

# Corpus, ressources et linguistique outillée · M2SOL034

## CM 4 : Automates et transducteurs à états finis · Unitex

---

Ljudmila PETKOVIĆ

Semestre 2, 2024-2025

21 février 2025

Sorbonne Université

Master « Langue et Informatique » (M1 ScLan)

UFR Sociologie et Informatique pour les Sciences Humaines

Cours adapté de FORT (*s.d.*), de NOUVEL (*s.d.*) et de TANGUY (*s.d.*).

# Automates et transducteurs

---

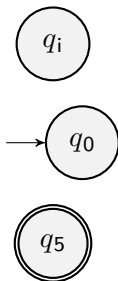
# Automates finis

angl. *Finite-State Automata*

- machine permettant de définir un langage
- capable d'indiquer si une chaîne fait partie ou non du langage
- chaîne entrée  $\rightarrow$  [automate]  $\rightarrow$  oui/non

# États

Indiquent où en est l'analyse d'un mot.

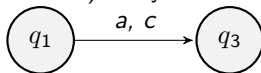


- États : **nœuds**
  - **Cercle**
  - **Étiquette** :  $q_i$  avec un  $i$  entier
- État **initial**
  - Ajout d'une **flèche** devant
  - Souvent  $q_0$  (mais pas obligatoire)
- État **final**
  - **Double** cercle

# Transitions

Indique quelles prochains symboles sont acceptés.

- Transitions : **arcs**
  - **Arc orienté** (flèche) qui relie deux états
  - **Étiquette** : liste (ensemble) de symboles de l'alphabet  $\Sigma$



- Reconnaît le langage  $\{a, c\}$  ou  $\{a\} \cup \{c\}$  (mais pas  $\{a.c\}$ )
    - Si, en  $q_1$ , le prochain symbole est  $a$  ou  $c$ , aller en  $q_3$
- Transition d'un état vers lui-même
  - **Boucle** au-dessus d'un état
  - Correspond à l'étoile de Kleene

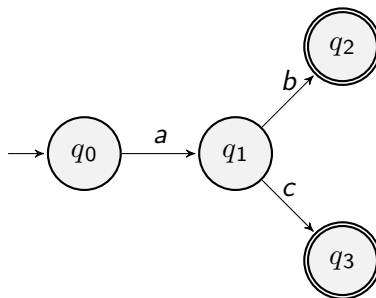


## Reconnaissance d'un mot

Chemin suivi au travers d'un automate.

- L'automate consomme les symboles
- Une liste d'états « visités » est établie
- Arrivée en fin de mot dans l'état final

Exemple : mots *ab* ou *ac*



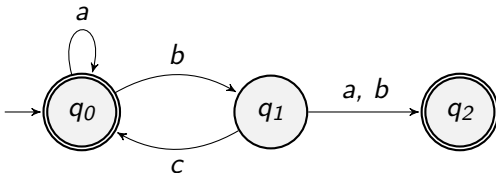
# Automates à états finis déterministes (AFD)

angl. *Deterministic Finite State Automata*

- un seul état initial
- déterministe : par nœud / symbole, une transition maximum
- autant d'états finaux que nécessaire
- l'état initial peut-être final
- des transitions peuvent partir d'un état final
- boucles possibles sur un état ou par cycles

## Exemple d'un AFD

Regex  $(a|bc)^*(b(a|b))$



Mots acceptés	Mots non acceptés
$ba$ (voir $(a bc)^*(b(a b))$ )	$ab$ (il manque le 2 <sup>e</sup> caractère après $b$ )
$bba$ (répétition de $(a bc)$ )	$bc$ (il ne finit pas par $b(a b)$ )
$bcba$	$abca$
$bcbb$	
$aaba$	



## Grammaires locales

- type particulier d'automates finis
- utilisées pour reconnaître des motifs sans transformation
- décrivent des motifs linguistiques à l'aide de graphes
- chemin conduisant de l'état initial à l'état final : motif accepté

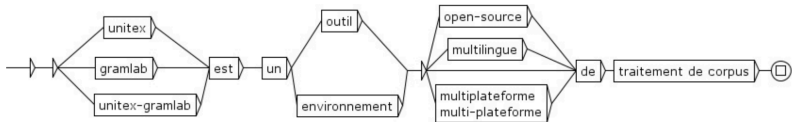


FIG. 1 – *Grammaire locale représentée sous forme de graphe*

- Unitex-GramLab est un outil multilingue de traitement corpus [RECONNUE]
- Unitex-GramLab est un outil de traitement de corpus [RECONNUE]
- Unitex-GramLab est difficile à apprendre [ECHEC]
- Unitex-GramLab est [ECHEC]

Figure 1 – (KYRIACOPOULOU et al., 2018).

