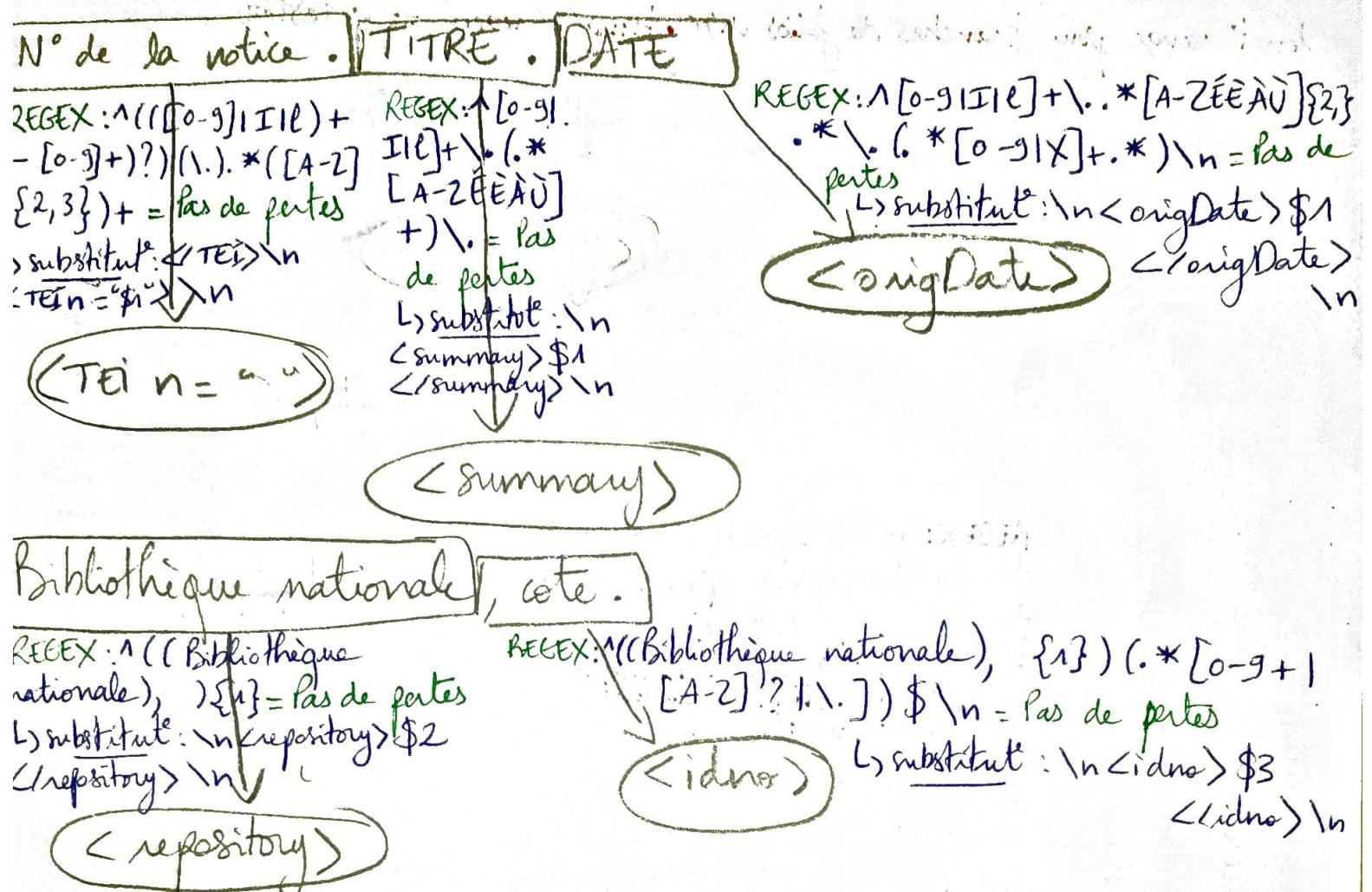


Arbre Notice papier



Si **1^{er} §** ne débute pas par un folio ou la mention d'un feuillet de garde (fol. 1 feuillett)
Condit Python ou XSLT: Si il y a un § apr. la cote et avant la description du feuillet de garde et/ou de l'ancienne cote (à tester)

