Tutoriel Prise en main d'Unitex 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

Préparation

Installation

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.

Quelques mots rapides sur Unitex

Une fois le logiciel installé, le manuel Unitex est disponible via le menu *Help/Manuals*.

Ouvrir un texte : voir le manuel, section 2.4-2.5.5.

Créer un graphe : voir le manuel, section 5.2.

En bref, pour créer un graphe, il faut retenir les trois points suivants :

1. Créer (ou supprimer) un chemin entre deux boites : cliquer sur la première, puis sur la deuxième boite.
2. Créer une boite : clic-droit de la souris et choisir *Create box*. Si une boite est sélectionnée, un chemin sera automatiquement tracé entre la boite sélectionnée et la nouvelle boite.

3. Remplir une boite : écrire sur la barre de formule et valider. Si la barre de formule est totalement vide, la boite est supprimée. Le symbole *<E>* désigne l'élément vide (*empty*) et permet de créer une boite vide.

Préparation

Le plus simple est de télécharger le fichier :
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 l'ouverture du corpus.

En cas d'impossibilité, poursuivre ci-dessous.

Création de dossiers

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

1. Dans *Corpus* : Créons un dossier nommé *80jours* et glissons dans ce dossier le fichier *80jours.txt* ("Le tour du monde en 80 jours" de Jules Verne), distribué avec Unitex ;
2. Dans *Graph* : Créons un premier dossier pour l'ensemble du tutoriel, nommé *Tutoriel_Unitex*. Créons, dans ce dossier, trois dossiers nommés *Avoir*, *Nombres* et *Romains*.

Ouverture du corpus

Commençons par ouvrir, dans le menu *Text*, le fichier *French/Corpus/80jours/80jours.txt*, en répondant *Yes* à la question *Do you want to preprocess the text* et en ne laissant que l'option *Apply all default Dictionaries*.

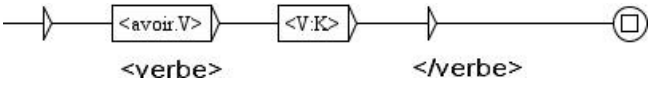
1 Sujet : annoter *avoir suivi d'un participe passé*

1.1 Le plus simple

Créons un premier graphe qui sera enregistré (comme les suivants) dans le dossier *French/Graphs/Tutoriel_Unitex/Avoir*. Ce graphe va reconnaitre toutes les formes du verbe *avoir* suivi de n'importe quelle forme d'un participe passé, en insérant le code XML *<verbe>…</verbe>* pour les formes reconnues.

En principe, le code *<avoir>* désigne toutes les formes du lexème *avoir*… Mais il existe deux lexèmes *<avoir>*, le verbe, mais aussi le nom (*un avoir*, *des avoirs*). Pour éviter de reconnaître *avoirs* comme une forme verbale, nous devons préciser la catégorie avec le code *<avoir.V>*[1].

L'information morphologique correspondant au participe passé, dans le dictionnaire distribué avec Unitex, est *K*. Le code permettant de reconnaître n'importe quelle forme d'un verbe au participe passé est donc *<V:K>*.

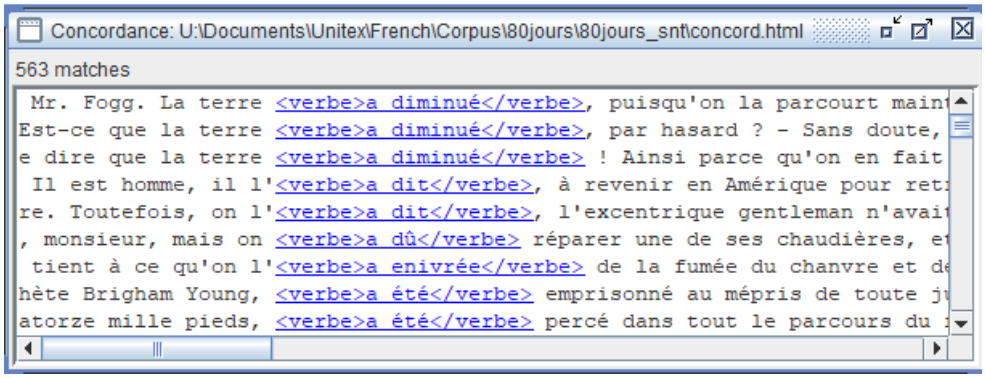


[1] Attention à respecter la casse à l'intérieur des codes Unitex.

Voir la vidéo : https://tln.lifat.univ-tours.fr/medias/video/avoirvk1_1580208610930-mp4?ID_FICHE=334589&INLINE=FALSE.

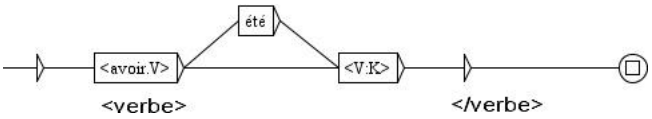
Enregistrons ce graphe dans le dossier *French/Graphs/Tutoriel_Unitex /Avoir* sous le nom *AvoirVK.grf*.

Lançons la recherche d'occurrences (*Text/Locate patterns*) en utilisant le graphe *AvoirVK.grf* (bouton *Set* de la ligne *Graph*) avec les options *Merge with input text* et *Index all occurrences in text*. Enfin, cliquons sur le bouton *Build concordance*.

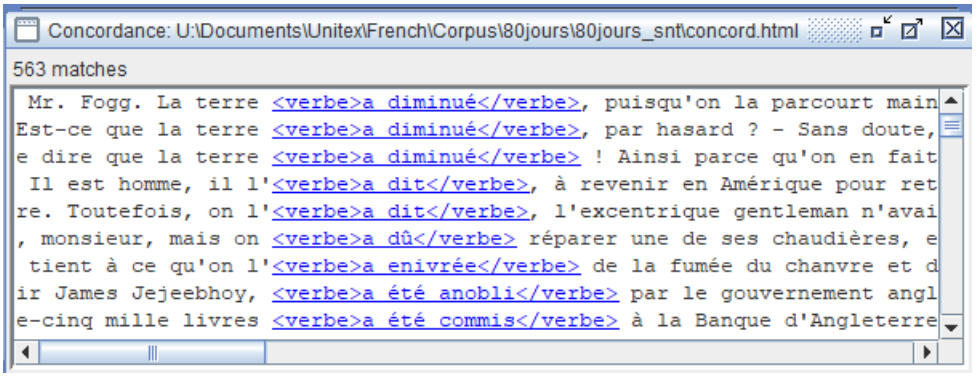


1.2 Prise en compte des passifs dans l'annotation

On constate ci-dessus que dans l'occurrence *a été emprisonné*, le participe passé *emprisonné* est à l'extérieur des balises. Corrigeons cela en créant un chemin optionnel comportant le mot *été* (et non le lexème *<été>* - un *été*, des *étés*).



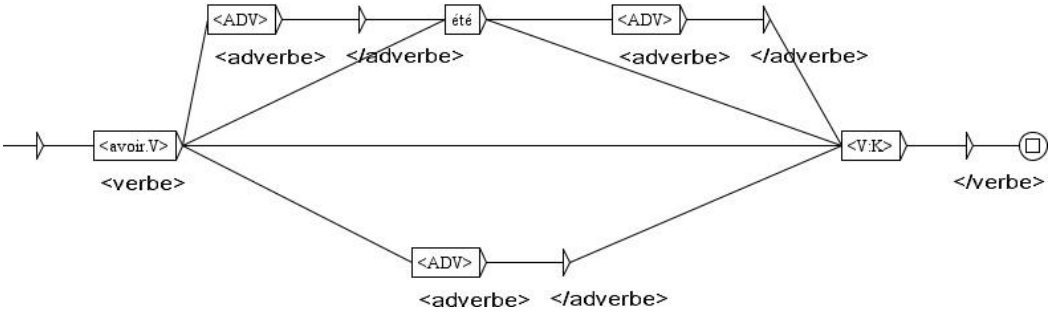
Voir la vidéo : https://tln.lifat.univ-tours.fr/medias/video/avoirvk2_1580208688695-mp4?ID_FICHE=334589&INLINE=FALSE.



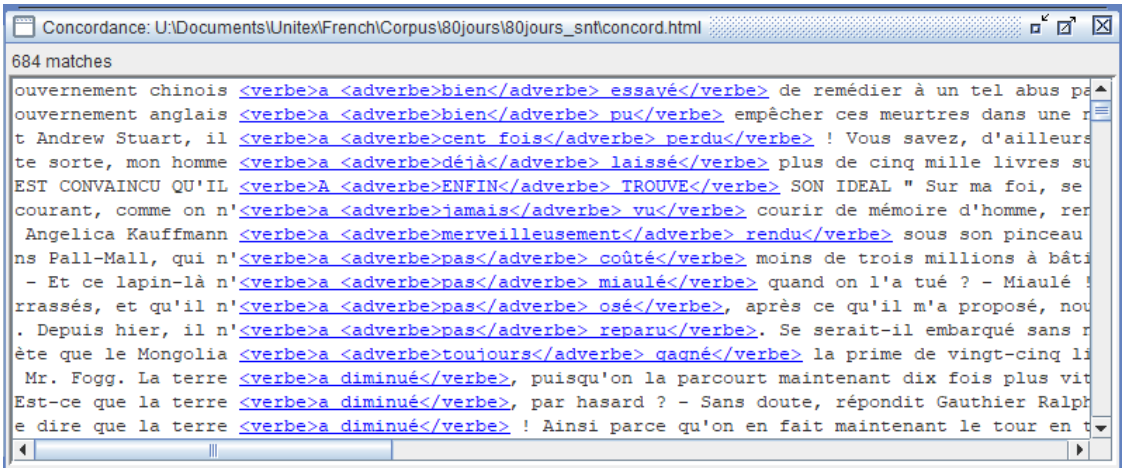
1.3 Prise en compte de l'insertion d'un adverbe

Prenons en compte maintenant qu'un adverbe peut être inséré entre le verbe *avoir* et le participe passé, mais aussi entre le verbe *avoir* et le mot *été* ou entre le mot *été* et le participe passé. L'adverbe sera balisé en utilisant le code XML *<adverbe>…</adverbe>*. Comme il s'agit de faire trois fois la même chose, nous allons utiliser des copiers-collers.