# Transducteurs à états finis

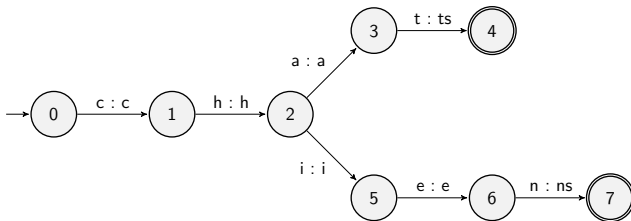
angl. *Finite State Transducer*

Machine abstraite fonctionnant sur le même principe que les AFD

- possibilité d'émettre un message à chaque transition
- capable de **reconnaissance** d'un langage formel, et de **production** d'une chaîne en sortie
- chaîne entrée  $\rightarrow$  [transducteur]  $\rightarrow$  chaîne sortie

## Fonctionnement d'un transducteur : exemple de flexion

- lecture de la chaîne d'entrée comme un automate à états finis
- à chaque transition, si un message est associé, il est émis
- aucun message ne sera émis si la chaîne n'est pas reconnue
- entrée : chaîne correspondant à un nom au singulier
- sortie : pluriel du nom fourni en entrée
  - chat → chats (exemple trivial)
  - cheval → chev**aux** (exemple moins trivial)



# Utilisations de transducteurs en TAL

## Phonétisation (*text to speech*)

- entrée : chaîne de caractères orthographique
- sortie : chaîne de symbole phonétiques

## Segmentation

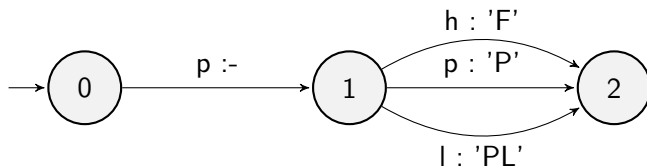
- entrée : chaîne de caractères (texte ou phrase)
- sortie : séquence de phrases, de mots ou de morphèmes

## Analyse d'unités lexicales

- entrée : mot
- sortie : informations diverses sur le mot
  - informations morphosyntaxiques, équivalents multilingues etc.

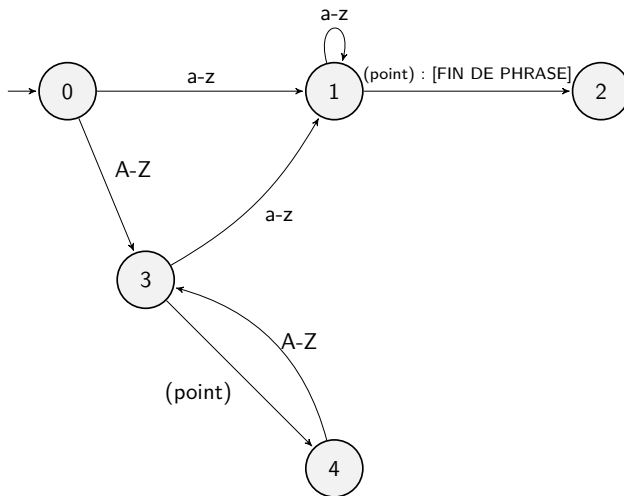
# Phonétisation

Comment se prononce la lettre « p » ?



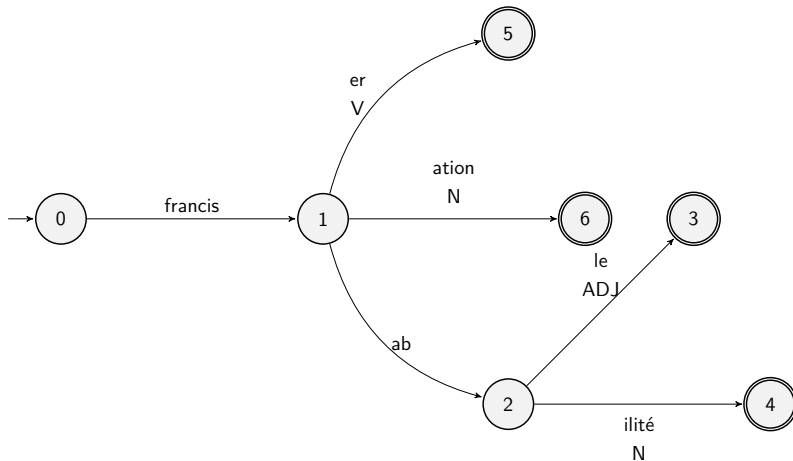
## Segmentation d'un texte

Le point est-il une fin de phrase ou un élément d'un sigle ?