→ <summary>

§ avec folios, citat et tirets longs :

→ Majuscule ou → tiret long .
 ↗ Regex trop dangereuses
Condit en Python ou XSLT: jouer sur les §, les tirets et les n° de folios (à tester)

↓ sauf si le tiret long est suivie d'une ((y calendrier))

<msItem>

→ si dans un même §, les n° de folios qui suivent sont compris dans la frange de folios qui précède (n₁ < n₂) → créer un <msItem> jusqu'au prochain

tiret long suivi d'un n° de folio > au dernier de la tranche définie, ou bien = avec v°. Condition XSLT ou Python avec fonction range pour tranches de folios avec format (regEx) → à tester

de folios sont parfois encadrés par des ()

↓
A faire apr. avoir encodé les autres éléments du <constItem>
étape ⑥

↳ n° de folio(s) + titre de la partie . -

REGEX: $-?((\text{à } \text{let})|(\text{<} \cdot \star))$
[0-91]
A-21 v° [1,3] \.) ([A-ZÉÉÀÙ]+
[a-zééàù]+ (.*)? [; \| \. \n])
=> pas de parties MAIS cela ne matche
pas les rubriques qui sont
officier de titre

étape ①

<titre>

↳ n° de folio(s) + << italienne >>

Comment jouer sur
italienne
(seule différence
avec italiquit)?

étape ②

<rubric>

↳ n° de folio(s) + (<=) / testé + { : } . . . -

étape ③

<incipit> ?

(n° de folios) (...) testé . . . -

étape ④

<quote> ?

↳ - (n°) . . . + testé + { : } . . . -

Excepté que l'id n'y a pas de différence
entre les 3 types de italique.
Majoritaire entre les 3.

REGEX: (-?) ((0-9 A-21 V°) {1,3} \-) | {3} ?! (\.) | (\.) | (\?) | (\?)
| < (.*)? | (\.) | (\?) | (\?)
| > (.*)? | (\.) | (\?) | (\?)
| = > ligne de testé
| = ligne de rubriques
| = matchées . . .
| = 35 citations perdues

Etape 5

③

< explicit > ?

Etape 1

Ancienne côte
au . ou au

↳ phrase avec 'f/fenillet de garde' et/ou

phrase avec 'autre moins' jusqu'

à) — d'un / / Lent du
calendrier de . — à))
avec jours entre ()

Etape 3

Credit XSLT ou Python (parment du calendrier)
car regex trop dangereuses (peuvent ⚡ qui il
ne faut) : si un titre a la ment "calendrier"
suivi d'un . — ou plus espaces et parfois
un saut de ligne prendre ce qui suit le T- (SAUT
fert) jusqu'au dernier)) suivi de fol/chiffre/Maj.

REGEX (fen feillet de garde & anc côte)

((.+)?) (Fenillet(s)?|feillet(s)? + (de garde))
l'ancien(s)? | Ancien(s)? côte) + (.+?)

(\ - [0-9]a-zA-Z) Mais trouver

moyen d'exclure la phrase qui contient
"Parch/Pareil"

↳ substitut : \n<note>\$1</note>\n

*

§ suivant ceux sur le contenu : ment de l'usage,
souvent de l'office et du calendrier

↓ Etape 1

< origin >

REGEX : ((.+)?) (l'usage | calendrier office) +
. + \n => pas de pertes MAIS
risque de prendre ⚡ qui il ne faut
(élevant de (decompte) notamment)

↳ substitut : \n<p>\$1</p>\n

↳ phrase(s) avec la ment de

date

→ < origDate >

Etape 2

REGEX : \. \s + ([A-ZÉÉÀÀÙÙ]) (.+?) date (.+?) \.)
=> risque de ne pas tout matcher
↳ substitut : \n<origDate>\$1</origDate>\n

Credit Python ou XSLT : matcher phrases contenant
vb "date".

Saut de ligne (double) pris Parch./Pareil.

