Projet Tutoré

M1 Technologies de l'Internet

Conception et développement d'une application d'annotation thématique dans l'environnement Gate

---

Auteurs :
Roland Bary
Charles Follet

Tuteurs :
Marie-Noëlle Bessagnet
Annig Lacayrelle
Albert Royer
Christian Sallaberry

Remerciements

Nous tenons à remercier nos tuteurs pour leur pédagogie et leur encadrement. Monsieur Royer pour sa précision et sa connaissance pointue du domaine. Madame Lacayrelle pour son soutien et sa clarté. Monsieur Sallaberry pour nous avoir remis sur de bonnes pistes quand nous nous égarions. Et enfin, madame Bessagnet pour avoir assuré la coordination et le suivi de ce projet.

Table des matières

Première partie

Introduction

L'évolution des volumes d'informations sur internet provoque une évolution du web vers une approche dans laquelle chaque donnée acquiert un sens afin de rendre possible une interprétation du contenu par des machines. Cette évolution constitue le web sémantique.
Sa principale motivation est la recherche d'information sémantique.

Dans ce cadre, nous sommes intervenus pour répondre à l'appel d'offre de la maîtrise d'ouvrage.
L'objectif correspond à la première étape dans un processus de recherche d'informations sémantiques et d'indexation : l'annotation sémantique d'un document texte spécifique.

En découle les problématiques suivantes :

- Existe-t-il des outils qui se prêtent aisément à l'annotation sémantique ?
- Quelle approche de conception nous permet de réaliser cette étape d'annotation sur un document texte non-structuré ?

La résolution de ces différentes problématiques nous a amenés à organiser notre réflexion :
Premièrement, nous définirons clairement les demandes et leurs contextes à travers le cahier des charges. Deuxièmement, nous présenterons l'état des connaissances actuelles sur le sujet. Troisièmement, nous détaillerons notre principe de résolution du projet. Nous finirons le bilan et le retour d'expérience de ce projet.

Deuxième partie

Cahier des charges

I Contexte

A partir des travaux de Georges DEPEYROT sur les monnaies carolingiennes, nous avons travaillé pour une équipe parisienne de numismates sur l'annotation du Numéraire Carolingien [1].
Sur celui-ci, l'équipe a besoin d'effectuer des recherches :

Temporelles : Quelles étaient les pièces en circulation de l'an 859 à l'an 865 ?

Spatiales : Dans quels ateliers, les pièces de type Obole de Charlemagne ont été produites ?

Thématiques : Combien d'exemplaires de la monnaie d'or de Charles le Chauve ont été étudiés ?

Répondre à cette demande implique de définir puis d'explorer les dimensions temporelles, spatiales et thématiques de l'ouvrage.
Pour cela, il est nécessaire de connaître le domaine et l'ouvrage afin de savoir quelle information correspond à quelle dimension.
Une fois cet apprentissage fait, nous pouvons construire des règles dans une chaine de traitement permettant d'annoter chaque information en fonction de sa dimension.

Les monnaies carolingiennes sont le domaine central pour la réalisation du projet. Les ressources nécessaires à l'annotation (ici sous forme de gazetiers) ont été construites à partir des données de l'ouvrage.

Forte de son expérience dans le domaine, la maîtrise d'ouvrage nous a demandé d'utiliser la boîte à outils logicielle GATE qui sera utile pour le traitement du langage naturel.

En résumé, les caractéristiques du projet sont :
- l'apprentissage et la compréhension du domaine considéré,
- l'étude des principes d'annotation de documents,
- le développement d'une chaîne d'annotation dans GATE,
- la mise en place d'une visualisation des résultats.

