenons alors :

```
CURRENT_WORK_DIR = {CORPUS_WORK_DIR}/{UNIQUE_VALUE}

DuplicateFile -p {CURRENT_WORK_DIR}

DuplicateFile --make-dir {CURRENT_WORK_DIR}/corpus_snt

DuplicateFile --make-dir {CURRENT_WORK_DIR}/corpus_csc_snt

Normalize {CURRENT_WORK_DIR}/corpus.txt -r{PACKAGE_DIR}/French/Norm.txt --
output_offsets={CURRENT_WORK_DIR}/corpus_snt/normalize.out.offsets -qutf8-no-bom

Tokenize {CURRENT_WORK_DIR}/corpus.snt -a{PACKAGE_DIR}/French/Alphabet.txt -qutf8-no-bom

Cassys -a{PACKAGE_DIR}/French/Alphabet.txt -t{CURRENT_WORK_DIR}/corpus.snt -
l{PACKAGE_DIR}/French/CasSys/Tutoriel_Unitex_CasSys/TexteBrut/analyse.csc -v -r{PACKAGE_DIR}/French/Graphs/ --
input_offsets={CURRENT_WORK_DIR}/corpus_snt/normalize.out.offsets -qutf8-no-bom

Normalize {CURRENT_WORK_DIR}/corpus_csc.txt -r{PACKAGE_DIR}/French/Norm.txt --
output_offsets={CURRENT_WORK_DIR}/corpus_csc_snt/normalize.out.offsets -qutf8-no-bom

Tokenize {CURRENT_WORK_DIR}/corpus_csc.snt -a{PACKAGE_DIR}/French/Alphabet.txt -qutf8-no-bom

Cassys -a{PACKAGE_DIR}/French/Alphabet.txt -t{CURRENT_WORK_DIR}/corpus_csc.snt -
l{PACKAGE_DIR}/French/CasSys/Tutoriel_Unitex_CasSys/Nombres/synthese.csc -v -r{PACKAGE_DIR}/French/Graphs/ --
input_offsets={CURRENT_WORK_DIR}/texte_csc_txt_offsets.txt -qutf8-no-bom

DuplicateFile -i {CURRENT_WORK_DIR}/corpus_csc_csc.txt {OUTPUT_FILE_1}

DuplicateFile --recursive-delete {CURRENT_WORK_DIR}
```

Ce script va lancer successivement les deux cascades sur un fichier (ou plusieurs) et donner le résultat sans écrire tous les nombreux fichiers intermédiaires.

4.1.3 Compression

Le dossier *Script_texte_lingpkg* est maintenant complet. Il ne reste plus qu'à créer un fichier compressé.

[36] En laissant le slash final. Même remarque pour le remplacement suivant.

Pour cela, sélectionnons les deux dossiers *resource* et *script*, puis compressons-les[37] et nommons le résultat *Script_texte_lingpkg.zip*. Ce fichier compressé doit être placé dans le dossier *Script_texte*.

4.2 Ligne de commande

4.2.1 Fichier de lancement

Créons maintenant, dans le dossier *Script_texte*, un fichier contenant la ligne de commande suivante (à adapter à votre ordinateur)[38] :

```
"C:\Program Files (x86)\Unitex-GramLab\App\UnitexToolLogger.exe" { BatchRunScript -i .\Origine -e -o .\Retours -t1 .\Script_texte_lingpkg.zip -f -s script\tutoriel_texte.uniscript }
```

Enregistrons-le sous le nom *multiLancementTexte.bat*[39].

4.2.2 Lancement

Ouvrons une fenêtre de commande, plaçons-nous dans le dossier *Script_texte* et lançons la commande *multiLancementTexte.bat*.

Nous obtenons dans le dossier *Retours* deux fichiers : *texteBrut.result.txt* et *texteXML.result.txt*.

4.3 Exercice

Créons, dans le dossier *Scripts*, un dossier *Script_nombres*, pour y placer un script concernant les cascades de la section 3 page 13.