L'information catégorielle correspondant à l'adverbe, dans le dictionnaire distribué avec Unitex, est *ADV*. Le code permettant de reconnaître n'importe quel adverbe est donc *<ADV>*.

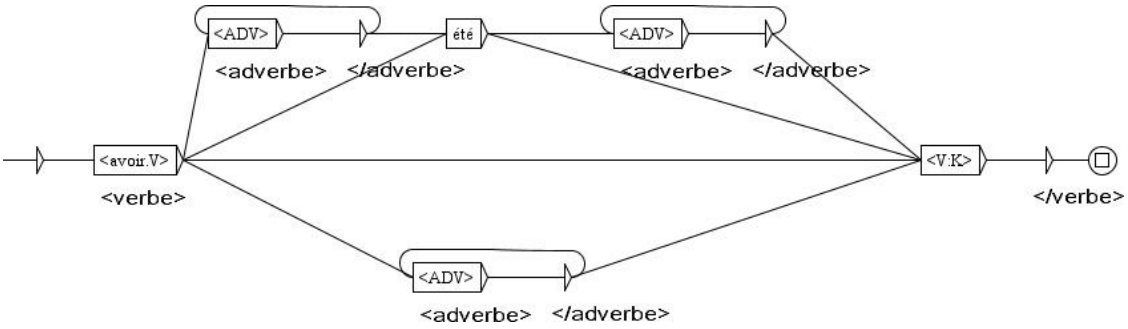


Voir la vidéo : https://tln.lifat.univ-tours.fr/medias/video/avoirvk3_1580215776691-mp4?ID_FICHE=334589&INLINE=FALSE.

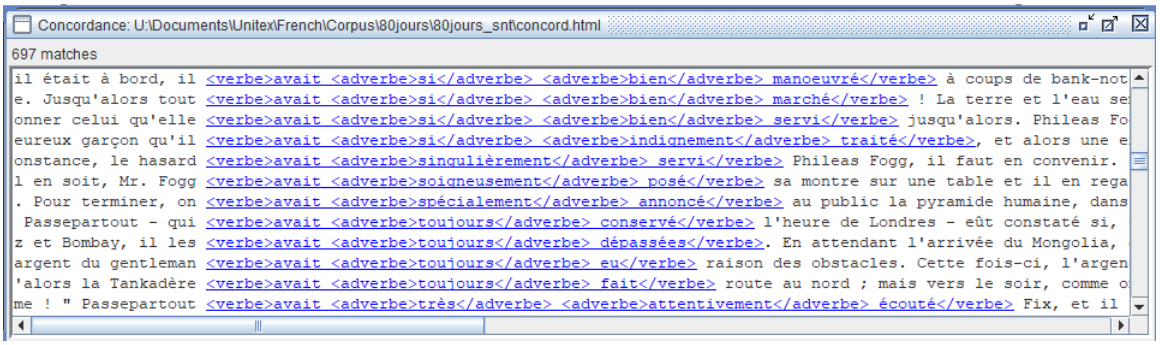


1.4 Prise en compte de plusieurs adverbes

Parfois, il arrive qu'il n'y ait pas un, mais plusieurs adverbes insérés. Utilisons un chemin en forme de boucle pour les reconnaitre. Le nombre de boucles n'est pas limité. Cela pourrait se faire (voir le manuel, section 6.2.4.).



Voir la vidéo : https://tln.lifat.univ-tours.fr/medias/video/avoirvk4_1580216053093-mp4?ID_FICHE=334589&INLINE=FALSE.

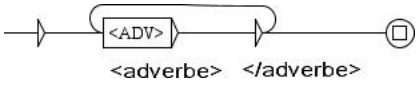


1.5 Un sous-graphe

Petit à petit le graphe se complexifie. Pour le garder lisible, nous allons transformer les parties redondantes en un sous-graphe. Pour cela, on en sélectionne une et, par un clic-droit, on choisit *Export as new graph*.

1.5.1 Sous graphe des adverbes

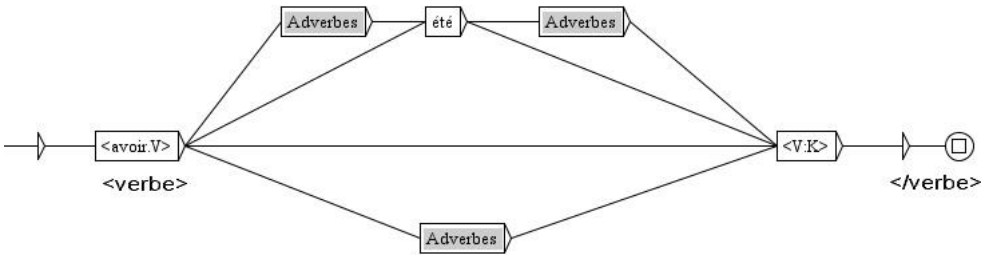
Ce sous-graphe sera enregistré sous le nom *Adverbes.grf*, dans le même répertoire que le graphe principal. On pourrait aussi choisir d'avoir un répertoire pour tous les sous-graphes, ce qui n'est intéressant que si ces sous-graphes sont communs à plusieurs projets dont les graphes sont placés dans des répertoires différents (voir le manuel, section 5.2.2.).



Voir la vidéo : https://tln.lifat.univ-tours.fr/medias/video/adverbes_1580216129838-mp4?ID_FICHE=334589&INLINE=FALSE.

1.5.2 Graphe principal

L'appel à un sous-graphe est constitué du caractère *:* suivi du nom du graphe sans son extension. La boite devient rouge si le sous-graphe (ou le graphe appelant) n'est pas enregistré. Grise, sinon.

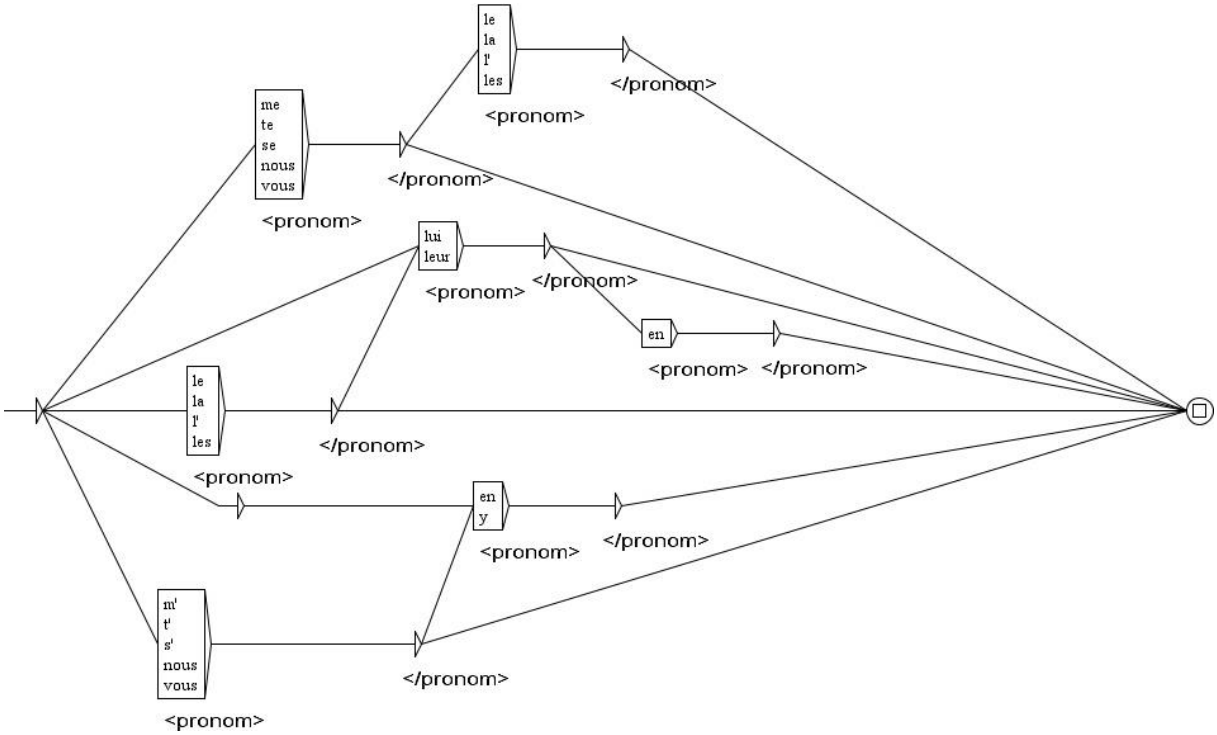


Voir la vidéo : https://tln.lifat.univ-tours.fr/medias/video/avoirvk5_1580216201991-mp4?ID_FICHE=334589&INLINE=FALSE.