1. `http://www.cgb.fr/le-numeraire-carolingien-moneta-77-3e-edition-depeyrot-georges,Ln71,a.html`

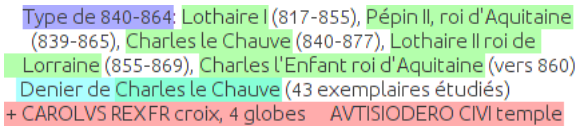
II Description

La chaine de traitement prend un document textuel en entrée, produit un document XML en sortie et le met en forme pour une meilleure lisibilité.
Exemple :

Illustrons par un scénario les objectifs de la chaîne de traitement. Elle prend par exemple en entrée le texte suivant

Type de 840-864: Lothaire I (817-855), Pépin II, roi d'Aquitaine (839-865), Charles le Chauve (840-877), Lothaire II roi de Lorraine (855-869), Charles l'Enfant roi d'Aquitaine (vers 860) Denier de Charles le Chauve (43 exemplaires étudiés)
+ CAROLVS REX FR croix, 4 globes AVTISIODERO CIVI temple

et l'annote



Chaque information pertinente est annotée. En bleu la période d'émission de la monnaie (Temporel), en vert les souverains qui l'ont faite produire (Thématique), en cyan la nature de la monnaie (Thématique) et en rouge la légende (Thématique).
Afin de développer cette chaîne,nous avons dû planifier notre travail et nos réunions avec la maîtrise d'ouvrage. Cette planification sera présentée dans la partie suivante.

III Diagramme de Gantt prévisionnel

Troisième partie

Cadre d'analyse

Introduction

Pour cerner le l'environnement du projet, nous allons d'abord, définir quelques connaissances actuelles liées au domaine de l'annotation sémantique qui nous ont permis d'aborder notre problématique. Ensuite, viendra une description des outils qui ont étés nécessaires lors de la phase de développement.

I Définition de concepts

L'annotation sémantique peut se définir comme une activité qui va mettre une "note" sur une partie d'un texte. Elle permet de travailler plus facilement sur un texte en apportant une sur-couche d'informations, qui va donner un sens aux textes. Comme nous l'avons indiqué en introduction, cette activité fait partie intégrante du Web sémantique. Mais, qu'es ce que le web sémantique ?

1 Le web sémantique

Le Web sémantique, ou toile sémantique, est un mouvement collaboratif mené par le World Wide Web Consortium (W3C) qui favorise des méthodes communes pour échanger des données.
Il vise à aider l'émergence de nouvelles connaissances en s'appuyant sur les connaissances déjà présentes sur Internet. Pour y parvenir, le Web sémantique met en œuvre le Web des données qui consiste à lier et structurer l'information sur Internet pour accéder simplement à la connaissance qu'elle contient déjà. Selon le W3C, « le Web sémantique fournit un Modèle qui permet aux données d'être partagées et réutilisées entre plusieurs applications, entreprises et groupes d'utilisateurs ». L'expression a été inventée par Tim Berners-Lee, l'inventeur du World Wide Web et directeur du World Wide Web Consortium (« W3C »), qui supervise le développement des technologies communes du Web sémantique. Il définit le Web sémantique comme « un web de données qui peuvent être traitées directement et indirectement par des machines pour aider leurs utilisateurs à créer de nouvelles connaissances ».

FIGURE 1 – Logo du W3C pour le Web sémantique

2 Les entités nommées

On peut définir les entités nommées comme étant des objets textuels (c'est-à-dire un mot, ou un groupe de mots) catégorisantes dans des classes telles que noms de personnes, noms d'organisations ou d'entreprises, noms de lieux, quantités, distances, valeurs, dates, etc, auxquelles ont associe un identifiant unique. Les entités nommées sont en effet "associés à des expressions linguistiques sollicités par des applications qui manipulent des documents textuels".
Voici l'illustration d'une recherche d'entités nommées :

```
Henri a acheté 300 actions de la société AMD en 2006
<ENAMEX TYPE="PERSON">Henri</ENAMEX> a acheté
<NUMEX TYPE="QUANTITY">300</NUMEX> actions de la société
<ENAMEX TYPE="ORGANIZATION">AMD</ENAMEX> en
<TIMEX TYPE="DATE">2006</TIMEX>.
```

En revanche, il est important de noter que la reconnaissance et la résolution d'entités nommées, peut s'avérer difficile dans la mesures où elles sont sujettes à des ambiguïtés liées aux phénomènes linguistiques tel que : La synonymie, l'homonymie ou encore la métonymie.
Voici un exemple avec l'entitée nommée "Paris" au sein des énoncés suivants :