Le script obtenu est le même que le précédent, à l'exception des chemins menant aux cascades sur les lignes Cassys :

```
Cassys -a{PACKAGE_DIR}/French/Alphabet.txt -t{CURRENT_WORK_DIR}/corpus.snt -l{PACKAGE_DIR}/French/CasSys/Tutoriel_Unitex_CasSys/Nombres/analyse.csc -v -r{PACKAGE_DIR}/French/Graphs/ --input_offsets={CURRENT_WORK_DIR}/corpus_snt/normalize.out.offsets -qutf8-no-bom
```

Et :

```
Cassys -a{PACKAGE_DIR}/French/Alphabet.txt -t{CURRENT_WORK_DIR}/corpus_csc.snt -l{PACKAGE_DIR}/French/CasSys/Tutoriel_Unitex_CasSys/Nombres/synthese.csc -v -r{PACKAGE_DIR}/French/Graphs/ --input_offsets={CURRENT_WORK_DIR}/texte_csc_txt_offsets.txt -qutf8-no-bom
```

Voici la ligne de commande correspondante :

```
"C:\Program Files (x86)\Unitex-GramLab\App\UnitexToolLogger.exe" { BatchRunScript -i .\Origine -e -o .\Retours -t1 .\Script_nombres_lingpkg.zip -f -s script\tutoriel_nombres.uniscript }
```

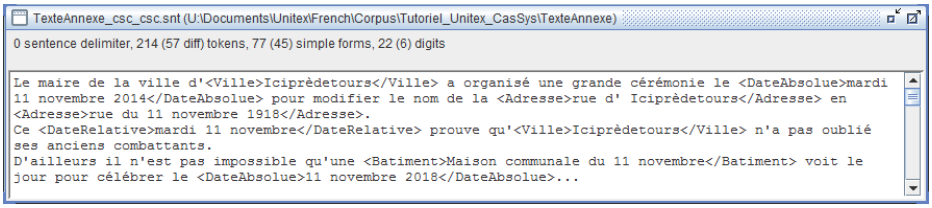
[37] Attention de ne pas compresser le dossier *Script_texte_lingpkg* à sa racine, car on obtiendrait un niveau supplémentaire dans le fichier compressé et rien ne fonctionnerait.
[38] Attention : l'accolade ouvrante est obligatoirement suivie d'un espace et l'accolade fermante est obligatoirement précédée d'un espace.
[39] L'option *–t1* signifie qu'on utilise un seul cœur. Si votre ordinateur contient plusieurs cœurs et si le nombre de fichiers est important, il est possible de lancer le script sur plusieurs fichiers en parallèle, en modifiant cette option en *–t2*, *-t3*…