# Analyse morphosyntaxique

Catégories des mots de la famille « francis. . . »



**Unitex**

---



## À propos d'Unitex

- suite logicielle<sup>1</sup> pour l'analyse des corpus
- fondée sur des ressources linguistiques :
  - dictionnaires électroniques
  - grammaires locales
  - tables lexico-syntaxiques (lexique-grammaire)
- multiplateforme, multilingue
- documentation (PAUMIER, 2021)<sup>2</sup>
- tutoriels
  - en français (univ. de Tours)<sup>3</sup>
  - en anglais (KRSTEV et al., 2022)

---

1. <https://unitexgramlab.org/fr>

2. <https://unitexgramlab.org/releases/3.1/man/Unitex-GramLab-3.1-usermanual-fr.pdf>

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

# Applications

- recherche de motifs complexes dans des textes
- concordance (visualisation des résultats en contexte)
- annotation
- analyse

→ par la création de grammaires locales ou de transducteurs

→ via une interface graphique

# Prétraitements du corpus

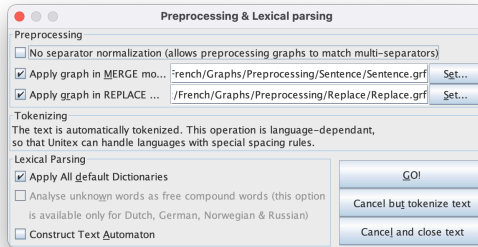


Figure 2 – Prétraitement du corpus *Tour du monde en 80 jours* de Jules Verne.

# Prétraitements du corpus

- Apply graph in MERGE mode
  - découpage du texte en phrases (délimiteur S)
- Apply graph in REPLACE mode
  - normalisations de formes non ambiguës (*puisque*' → *puisque*)
- découpage en unités lexicales : tokenisation
- Apply All default Dictionaries
  - appliquer au texte des dictionnaires au format DELA<sup>4</sup>
- construction de l'automate du texte

---

4. Dictionnaires Électroniques du LADL

# Corpus prétraité

80jours.snt (/Users/ljudmilapetkovic/workspace/Unitex-GramLab/Unitex/French/Corpus)

3652 sentence delimiters, 165239 (9452 diff) tokens, 71859 (9422) simple forms, 438 (10) digits  
67464 occurrences (12178 DLF entries) simple words, 1617 occurrences (1544 DLC entries) compound words, 4233 occurrences (...)

Chapitre I  
(S)DANS LEQUEL PHILEAS FOGG ET PASSEPARTOUT S'ACCEPTENT RÉCIPROQUEMENT L'UN COMME MAÎTRE, L'AUTRE COMME DOMESTIQUE  
(S)En l'année 1872, la maison portant le numéro 7 de Saville-row, Burlington Gardens \_ maison dans laquelle Sheridan mourut en 1814 \_ , était habitée par Phileas Fogg, esq., l'un des membres les plus singuliers et les plus remarqués du Reform-Club de Londres, bien qu'il semblât prendre à tâche de ne rien faire qui pût attirer l'attention.  
(S)A l'un des plus grands orateurs qui honorent l'Angleterre, succédait donc ce Phileas Fogg, personnage énigmatique, dont on ne savait rien, sinon que c'était un fort galant homme et l'un des plus beaux gentlemen de la haute société anglaise.  
(S)On disait qu'il ressemblait à Byron \_ par la tête, car il était irrécusable quant aux pieds \_ , mais un Byron à moustaches et à favoris, un Byron impassible, qui aurait vécu mille ans sans vieillir.  
(S)Anglais, à coup sûr, Phileas Fogg n'était peut-être pas Londonner.(S) On ne l'avait jamais vu ni à la Bourse, ni à la Banque, ni dans aucun des comptoirs de la Cité.(S) Ni les bassins ni les docks de Londres n'avaient jamais reçu un navire ayant pour armateur Phileas Fogg.(S) Ce gentleman ne figurait dans aucun comité d'administration.(S) Son nom n'avait jamais retenti dans un colléne d'avocats. ni au Temple. ni à Lincoln's-inn. ni à Gray's-inn.(S) Jamais il ne