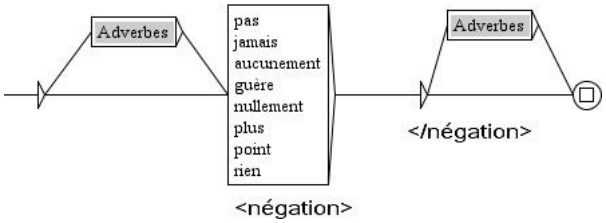
1.6 Formes négatives

Pour introduire les formes négatives dans notre graphe, nous allons avoir besoin de décrire les pronoms préverbaux dans un sous-graphe (qu'on va créer directement par le menu *FSGraph/New*), puis d'un sous-graphe contenant la liste des négations. Pour obtenir plusieurs lignes dans une boîte, on les sépare par le caractère *+*.

1.6.1 Graphe Preverbaux.grf



1.6.2 Graphe Negation.grf



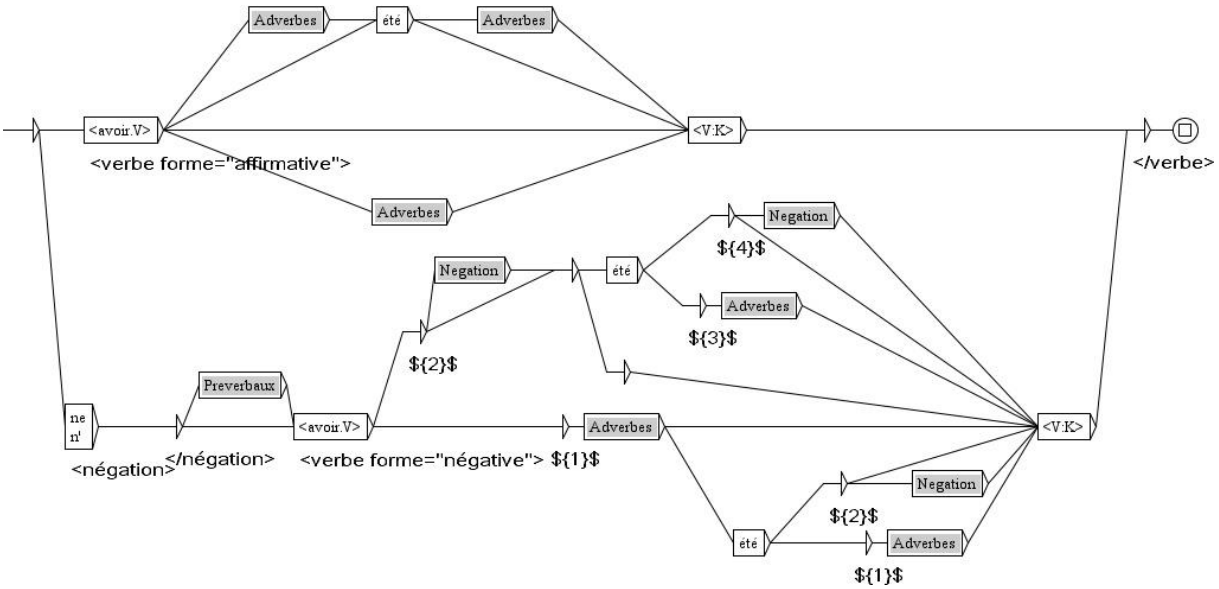
1.6.3 Graphe principal

Utilisation des poids : voir le manuel, section 5.2.4.

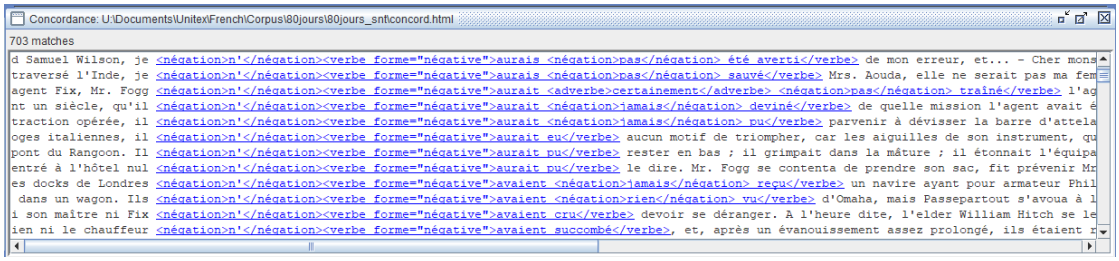
Ensuite, il nous faut remplacer la possibilité d'un adverbe par celle d'un adverbe ou d'une négation. Mais, comme la plupart des négations sont des adverbes, il faut privilégier le chemin contenant le sous-graphe *Negation.grf* par rapport au sous-graphe *Adverbes.grf*.

D'une manière générale, Unitex privilégie le chemin le plus long (en nombre de *tokens*[2]). Mais à nombre de *tokens* identique, le choix d'un chemin par Unitex est imprévisible. Dans ce cas, il faut discriminer par des poids. Un poids est une sortie notée par un chiffre placé à l'intérieur des symboles *$\{* et *\}$*. Le poids final le plus fort sur un chemin l'emporte. À cause de cette règle, le graphe des négations aura des poids variant de 1 à 4.

[2] Un *token*, pour Unitex, est soit une séquence de lettres, soit un autre caractère qui n'et pas une lettre et qui constitue un *token* à lui seul. Par exemple *en 2001* contient 5 *tokens* ("en", "2", "0","0" et "1").



Voir la vidéo : https://tln.lifat.univ-tours.fr/medias/video/adverbesounegation_1580216275478-mp4?ID_FICHE=334589&INLINE=FALSE.

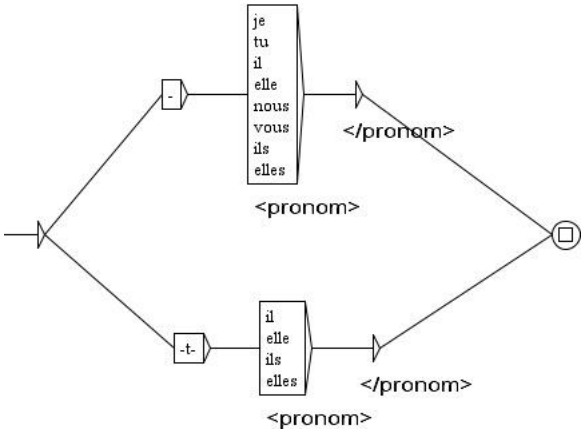


Par exemple, dans la concordance, nous trouvons la séquence *je n'aurais pas été averti de mon erreur* (4 *tokens* reconnus) qui est annotée *je <négation>n'</négation><verbe forme="négative">aurais <négation>pas</négation> été averti</verbe> de mon erreur* en passant par un chemin de poids final 4 alors que le chemin *je <négation>n'</négation><verbe forme="négative">aurais <adverbe>pas</adverbe> été averti</verbe> de mon erreur* correspond à un poids final 2.

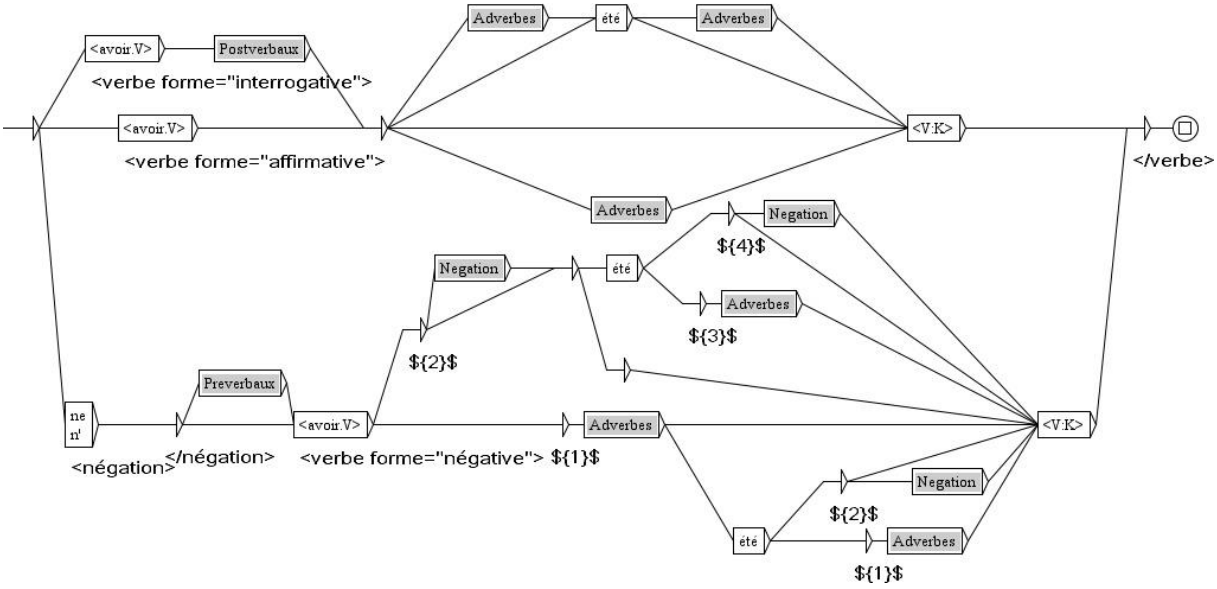
1.7 Formes interrogatives

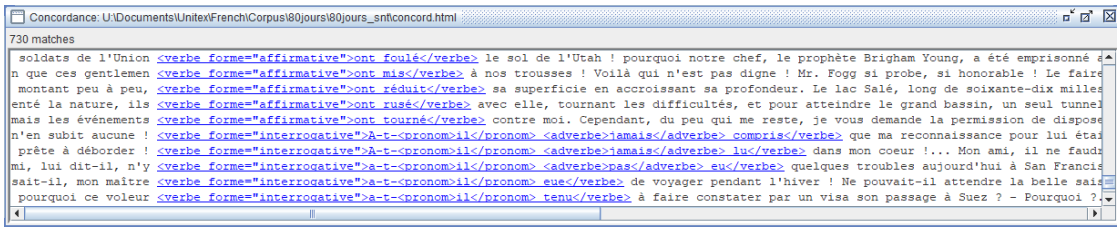
Pour la forme interrogative, il nous faut juste ajouter la présence d'un pronom postverbal. Ces pronoms seront décrits dans un sous-graphe.