5 Annexe pour utilisateurs avancés

5.1 Complément de la section 1.2.2, page 8

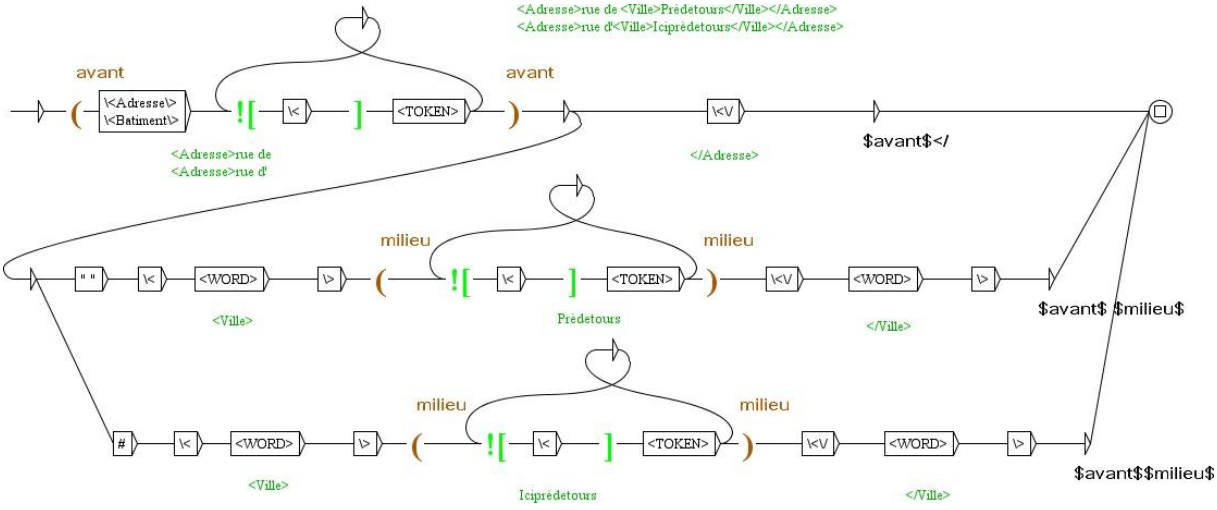
Reprenons la première cascade de synthèse et le graphe *suppressionInterne*. Un problème va surgir s'il n'y a pas d'espace entre les variables $avant$ et $milieu$, par exemple dans le cas d'une apostrophe. Considérons le même texte où le nom de la ville a été changé en *Iciprèdetours*. Ce texte se trouve[40] dans le dossier *French\Corpus\Tutoriel_Unitex_CasSys\TexteAnnexe*.

Pour lancer l'analyse, commençons par ouvrir, toujours dans le menu *Text*, le fichier *French\Corpus\Tutoriel_Unitex_CasSys\TexteAnnexe\texteAnnexe.txt*, en répondant *No* à la question *Do you want to preprocess the text*. Puis passons la cascade d'analyse. Ouvrons ensuite, dans le menu *Text*, le fichier *French\Corpus\Tutoriel_Unitex_CasSys\TexteAnnexe\texteAnnexe_csc.txt*, en répondant *No* à la question *Do you want to preprocess the text*. Passons la cascade de synthèse. Le résultat obtenu est le fichier *texteAnnexe_csc_csc.txt*

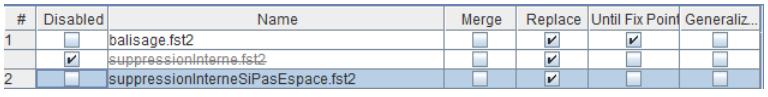


Une erreur s'est produite : *<Adresse>rue d' Iciprèdetours</Adresse>* avec un espace entre l'apostrophe et le nom.

Une première solution pour éviter cela consiste à créer un graphe *suppressionInterneSiPasEspace.grf* qui duplique les chemins en fonction ou non de la présence ou de l'absence de l'espace, en utilisant le symbole *#* qui signifie "pas d'espace entre les boites"[41] et le code " " qui désigne un espace.



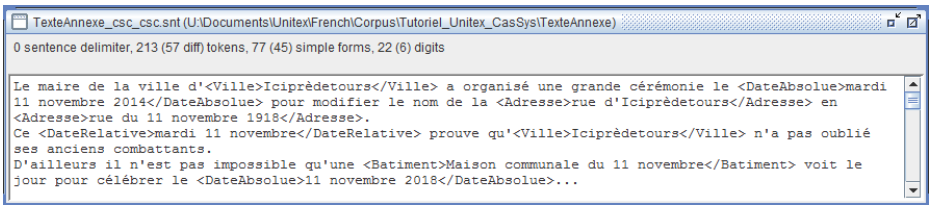
Si le graphe *suppressionInterneSiPasEspace* remplace dans la cascade le graphe *suppressionInterne*, nous obtenons le résultat souhaité, *<Adresse>rue d'Iciprèdetours</Adresse>* sans espace entre l'apostrophe et le nom.