Word Lists in /Users/ljudmilapetkovic/workspace/Unitex-GramLab... Token list

DLF: 12178 simple-word lexical entries

a..N+zl:ms:mp  
à..PREP+zl  
a..avoir.V+zl:P3s  
abaissait,abaissier.V+zl:I3  
abaissant,.A+2:ms  
abaissant,abaissier.V+zl:G  
abaissé,.A+zl:ms  
abaissé,abaissier.V+zl:Kms  
abaissement,.N+2:ms  
abandonna,abandonner.V+zl  
abandonnait,abandonner.V+zl

DLC: 1544 compound lexical entries

à base de,.PREP+zl  
à bon droit,.ADV+zl  
à bord d,à bord de,.PREP+zl  
à bord de,.PREP+zl  
à bord des,à bord de,.PREP+zl  
à bord du,à bord de,.PREP+zl  
à l'issue de,.PREP+zl

ERR: 477 unknown simple words

Filter unknown words with tags.ind

By Frequency By Char Order

69924  
Abraham 6940  
Aden 4566  
Afrique 3797  
Agra 3652 {S}  
Ahmémnagara 2807 de  
Alabama 1695 à  
Albermale 1608 le  
Allahabad 1488 la  
Allemagne 1466 et  
American 1161 -  
and 1137 l  
Andaman 1106 il  
Andrew 948 les  
Angelica 878 -  
Angleterre 824 un  
Annam 784 en  
Aouda 753 du  
Arkansas 726 d  
Armonica 717 "

Figure 3 – Corpus, liste de mots et de tokens.

## Corpus prétraité : statistiques

- 3 652 délimiteurs de phrases
- 165 239 tokens
- 9 452 types
- 9 422 formes simples (DLS<sup>5</sup>, lemmes)
- 10 chiffres (types)
- 12 178 mots simples (DLF<sup>6</sup>, entrées)
- 1 544 mots composés (DLC<sup>7</sup>, entrées)
- 477 mots inconnus (ERR, entrées)

---

5. DELA de formes Simples

6. DELA de formes Fléchies

7. DELA de formes Composées

## Dictionnaires Unitex

1. dictionnaires de formes simples (DELAS)
2. dictionnaires de formes fléchies (DELAF)

qui comprennent des formes simples ou composées (DELAC)

DELAS : cheval,N4,An1

DELAF :

mercantiles,mercantile.A+z1:mp:fp

grand=mères,grand=mère.N:fp

# Contenu d'un dictionnaire Unitex

Ensemble d'entrées lexicales :

- forme de base (canonique, lemme) : Descartes
- catégorie grammaticale : nom (N)
- informations flexionnelles (genre, nombre) : ms
- forme fléchie : René Descartes
- traits syntactico-sémantiques : Hum+NPropre

## Exemple

Descartes,René Descartes.N+Hum+NPropre:ms



# Construction des dictionnaires

1. dictionnaire de formes canoniques (ou formes de base)
2. modules de flexion automatique (transducteurs)
3. à chaque forme de base, on associe une classe flexionnelle
  - un ensemble de règles

DELAS  $\rightarrow$  Flexion automatique DELAF

# Gestion du multilinguisme

Les traitements sont tous dépendants des langues :

- avantages : précision, adaptation aux spécificités
- inconvénients : lourdeur, maintenance compliquée

Alphabets :

- un fichier qui définit les caractères d'une langue  
(Alphabet.txt)
- un fichier indiquant les préférences pour le tri  
(Alphabet\_sort.txt )

# Alphabets

Ouvrir l'alphabet du français :

- que manque-t-il ? Comment est-ce géré ?

→ p. ex. ligatures françaises : æ, Æ, œ, Œ

Pour certaines langues comme le français, il arrive qu'à une lettre minuscule correspondent plusieurs majuscules.