1.7.1 Graphe Postverbaux.grf



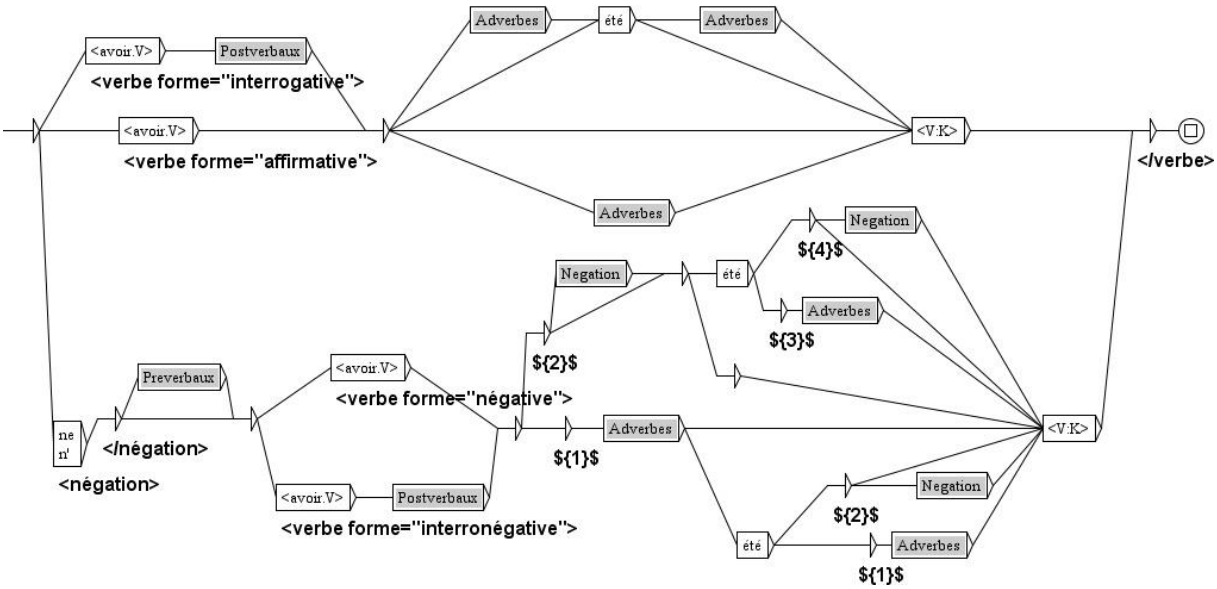
1.7.2 Graphe principal

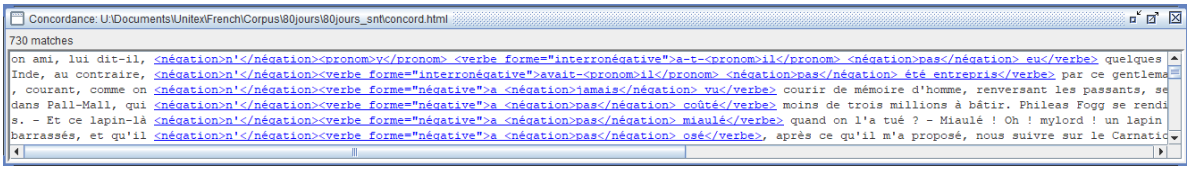




1.8 Formes interronégatives

Pour les formes interronégatives, il suffit de copier-coller à nouveau ce même chemin (en modifiant l'attribut= :

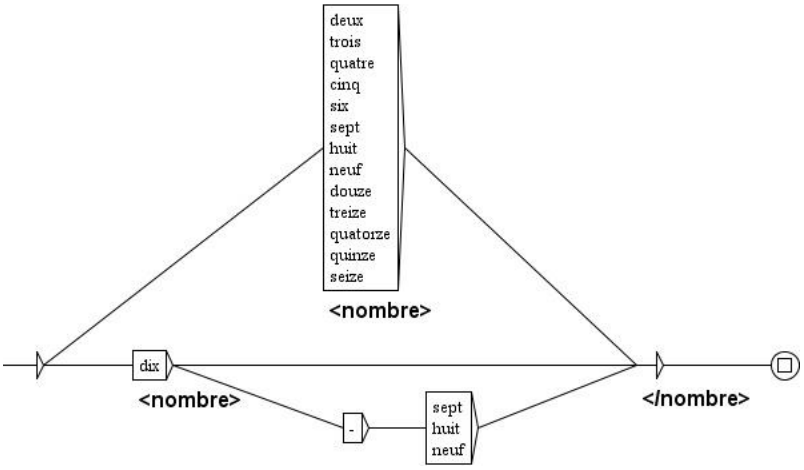

<avoir.V>
Postverbaux
<verbe forme="interrogative">
<avoir.V>
<verbe forme="affirmative">
Adverbes
été
Adverbes
<V:K>
</verbe>
Adverbes
Negation
${4}$
Negation
été
Adverbes
${3}$
${2}$
Preverbaux
<avoir.V>
<verbe forme="négative">
ne
n'
</négation>
<négation>
Adverbes
${1}$
<V:K>
<avoir.V>
Postverbaux
<verbe forme="interronégative">
Negation
${2}$
été
Adverbes
${1}$

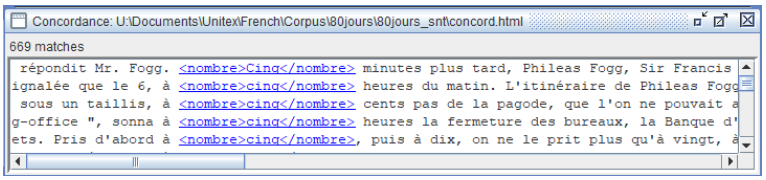

Concordance: U:\Documents\Unitex\French\Corpus\80jours\80jours_snt\concord.html
730 matches
on ami, lui dit-il, <négation>n'</négation><pronom>y</pronom> <verbe forme="interronégative">a-t-<pronom>il</pronom> <négation>pas</négation> eu</verbe> quelques
Inde, au contraire, <négation>n'</négation><verbe forme="interronégative">avait-<pronom>il</pronom> <négation>pas</négation> été entrepris</verbe> par ce gentlema
, courant, comme on <négation>n'</négation><verbe forme="négative">a <négation>jamais</négation> vu</verbe> courir de mémoire d'homme, renversant les passants, se
dans Pall-Mall, qui <négation>n'</négation><verbe forme="négative">a <négation>pas</négation> coûté</verbe> moins de trois millions à bâtir. Phileas Fogg se rendi
s. - Et ce lapin-là <négation>n'</négation><verbe forme="négative">a <négation>pas</négation> miaulé</verbe> quand on l'a tué ? - Miaulé ! Oh ! mylord ! un lapin
barrassés, et qu'il <négation>n'</négation><verbe forme="négative">a <négation>pas</négation> osé</verbe>, après ce qu'il m'a proposé, nous suivre sur le Carnatic

2 Sujet : annoter *les nombres écrits en toutes lettres*

Notre deuxième exercice va consister à annoter les nombres écrits en toutes lettres dans un texte. Nous allons procéder par étape successive en utilisant des sous graphes à chacune d'entre elles. Commençons par les nombres de *deux* à *dix-neuf*. Le nombre *un* ne sera pas annoté à cause de son ambiguïté avec le déterminant. Le nombre onze non plus, car il nécessite un traitement particulier dans les nombres *soixante-et-onze* et *quatre-vingt-onze*.