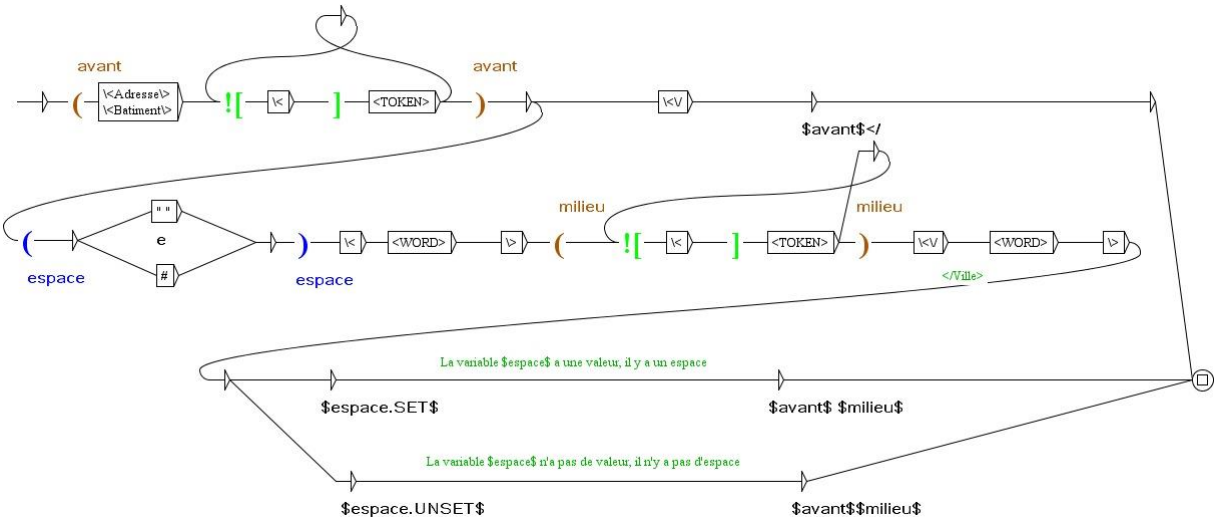
#	Disabled	Name	Merge	Replace	Until Fix Point	Generaliz...
1	☐	balisage.fst2	☐	☑	☑	☐
	☑	~~suppressionInterne.fst2~~	☐	☑	☐	☐
2	☐	suppressionInterneSiPasEspace.fst2	☐	☑	☐	☐

[40] Si vous n'avez pas pu télécharger la préparation du tutoriel, il faut tout d'abord se placer dans le dossier *Corpus/Tutoriel_Unitex_CasSys* et y créer un dossier nommé *TexteAnnexe*. Puis copier le fichier *texteBrut.txt* du dossier *TexteBrut* et le coller dans le dossier *TexteAnnexe*, en changeant son nom en *texteAnnexe.txt*. Ouvrir ensuite ce fichier via l'éditeur d'Unitex (menu *File Edition/New File*) et remplacer *Prèdetours* par *Iciprèdetours*. Et, enfin, enregistrer le fichier et fermer l'éditeur.

[41] Voir le manuel, section 4.3.1.