- é → E ou É

Ee, Eé, Éé, Eè, Èè, Eë, Èë, Êê, Êê

# Mots simples vs. composés

## Mot simple

Une séquence de lettres délimitée par des séparateurs (espaces, ponctuation etc.) : *pomme*

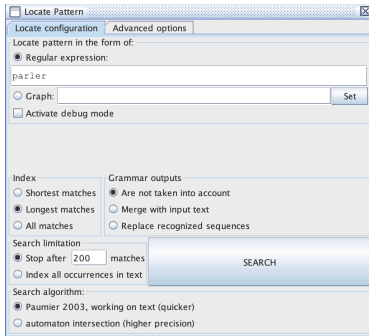
## Mot composé

Une séquence de mots simples dont le sens est non compositionnel : *cordon bleu*, *pomme de terre*, *belle famille*, *porte-manteau*, etc.

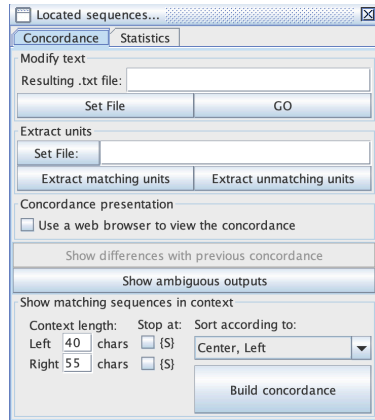
## Recherches simples

1. rechercher le motif parler en cliquant sur Locate Pattern dans le menu Text
  - regarder le résultat avec le concordancier
  - modifier les différentes options et observer les résultats
2. même question avec le motif <parler>
3. même question avec le motif <V:P3p>
4. à quoi correspondent les motifs précédents ?

# Locate Pattern et concordancier



**Figure 4 – Locate Pattern.**



**Figure 5 – Lancer le concordancier.**

## Expression régulières ou rationnelles (regex)

Une regex peut être :

- une **unité lexicale** (livre) ou un masque lexical (<manger.V>)
- une **position** particulière du texte : le début (^) ou la fin \$
- la **concaténation** de deux regex (je mange)
- l'**union** de deux regex (Pierre+Paul)
- l'**étoile de Kleene** d'une regex (très\*)

# Concordancier

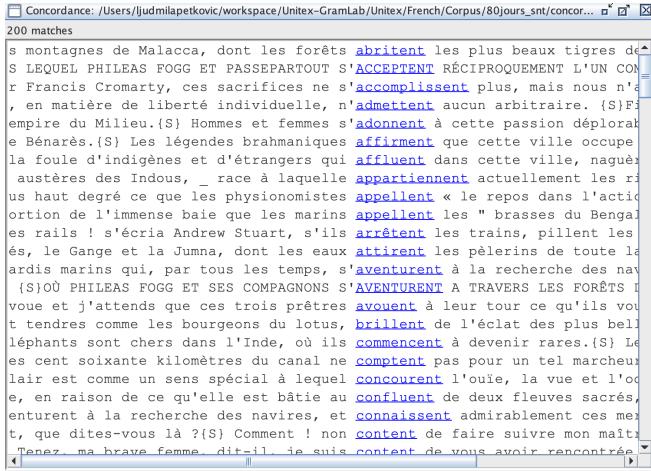


Figure 6 – Concordance sur le motif V:P3p – verbes de la 3<sup>e</sup> personne du pluriel.



# Statistiques de collocations

Located sequences... [X]

Concordance Statistics

Modify text

Resulting .txt file: [ ]

[ Set File ] [ GO ]

Extract units

[ Set File: ] [ ]

[ Extract matching units ] [ Extract unmatching units ]

Concordance presentation

☐ Use a web browser to view the concordance

[ Show differences with previous concordance ]

[ Show ambiguous outputs ]

Show matching sequences in context

Context length: Left 40 chars Right 55 chars

Stop at: ☐ {S} ☐ {S}

Sort according to: Center, Left [v]

[ Build concordance ]

## Collocations du motif V:P3p

Statistics			
Collocate	Occurrences in corpus	Occurrence in match context	z-score
entrés	2	2	17.346
ils	96	11	13.075
exposés	1	1	12.266
physionomistes	1	1	12.266
économés	1	1	12.266
bruns	1	1	12.266
trouvées	1	1	12.266
transocéaniques	1	1	12.266
engagées	1	1	12.266
mesquins	1	1	12.266
venus	1	1	12.266
PROPICES	1	1	12.266
activement	1	1	12.266
steamboats	1	1	12.266
sommets	1	1	12.266
trompés	1	1	12.266
emparés	1	1	12.266
chers	1	1	12.266
réduits	1	1	12.266
orteils	1	1	12.266
surchargés	1	1	12.266
contraste	1	1	12.266
réfugiés	1	1	12.266
demeurés	1	1	12.266

Figure 8 – Collocats, occurrences dans le corpus / contexte de cooccurrence, score z.