- "La star **Paris** Hilton s'est achetée un nouveau chien." (Personne)
- "Je suis parti en vacances à **Paris**." (Lieu)
- "Le **Paris** Saint-Germain a battu Marseille à domicile." (Organisation sportive)

3 Les gazetiers

Un gazetier peut être assimilé à un ensemble de listes contenant des noms d'entités telles que les villes, les organisations, les jours de la semaine, les métiers, etc. Ces listes sont utilisés pour trouver des occurrences de ces noms dans un texte donné, par exemple pour l'activité de recherche d'entités nommées. Le terme Gazetier est souvent utilisé de manière interchangeable pour l'ensemble des listes d'entités et la ressource de traitement qui permet l'utilisation de ces listes pour trouver des occurrences de noms dans le texte. Chaque gazetier est fichier texte d'extension "lst" avec une entrée par ligne. Ci-dessous un exemple de Gazetier : Sur chaque entrée du gazetier, on distingue plusieurs colonnes de par des séparateurs (: ,;, ect). La première colonne correspond à la valeur d'une entité, la seconde constitue le MajorType de l'entité et la troisième le MinorType. /* If you want to add some things here, as you want */

FIGURE 2 – Exemple de Gazetier

4 Les expressions régulières

Les expressions régulières (en anglais "regular expressions" dont l'abrégé est regex) sont une famille de notations compactes et puissantes pour décrire certains ensembles de chaînes de caractères. Elles permettent de rechercher automatiquement des morceaux de texte ayant certaines formes, et éventuellement remplacer ces morceaux de texte par d'autre. Les expressions régulières sont utilisées par un grand nombre d'éditeurs de textes et utilitaires (particulierement sous Unix), par exemple Vim, Emacs, sed ou awk. On les retrouve également dans la majorité des langages de programmation modernes, soit sous forme de bibliothèque externe, soit directement implémentées dans le langage.
Exemple d'utilisation d'une expression régulière sur un bout de texte :

- Expression : `(?:\d*\.)?\d+`
- Texte : **10**rats + **.36**geese = **3.14**cows

II Outils

Il est indiqué dans notre cahier de charge, que l'annotation des textes, devra se faire dans l'environnement GATE. Il est donc impératif pour nous, de présenter cet outil.

1 L'environnement GATE

GATE est un logiciel développé en Java à l'université de Sheffield en 1995 et utilisé pour le traitement du langage naturel, y compris l'extraction d'information dans de nombreuses langues.

Fonctionnement général

C'est un outil qui repose sur le principe d'une chaîne de traitement ("pipeline") composé de plusieurs modules (dits "Processing ressources" PR) appliqués successivement sur un ou plusieurs textes (dits "langages Ressources" LR). Les documents donnés en entrée peuvent être un simple texte ou un corpus en local (cf. figure), que l'on charge avec une URL. Les différents composants annotent chacun à leur tour le texte en prenant en compte les annotations précédentes puis le document est retourné a l'utilisateur au format XML. Par un système de plugins, GATE met à disposition de ses