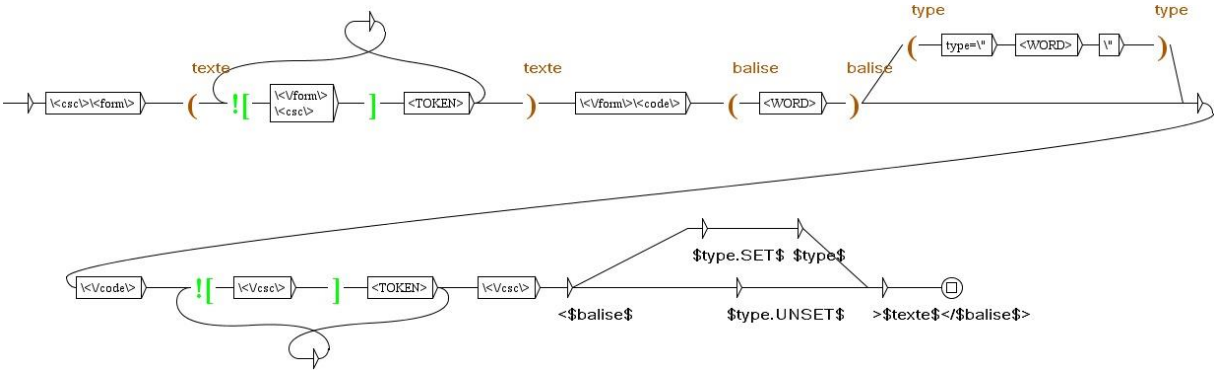
Une autre solution, plus élégante, mais plus complexe, permet d'éviter la duplication des chemins : il s'agit de faire un test sur une variable de sortie[42]. Le test qui permet de savoir si la variable *$espace$* possède une valeur ou non utilise les sorties *$espace.SET$* et *$espace.UNSET$*[43].



Si le graphe *suppressionInterneAvecTest* remplace dans la cascade le graphe *suppressionInterneSiPasEspace*, nous obtenons à nouveau le résultat souhaité, *<Adresse>rue d'Iciprèdetours</Adresse>* sans espace entre l'apostrophe et le nom.

5.2 Complément de la section 3.2.3, page 18

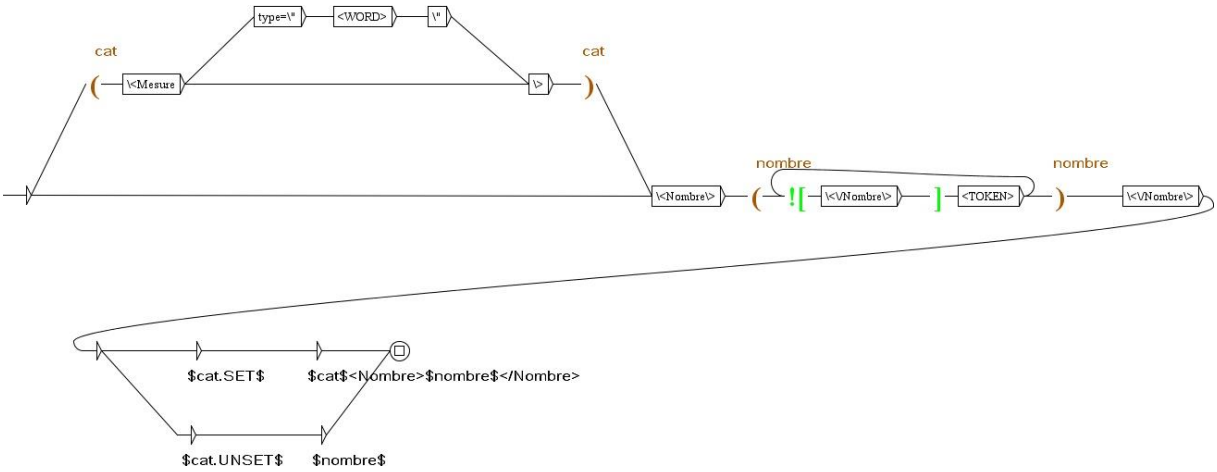
De même, on pourrait remplacer le graphe *balisage.grf* par un graphe plus élégant, en utilisant un test plutôt qu'une répétition. C'est le graphe *balisageTest.grf*.



5.3 Complément de la section 3.2.5, page 19

Comme à la section précédente, nous pouvons utiliser un test.

[42] Voir le manuel, section 6.8.
[43] Voir le manuel, section 6.9.1.


cat
type=\"
<WORD>
\"
\<Mesure
\>
cat
\<Nombre\>
nombre
!
\<\/Nombre\>
]
<TOKEN>
nombre
\<\/Nombre\>
$cat.SET$
cat<Nombre>$nombre$</Nombre>
$cat.UNSET$
$nombre$