2.1 De 2 à 19 (11 excepté) : graphe *NB2-10_12-19.grf*

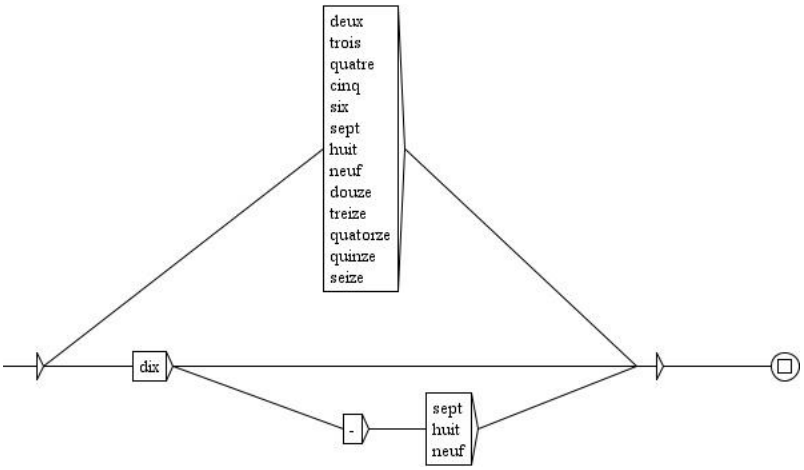




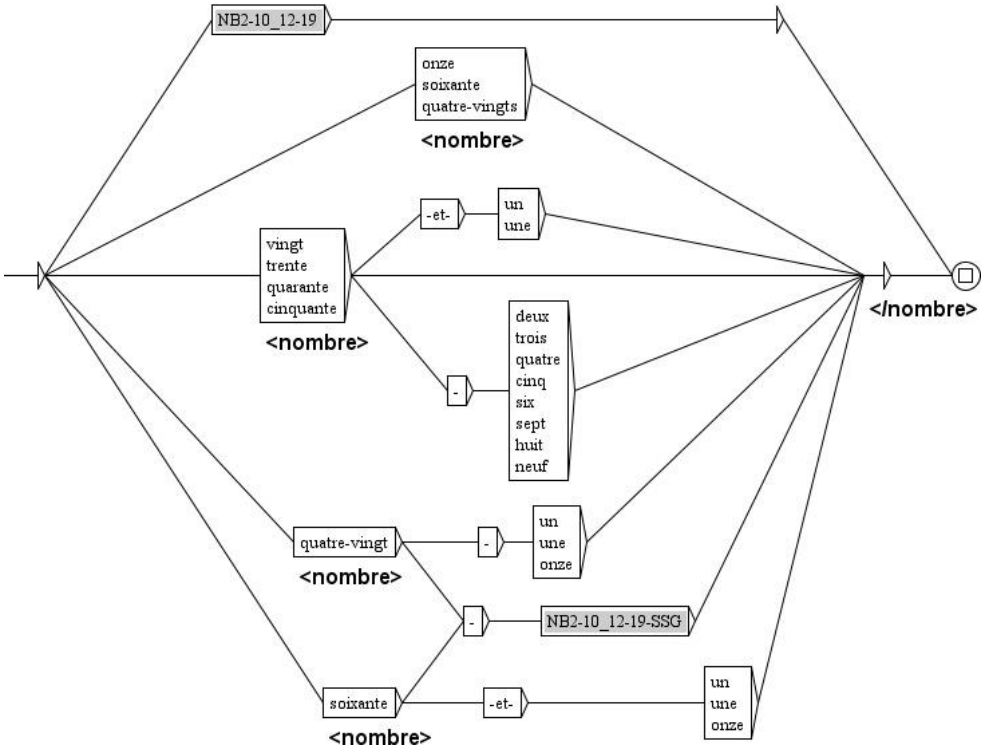
2.2 De 2 à 99 : graphe *NB2-99.grf*

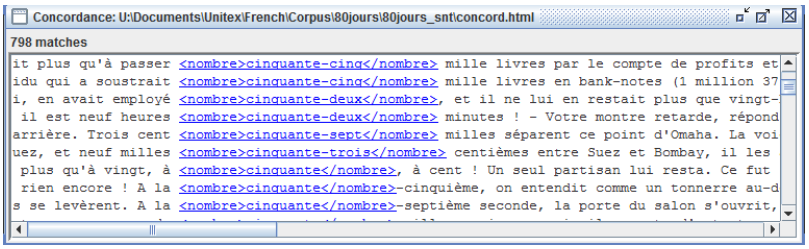
Pour les nombres supérieurs à *soixante*, nous allons utiliser le graphe précédent, ou, plutôt, une version légèrement différente, où nous avons enlevé les annotations. En effet, nous voulons annoter *cinquante-cinq* sous la forme *<nombre>cinquante-cinq</nombre>* et non pas sous la forme *<nombre>cinquante<nombre>cinq</nombre></nombre>*.

2.2.1 Graphe *NB2-10_12-19SSG.grf*



2.2.2 Graphe *NB2-99.grf*



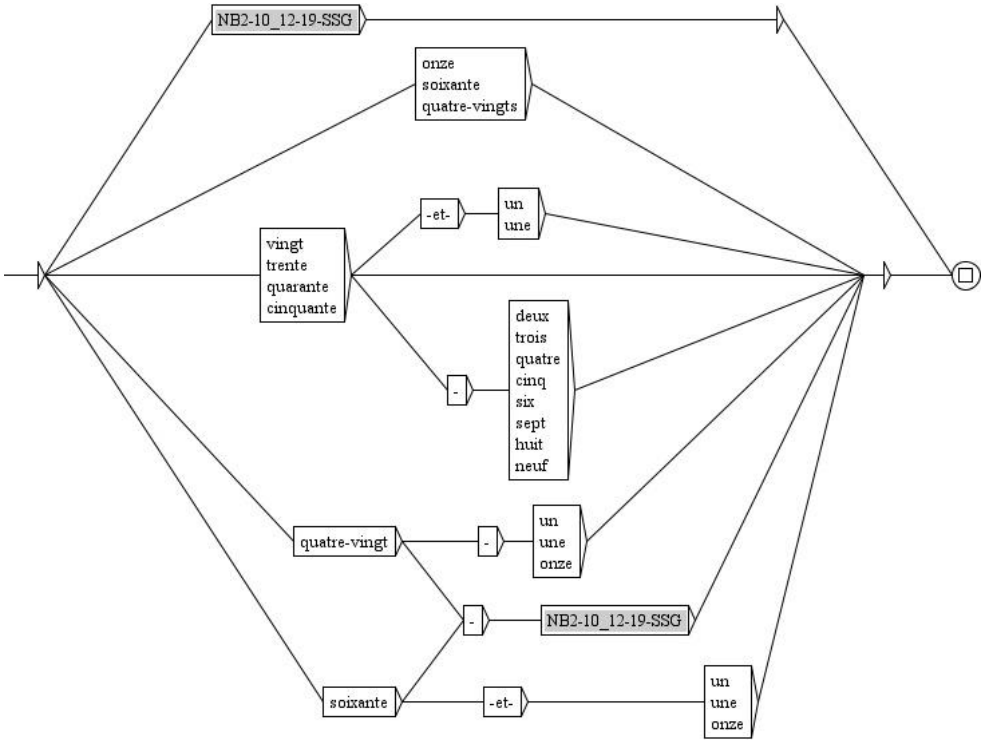


2.3 De 2 à 999

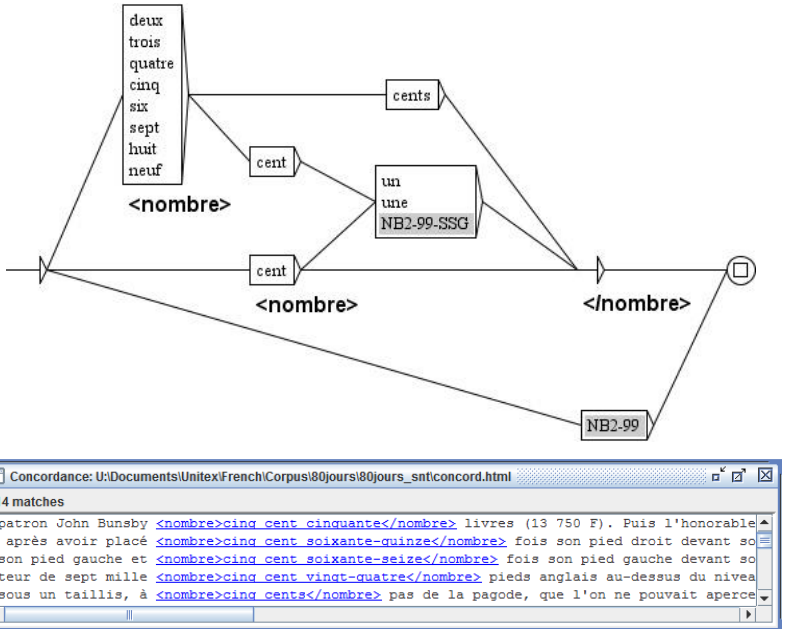
Il en est de même pour les nombres supérieurs à *cent*.

2.3.1 Graphe NB2-99-SSG.grf

Attention à penser à modifier l'appel au sous-graphe *NB2-10_12-19.grf* en un appel au sous-graphe *NB2-10_12-19SSG.grf*.