REGEX : \n\s(Parch\.\. | Pareil\.\.), => pas de
pertes

↳ substitut : \n<material>\$1</material>\n

→ < material >

nbre de folios

REGEX : ([0-9]+ col\.)

=> pas de pertes

↳ substitut : ([0-9]+ col\.)

< layout >

(nbre de col.) jusqu'au .(—)

jusqu' au .(—)

REGEX : \n^ (Parch\.\. | Pareil\.\.), \s

([0-9]+ \s) \. (.+?) (col\.)? (.+?) \.)

=> pas de pertes

↳ substitut : \n<measure>\$2</measure>\n

< measure >

✓ chiffre sur chiffre mill.

(4)

↓
(height)

↓
(width)

REGEX: $([0-9])^+$ sur $([0-9])^+$ mill
=> pas de pertes
↳ substitut: \n < height > \$1 < height >
\n < width > \$2 < width >

— phrase(s) ou § avec mention de l'"écriture"
1° "décorat", 2° "initiales", 3° "miniatures" → entre
les dimensions du manuscrit et la reliure.

+ Condition Python / XSLT (règles trop dangereuses): à chaque notice, matcher ce qui est
entre "mill." - et Rel. / Rcl. / Demi-reliure.
étape ①

(decoNote)

étape ② => si "usage" et/ou "possesseurs"

REGEX: \n (.*)? (usage | possesseur) (.*) \n => pas
de pertes pour ce qui a été
matché pour (decoNote)

(origin)

+ Condition Python ou XSLT pour
extraire la phrase et les suivantes
contenant "possesseur".

(P)

(provenance)

(CPS)

§ commençant par "Rel. / Rcl. / Demi-reliure" jusqu'
au "•(→)"

↓
binding
↳

REGEX: ((Rel). | Rcl). | Demi)-
reliure)(.*?).))((\ -) [A-ZÉ
ÈÀÙ])? => pas de pertes
↳ substitut: \n < cp > \$1 < cp > \n

Si — Non (jusqu'à chiffre ou MAISULT. 5)

étape ① REGEX: (-)?((\D)?[A-ZÉÉÀÙ]{2,3}\.(.|*\?)(-|\n)) => pas de pertes pour biblio. à la fin de la notice MAIS ↗ si dans la notice Maj. suivie de (et de lettres dans la notice, ça marche aussi. Ne pas prendre en compte ce qui a déjà été encodé?

étape ② REGEX: ((\D)?[A-ZÉÉÀÙ]{2,3}\.(.|*\?)(\.) +|- +|\n)) => Sépare les réf à partir des tirets MAIS ↗ ne marche pas la dernière réf.
Corriger les erreurs d'OCR rencontrées à la fin de l'encodage :

- Pareil. → Panch.
- Arwen → Arwen
- Rcl. → Rel.
- ds @n de <tti> → remplacer "I" et/ou "l" par 1

* <note> au sein d'un <msItem> :

étape ④

- fin d'un § après >> -

? Regles trop dangereuse car \n.*>>\s+ - \s+.*) peut matcher la bonne <note> mais aussi d'autres fins de § construites de la même manière (citat ou titre d'un <msItem>)

étape ⑤

- § au milieu de 2 autres sans mention de folios suivis d'un titre ↴ aux sauts de lignes dans un m§ crées par l'OCR

Condition XSLT ou Python : regles (>>\s+\n?|\n.*?\n*\n*\n*\n*)\n fol | [0-9A-Z]) trop dangereuse car matche ④ qu'il ne faut. Bien matcher un § en entier dans la condition.

étape ②

- Pour la ment de "L/lacune" entre - et . - / ou \s+ de ligne

REGEX : ((lacune|lacune) entre(.*)?)\.\{1\}(\s+|-)? => pas de pertes

↳ substitutP = \n<note>\$1</note>\n

étape ⑥

- Pour la ment d'une autre main précédée de - ou de . ou de) et suivie de (ou de :

Regles délicate : (\-|([0-9A-Z]\n|.))|([a-zA-Z]\n))\{1\}+((.*)?) une autre main (.*)?)) (\(1:)) => Pas de pertes mais cela matche parfois trop. Match juste pour ". D'une autre main : "