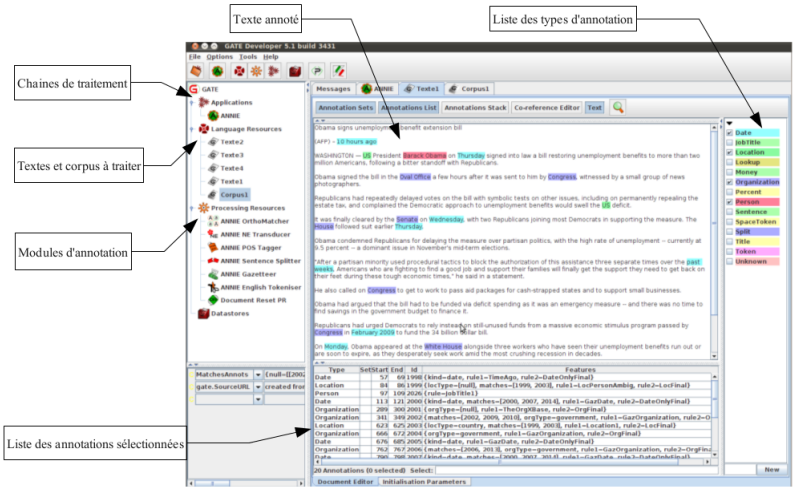


FIGURE 3 – Exemple de texte annoté dans l'outil GATE

utilisateurs un grand nombre de modules dédiés à l'analyse textuelle. Les plus courants sont les segmenteurs (Tokenizers), les étiqueteurs morpho-syntaxiques (Part Of Speech Taggers), les lexiques (Gazetteers) ou encore les transducteurs (JAPE transducers). L'interface graphique permet de charger de nouveaux plugins et ressources, de les paramétrer et de les combiner au sein d'une même chaine de traitement (cf. figure). GATE comprend entre autres un système d'extraction d'information nommé ANNIE. Il est constitué d'un certains nombre de modules, y compris ceux énoncé dans le paragraphe précédent. ANNIE est une chaîne de traitement par défaut et peut servir de point de départ pour des tâches plus spécifiques. Ci-dessous une représentation de ANNIE :

2 Le formalisme JAPE

Une partie des différents modules proposés dans GATE est basé sur le formalise JAPE (Java Annotation Patterns Engine), un transducteur à états finis

Figure 2.5 : Exemple de chaîne de traitement GATE

FIGURE 4 – Processus d'annotation

permettant de reconnaître des expressions régulières sur les annotations. ce système s'avère très utile en extraction de l'information car il permet de définir les contextes d'apparition des éléments à extraire pour ensuite les repérer et les annoter. le principe consiste a combiner différentes annotations dites basiques (tokens, relations syntaxiques, etc) pour en créer de nouvelles plus complexes (entités nommées, relations, événements, etc.) : Cela revient à l'écriture de règles de production et donc à l'élaboration d'une grammaire régulière.
Une grammaire JAPE se décompose en plusieurs phases exécutées consécutivement et formant une cascade d'automates à états finis. Chaque phase correspond à un fichier « .jape » et peut être constituée d'une ou plusieurs règle(s) écrite(s) selon le formalisme associé à JAPE. Classiquement, ces règles sont divisées en deux blocs : une partie gauche (« Left Hand Side » ou LHS) définissant un motif d'annotations à repérer et une partie droite (« Right Hand Side » ou RHS) contenant les opérations à effectuer sur ce motif. Le lien entre ces deux parties se fait par l'attribution d'une étiquette au motif (ou à ses constituants) en LHS et par sa réutilisation en RHS pour y appliquer les opérations nécessaires. Pour plus de clarté, prenons l'exemple d'une règle simple :

```
1. Rule: OrgAccronym
2. ((
3. {Organisation}
4. {Token.string == "("}
5. ({Token.orth == "allCaps"}):org
```

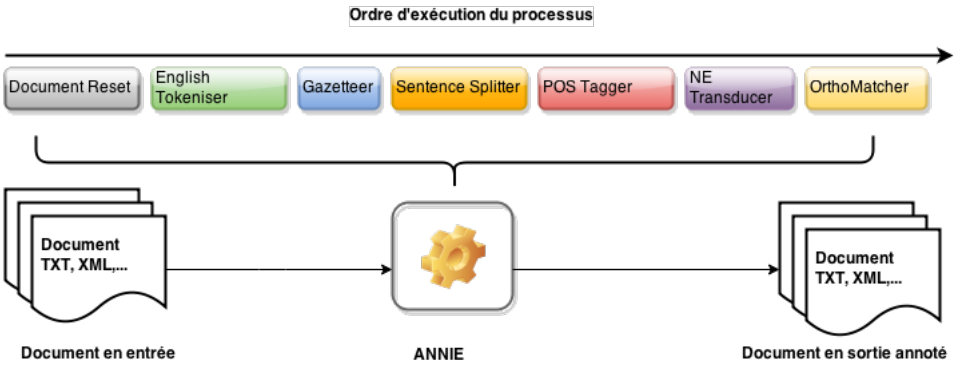


FIGURE 5 – Représentation de la chaîne de traitement ANNIE

```
6. {Token.string == ")"}
7.  )
8.  -->
9. :org.Organisation = {rule = "OrgAcronym"}
```