2.3.2 Graphe NB2-999.grf

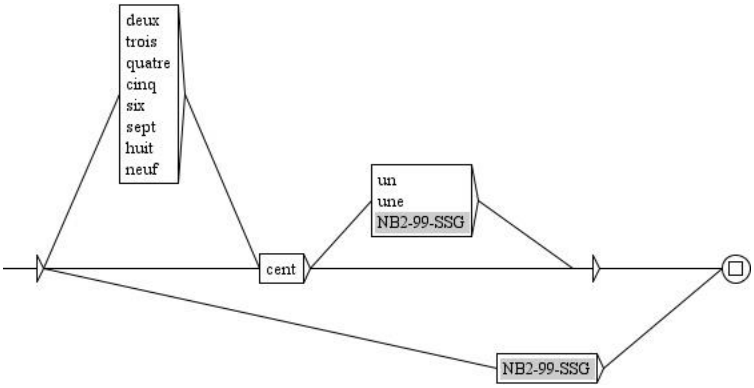


2.4 De 2 à 999 999

Et pour les nombres supérieurs à *mille*.

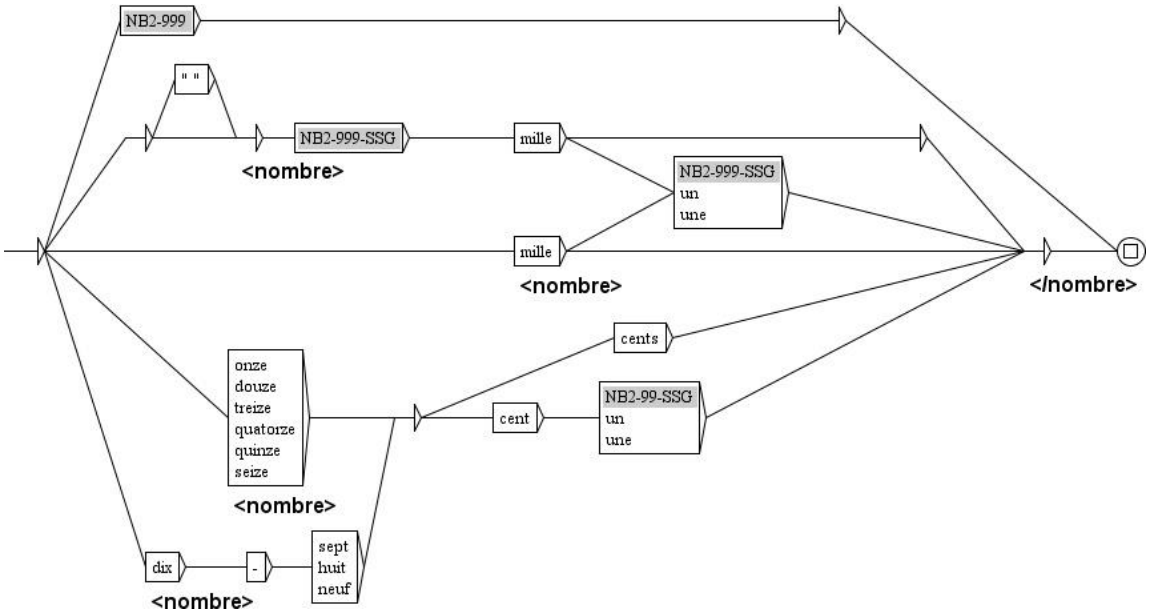
2.4.1 Graphe NB2-999-SSG.grf

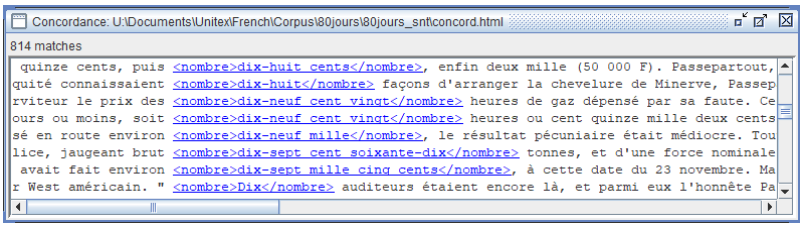
Ce graphe est légèrement différent du graphe NB2-999.grf, car la particularité du pluriel du nombre cent disparaît.



2.4.2 Graphe NB2-999999.grf

Une petite difficulté surgit : nous ne pouvons pas mettre de sortie sur un sous graphe. Or, pour écrire *dix mille*, nous allons utiliser le sous graphe *NB2-999-SSG.grf* avant la boite contenant *mille*. Nous plaçons la balise *<nombre>* dans une boite vide avant la boite contenant le sous-graphe.





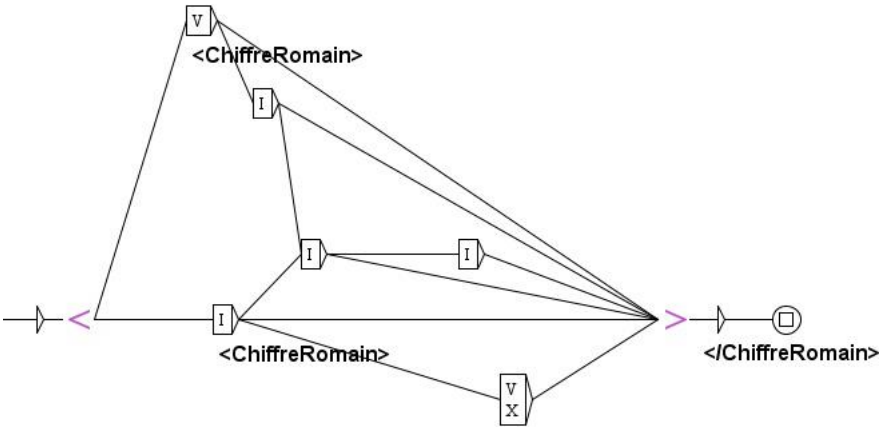
3 Sujet : annoter les chiffres romains

Les graphes morphologiques : voir le manuel, section 6.4.1.

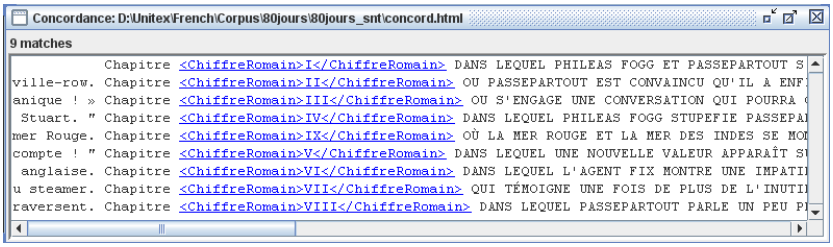
Passons maintenant aux chiffres romains. On aurait envie de les décrire par des graphes… mais trois boites successives contenant la lettre *I* nous permettent de reconnaitre les trois mots *I I I*, mais pas le mot *III* ! Pour cela nous allons utiliser le mode morphologique qui permet de décrire l'intérieur d'une séquence de lettres.

3.1 De 1 à 9 : graphe *CR1-9.grf*

Une fois le graphe réalisé, nous pouvons le sélectionner la partie morphologique et cliquer sur le bouton *surround box selection with morphological mode tags*.



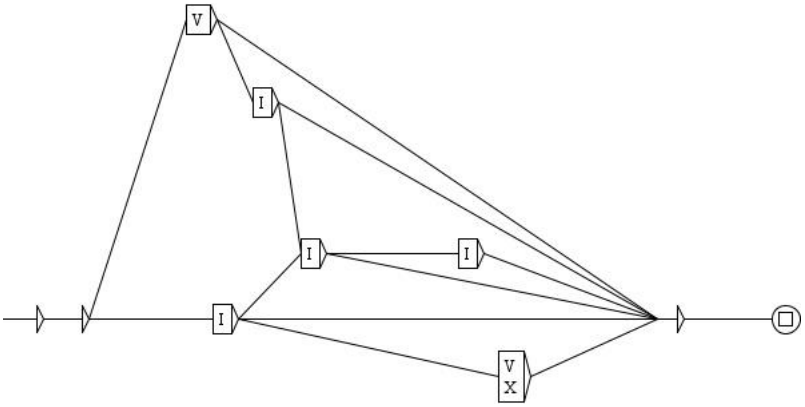
Voir la vidéo : https://tln.lifat.univ-tours.fr/medias/video/cr1-9_1580216341302-mp4?ID_FICHE=334589&INLINE=FALSE.



3.2 De 1 à 99

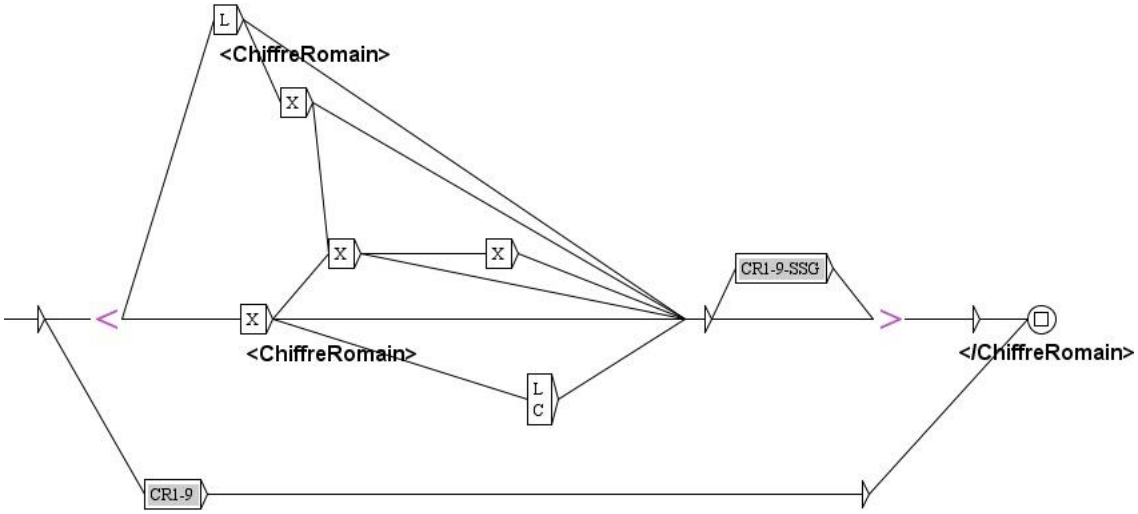
Pour monter à 99, il nous faut comme à l'exercice 2 créer un sous-graphe, de 1 à 9, uniquement descriptif, sans les annotations et sans les boites morphologiques.