# Opérateurs

## ■ Concaténation

- point : `<DET>.<N>` : reconnaît un déterminant suivi par un nom
- espace : `le <A> chat` : reconnaît l'unité lexicale *le*, suivie d'un adjectif et de l'unité lexicale *chat*
- parenthèses : servent de délimiteurs

## ■ Union :

- `+` : `chat+chien <v>` : reconnaît l'unité lexicale *chat* ou *chien*, suivie par un verbe
- epsilon : `le (petit+<E>) chat` : reconnaît les séquences *le chat* et *le petit chat*

## ■ Étoile de Kleene `*` : reconnaît zéro, 1+ occur. d'une regex

- `il fait très* froid` : reconnaît *il fait froid*, *il fait très froid*, *il fait très très froid* etc.
- prioritaire sur les autres opérateurs
- parenthèses pour appliquer l'étoile à une regex complexe

# Codes grammaticaux usuels

Code	Signification	Exemples
A	adjectif	fabuleux, broken-down
ADV	adverbe	réellement, à la longue
CONJC	conjonction de coordination	mais
CONJS	conjonction de subordination	puisque, à moins que
DET	déterminant	ses, trente-six
INTJ	interjection	adieu, mille millions de mille sabords
N	nom	prairie, vie sociale
PREP	préposition	sans, à la lumière de
PRO	pronom	tu, elle-même
V	verbe	continuer, copier-coller

Figure 9 – Exemples de codes grammaticaux usuels.

## Codes sémantiques usuels

Code	Signification	Exemple
z1	langage courant	blague
z2	langage spécialisé	sépulcre
z3	langage très spécialisé	houer
Abst	abstrait	bon goût
An1	animal	cheval de race
An1Coll	animal collectif	troupeau
Conc	concret	abbaye
ConcColl	concret collectif	décombres
Hum	humain	diplomate
HumColl	humain collectif	vieille garde
t	verbe transitif	foudroyer
i	verbe intransitif	fraterniser
en	particule pré-verbale (PPV) obligatoire	en imposer
se	verbe pronominal	se marier
ne	verbe à négation obligatoire	ne pas cesser de

Figure 10 – Exemples de codes sémantiques usuels.

## Codes flexionnels usuels

Code	Signification
m	masculin
f	féminin
n	neutre
s	singulier
p	pluriel
1, 2, 3	1st, 2nd, 3rd personne
P	présent de l'indicatif
I	imparfait de l'indicatif
S	présent du subjonctif
T	imparfait du subjonctif
Y	présent de l'impératif
C	présent du conditionnel
J	passé simple
W	infinitif
G	participe présent
K	participe passé
F	futur

Figure 11 – Exemples de codes flexionnels usuels.

## Méta-motifs Unitex

- `<E>` : mot vide, ou epsilon. Reconnaît la séquence vide
- `<TOKEN>` : n'importe quelle unité lexicale sauf l'espace
- `<MOT>` : n'importe quelle unité lexicale formée de lettres
- `<MIN>` : [...] de lettres minuscules
- `<MAJ>` : [...] de lettres majuscules
- `<PRE>` : [...] de lettres et commençant par une majuscule
- `<DIC>` : n'importe quel mot figurant dans les dictionnaires du texte
- `<SDIC>` : [...] mot simple [...]
- `<CDIC>` : [...] mot composé [...]
- `<NB>` : n'importe quelle suite de chiffres contigus

# Négation et interdiction

- ! (immédiatement après <) : négation d'un motif, possible sur :
  - les métas <MOT>, <MIN>, <MAJ>, <PRE>, <DIC>
  - les masques lexicaux ne comportant que des codes grammaticaux, sémantiques ou flexionnels (<!V + z3 : P3)
- ~ : exclut des codes (<A~z3 reconnaît toutes les entrées qui ont le code A sans le code z3)
- # : interdit la présence de l'espace



# Filtres morphologiques Unitex

## Format

motif <<motif morphologique>>

sous la forme de regex au format POSIX<sup>8</sup>

Par défaut, un filtre morphologique tout seul s'applique au méta <TOKEN>, c'est-à-dire à n'importe quelle unité lexicale sauf l'espace.