L'objectif de cette règle est d'annoter en tant qu'organisation les acronymes entre parenthèses positionnés après une annotation de type "Organisation". Tout d'abord, l'on commence par donner un nom à la règle (l1). Les lignes 2 à 7 définissent le motif à repérer dans le texte. Le signe `-->` (ligne 8) sert de séparateur entre LHS et RHS. Enfin, la dernière ligne (ligne 9) exprime l'opération souhaitée. Précisons quelques règles syntaxiques de base du formalisme JAPE :

- La partie gauche de la règle est toujours entre parenthèses
- La partie droite commence par le signe `"-->"`
- Les types d'annotation sont encadrés par des accolades
- `"Token.string"` permet d'obtenir la valeur de la propriété `"string"` associé à l'annotation `"Token"`
- `":org"` permet d'identifier une partie du motif en LHS pour l'utiliser en RHS
- La ligne 9 attribue une annotation de type "Organisation" au segment étiqueté "org" en LHS ; l'ajout de la propriété "rule" à cette annotation permet d'indiquer quelle règle en est à l'origine.

La liste des annotations utilisées en LHS de la règle est déclarée en début de phase grâce à l'attribut « Input » : par exemple, « Input : Lookup, Token, Person ». Par ailleurs, l'attribut « Control » permet de définir l'ordre d'exécution des différentes règles d'une phase et « Debug » d'obtenir un affichage des éventuels conflits rencontrés entre règles. Pour finir, précisons qu'un système de macros est également disponible : une macro permet de définir et de nommer une séquence d'annotations afin de la réutiliser plus

rapidement par la suite.Ci-dessous un exemple de règle JAPE plus complet :

```
1. Phase: MatchingStyles
2. Input: Lookup
3. Options: control = first
4. Rule: Test1
5. (
6. {Lookup.majorType == location}
7. ({Lookup.majorType == loc_key})?
8. ):match
9. -->
9. :match.Location = {rule=Test1}
```

3 RegexEr : Outil en ligne pour manipuler des expressions régulières

L'avantage de cet outil est le fait de pouvoir prendre en main, assez rapidement et de manière aisée la manipulation des expressions régulières.