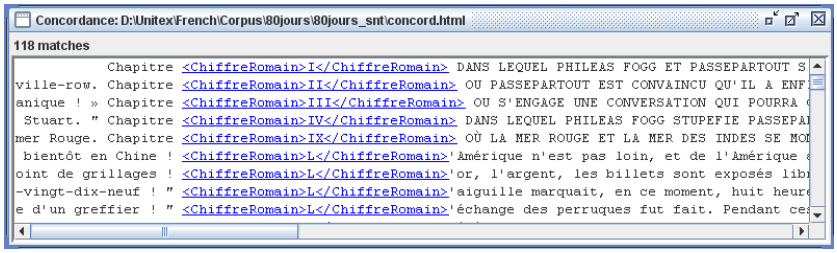
3.2.1 Graphe *CR1-9-SSG.grf*



3.2.2 Graphe exCR1-99.grf

Un problème survient lors de la création du graphe suivant. L'idée, bien sûr, est d'utiliser le premier graphe comme schéma, en remplaçant *I* par *X*, *V* par *L* et *X* par *C*. Mais le *L* seul est confondu avec l'article et pose problème.

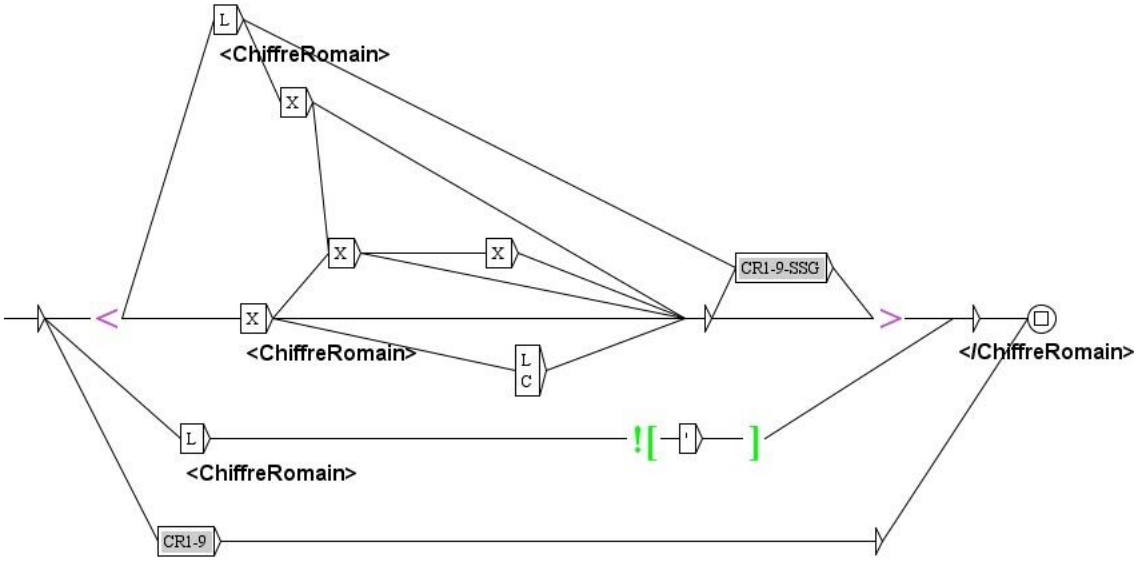




3.2.3 Graphe CR1-99.grf

Les contextes droits : voir le manuel, section 6.3.1.

Pour résoudre le problème, nous allons indiquer que *L* ne sera étiqueté *<ChiffreRomain>* que s'il n'est pas suivi d'une apostrophe. Pour cela, on sélectionne la boite concernée et on clique sur le bouton *surround box selection with negative right context tags*.

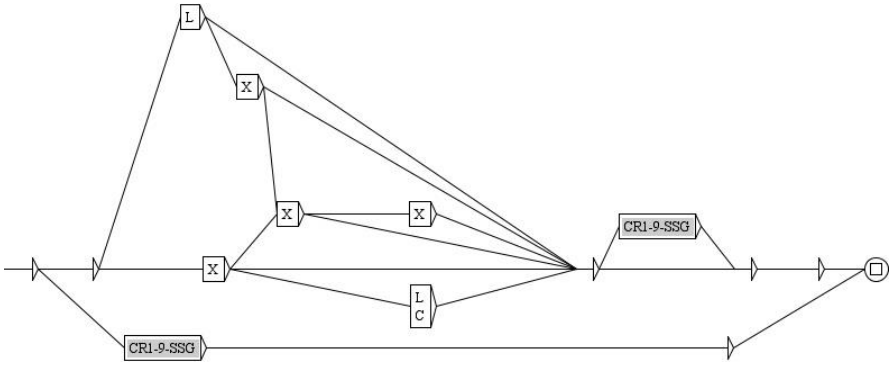


Voir la vidéo : https://tln.lifat.univ-tours.fr/medias/video/cr1-99_1580216416271-mp4?ID_FICHE=334589&INLINE=FALSE.

3.3 De 1 à 999

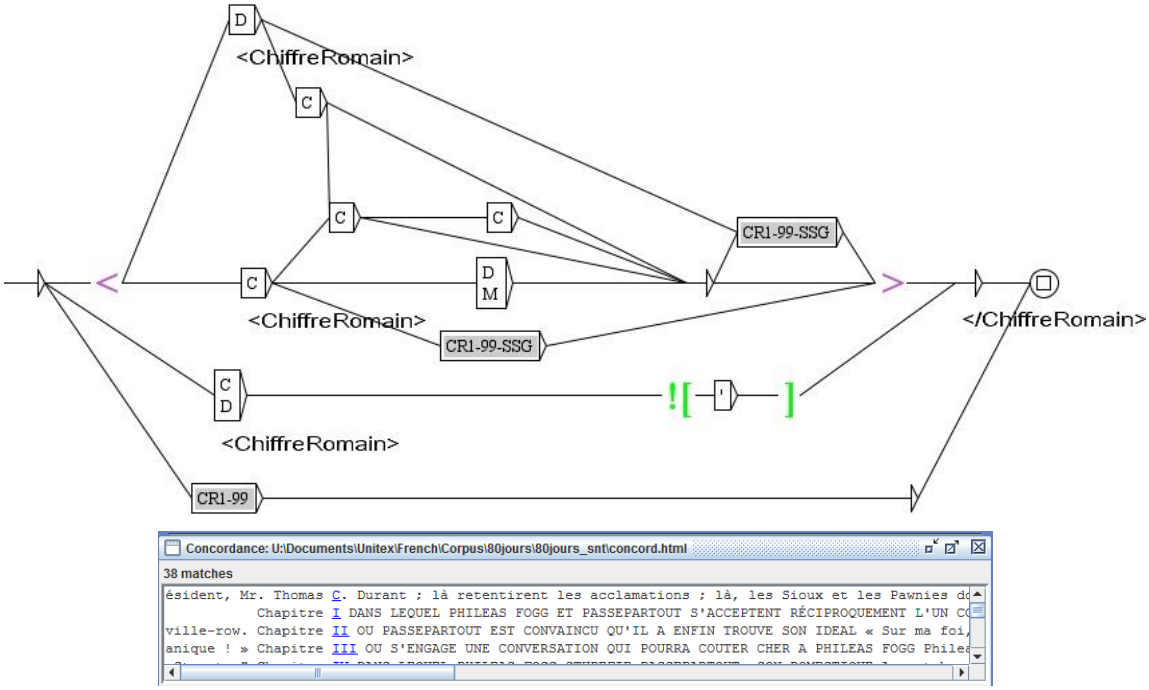
Pour passer à 999, il faut comme d'habitude créer un sous graphe de 1 à 99 sans les annotations et sans les boites morphologiques.

3.3.1 Graphe CR1-99-SSG.grf



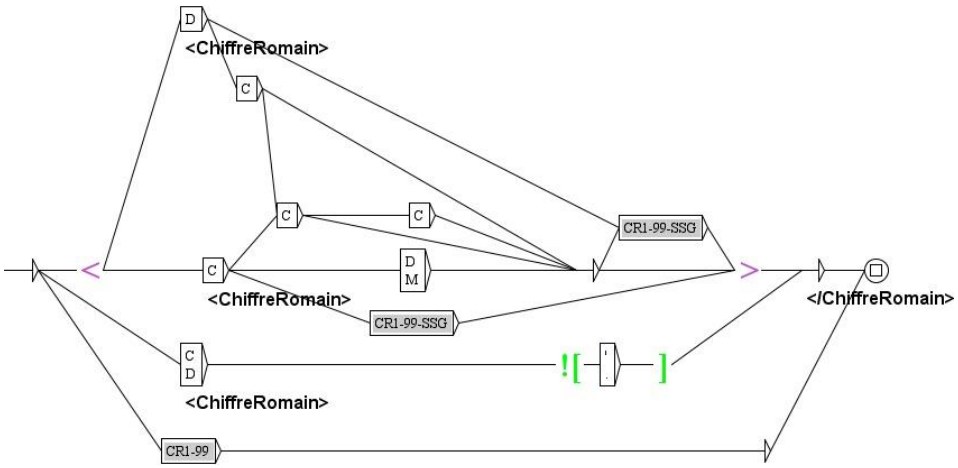
3.3.2 Graphe exCR1-999.grf

Ensuite, utilisons le graphe précédent, en remplaçant *X* par *C*, *L* par *D* et *C* par *M*. Comme le *C'* va aussi poser problème, nous le corrigeons de suite.