---

8. [https://fr.wikipedia.org/wiki/Expression\\_r%C3%A9guli%C3%A8re#Expressions\\_rationnelles\\_.C3.A9tendances POSIX](https://fr.wikipedia.org/wiki/Expression_r%C3%A9guli%C3%A8re#Expressions_rationnelles_.C3.A9tendances POSIX)

## Filtres simples

- `<<ss>>` : contient *ss*
- `<<^a>>` : commence par *a*
- `<<ez$>>` : finit par *ez*
- `<<a.*s>>` : contient *a* suivi par un nombre de caractères quelconque, suivi par *s*
- `<<ss|tt>>` : contient *ss* ou *tt*
- `<<[aeiouy]>>` : contient une voyelle non accentuée
- `<<[aeiouy]3,5>>` : contient une séquence de voyelles non accentuées, de longueur comprise entre 3 et 5
- `<<es?>>` : contient *e* suivi par un *s* facultatif
- `<<ss[^e]?>>` : contient *ss* suivi par un caractère qui n'est pas une voyelle *e*
-

## Filtres plus complexes




- $\langle\langle[\text{ai}] \text{ble}\$ \rangle\rangle$  : finit par *able* ou *ible*
- $\langle\langle^{\wedge}[\text{rst}][\text{aeiouy}]2,\$ \rangle\rangle$  : mot formé de 2 ou plus séquences commençant par un *r*, *s* ou *t* suivi d'une voyelle non accentuée




Lorsqu'un filtre suit immédiatement un motif, il s'applique à ce qui est reconnu par le motif :

- $\langle\text{V:K}\rangle\langle\langle\text{i}\$ \rangle\rangle$  : participe passé finissant par *i*
- $\langle\text{CDIC}\rangle\langle\langle\text{.*}\rangle\rangle$  : mot composé contenant deux espaces
- $\langle\text{A:fs}\rangle\langle\langle^{\wedge}\text{pro}\rangle\rangle$  : adjectif fém. sing. commençant par *pro*

# Références

---

-  FORT, K. (s.d.). **TXM : présentation et commandes de base**. Cours « Corpus, ressources et linguistique outillée », [https://members.loria.fr/KFort/files/fichiers\\_cours/TXM\\_1.pdf](https://members.loria.fr/KFort/files/fichiers_cours/TXM_1.pdf). Consulté le 21 février 2025 (*voir p. 1*).
-  KRSTEV, C., . LAPORTE et D. MAUREL (2022). **Tutoriels Unitex en anglais**. Matériels, <https://unitexgramlab.org/fr/blog/annoncements/tutorials-in-english>. Consulté le 21 février 2025 (*voir p. 17*).
-  KYRIACOPOULOU, T., C. MARTINEAU et C. MARTINEZ (2018). **UNITEX/GRAMLAB : plateforme libre basée sur des lexiques et des grammaires pour le traitement des corpus textuels**. In : *Revue des Nouvelles Technologies de l'Information*. <https://hal.science/hal-01702235/>, p. 467-470 (*voir p. 9*).

-  NOUVEL, D. (s.d.). **Automates à états finis**. Diapositives, [https://damien.nouvels.net/cours/langages/03\\_AutomatesEtatsFinis.pdf](https://damien.nouvels.net/cours/langages/03_AutomatesEtatsFinis.pdf). Consulté le 21 février 2025 (*voir p. 1*).
-  PAUMIER, S. (2021). **Unitex 3.3 – Manuel d'utilisation**. In : <https://unitexgramlab.org/releases/3.3/man/Unitex-GramLab-3.3-usermanual-fr.pdf> (*voir p. 17*).
-  TANGUY, L. (s.d.). **SL03OP1Y : Informatique pour l'analyse des textes**. Diapositives, <http://w3.erss.univ-tlse2.fr/membre/tanguy/Cours/SL03OP1/C4.pdf>. Consulté le 21 février 2025 (*voir p. 1*).

# Licence

Le contenu de cette présentation est sous licence CC-BY-NC-SA 4.0  
Utilisation non commerciale – Partage dans les mêmes conditions.