Comme vu sur la concordance ci-dessus, nous avons fait une erreur, car le *C* suivi d'un point est ici l'abréviation d'un prénom ! Ce qui pourrait arriver aussi au *D*. Nous ajoutons donc le point dans le contexte droit négatif.

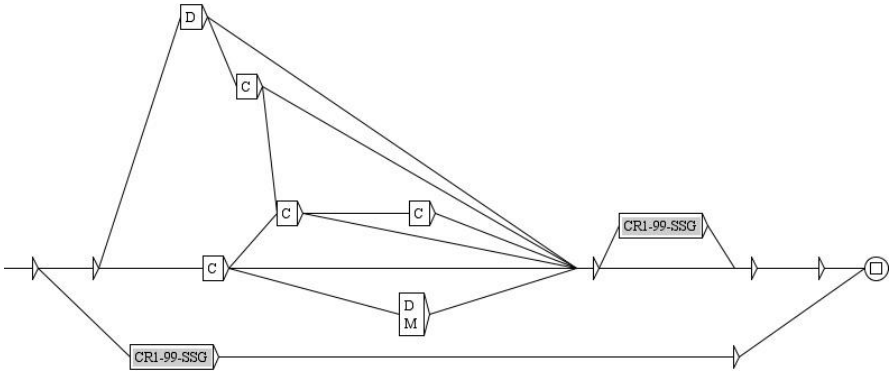
3.3.3 Graphe CR1-999.grf



3.4 De 1 à 3999

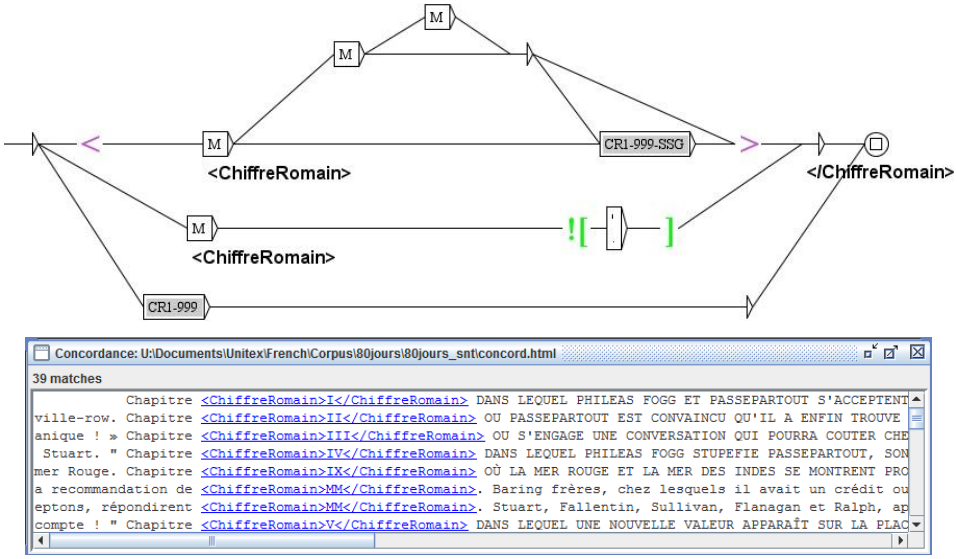
Commençons par le sous-graphe habituel.

3.4.1 Graphe CR1-999-SSG.grf



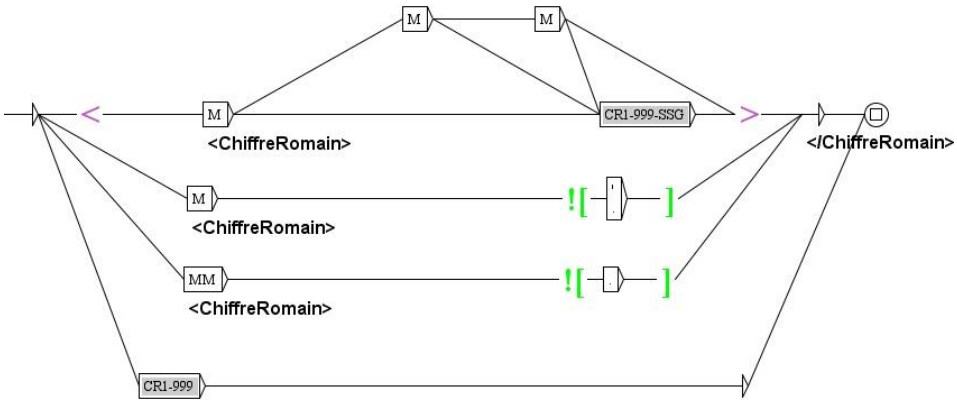
3.4.2 Graphe exCR1-3999.grf

Ensuite, utilisons le graphe précédent, en remplaçant *C* par *M*[3].



3.4.3 Graphe CR1-3999.grf

Cependant nous constatons une nouvelle erreur : la reconnaissance de *MM.*, abréviation de *Messieurs*. Ajoutons cette condition négative sur le graphe.



[3] Les chiffres romains s'arrêtent à 3 999. Il n'y a pas de symboles pour 5 000, ni pour 10 000…

3.5 Reprise de l'ensemble de l'exercice

Arrivé à la fin de cet exercice, nous pourrions être satisfaits… mais nous avons commis une effroyable erreur méthodologique en nous basant uniquement sur notre corpus…

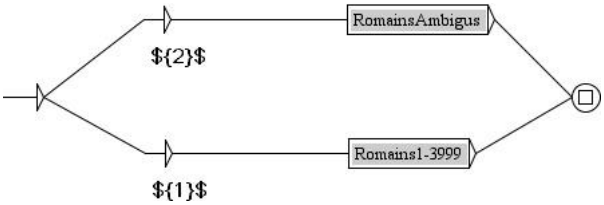
En effet, chaque lettre correspondant à un chiffre romain pourrait être utilisée comme abréviation d'un prénom. Nous venons d'éliminer *C.* et *M.*, mais il faudrait éliminer aussi *I.*, *V.*, *X.*, *L.* et *D.*, non présents dans notre corpus !

D'autre part, nous avons rencontré l'abréviation *MM.*, mais omis aussi les acronymes *CD*, *DC*, *CM*… et les mots *CI*, *DIX*, *MI*…

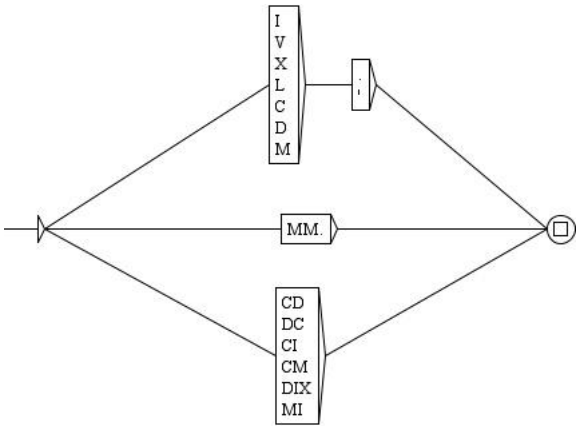
Et que dire de certains noms propres, comme *Tristan M'Bongo*, qui commencent par une lettre majuscule suivie d'une apostrophe ?

En bref, il faut tout reprendre. Nous allons créer un sous-graphe des ambiguïtés à éviter et un sous-graphe des chiffres romains sans aucune exception. Le choix se fera en utilisant les poids.

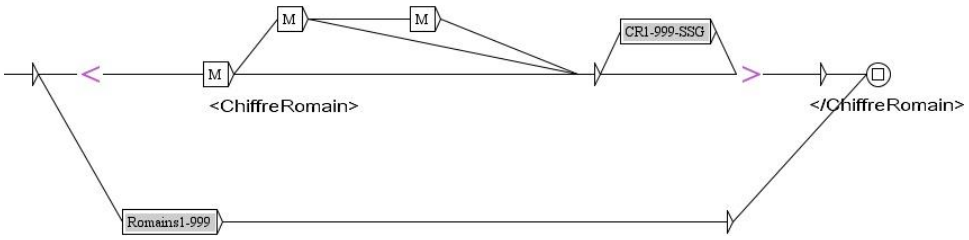
3.5.1 Graphe Romains.grf



3.5.2 Sous-graphe RomainsAmbigus.grf



3.5.3 Sous-graphe Romains1-3999.grf



3.5.4 Sous-graphe Romains1-999.grf

Du coup, dans ce graphe, nous utilisons le sous-graphe *exCR1-99.grf*.