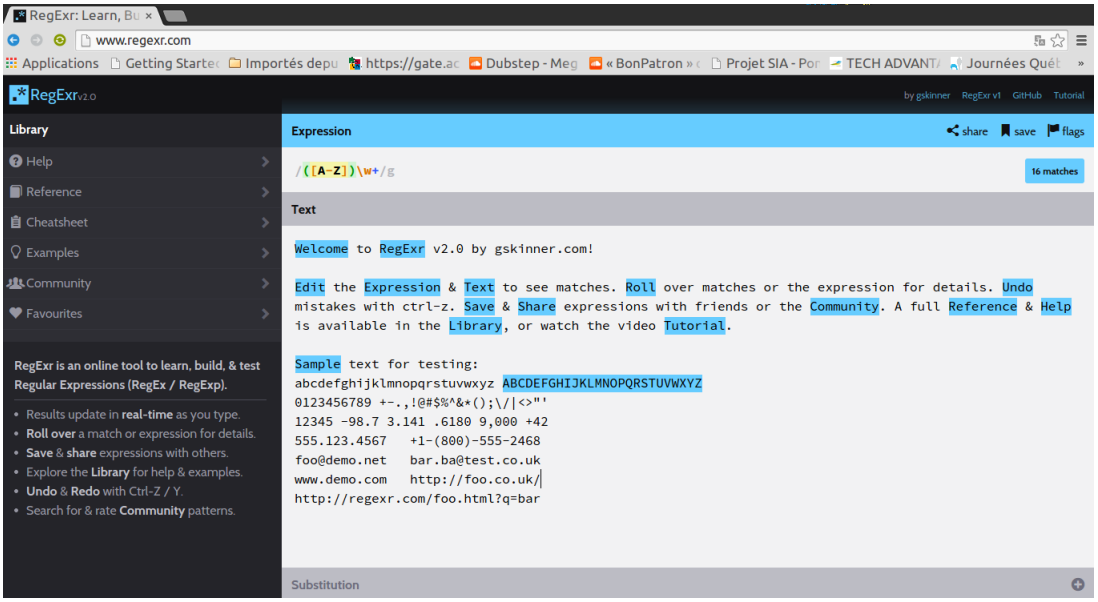


FIGURE 6 – Exemple de manipulation d'expression régulière dans RegExr

Quatrième partie

Développement

Introduction

Avant de démarrer le développement de la chaîne de traitement, nous devons nous familiariser avec notre environnement de travail. Ensuite, nous allons pouvoir définir nos annotations, leur domaine et choisir de quelle façon nous allons les capturer.

I Prise en main de l'environnement

Au lancement du projet, la seule ressource a disposition été le Numéraire Carolingien au format papier. Il fallait le numériser et l'OCRiser.
Nous avons numérisé une cinquantaine de pages à la main pour les OCRiser automatiquement par la suite à l'aide de l'outil `tesseract`.

L'OCRisation s'est déroulée de la façon suivante :

La scanner que nous avons utilisé permettait d'obtenir une image pour chacune des pages au format PDF. Ensuite, étant donné que `tesseract` est plus performant et précis avec des fichiers TIFF, il a fallu convertir les fichiers PDF en TIFF. Cependant, quelques erreurs d'OCRisation sont apparues. Pour finir, `tesseract` nous donnait des fichiers TXT.
Les étapes de conversion du schéma précédent ont été réalisées à l'aide de script en langage SHELL :

1. (a)
```
for file in *.pdf
do
  convert -density 300 ../pdf/$file -depth 8 ‘basename $file .pdf‘.tiff
done
```

   (b) les pages impaires ont été numérisée à l'envers, il fallait les mettre dans le bon sens.

```
for file in *.tiff
do
    if [ $((‘basename $file .tiff‘ % 2 )) = 1 ]; then
        convert $file -rotate 180 $file;
    fi
done
```

2.
```
for file in img/*.tiff
do
```

```
  tesseract $file txt/‘basename $file .tiff‘ -l fra
done
```

II Définition des dimensions d'annotation et leur contenu

Dans l'introduction, nous avons vu qu'il y a trois dimensions d'annotation mais ne les avons pas définies.
Note : Le détail du de chaque annotation peut être trouvé dans les parties II et IV.

1 Dimension spatiale

Définition

La dimension spatiale contient toutes les informations de lieux. Peu spécifique au domaine considéré.

Contenu

Le contenu de cette dimension est minime. Hormis les ateliers appartenant à la dimension thématique, les lieux présents dans le document n'intéressent que très peu la maîtrise d'ouvrage. Nous n'avons donc pas de contenu pour cette dimension.

2 Dimension temporelle

Définition

La dimension temporelle contient toutes les informations de temps et de durées. Peu spécifique au domaine considéré.

Contenu

les périodes d'émission des monnaies,

les périodes de règne des souverains.

3 Dimension thématique

Définition

La dimension thématique contient toutes les informations spécifiques domaine considéré.

Contenu

La nature des monnaies,

La légende des monnaies

Le type des monnaies,

Les collections des monnaies,

Les trésors des monnaies,

Les trouvailles des monnaies.

III Première recherche d'entités nommées avec les gazetiers

Ateliers

■ *Vulgarisation :*

L'ouvrage est décomposé en ateliers qui donnent leur nom à chaque début de "partie" ou "chapitre". Ce nom correspond à un endroit de France ou pays limitrophes dans lequel est produite la monnaie. Ce lieu peut être une ville, un lieu-dit dont le nom peut ne plus exister. Il fût donc difficile de trouver un pattern via les expressions régulières. Nous avons alors dû construire ce gazetier à partir de la liste en début d'ouvrage qui recense tous les ateliers.
■

■ *Formalisation :*

```
Aix-la-Chapelle (Allemagne)
Agen (Lot-et-Garonne)
Aix-la-Chapelle
Alsheim (Allemagne)
Altenheim (Bas-Rhin)
Amiens (Somme)
```

■

Souverains

■ *Vulgarisation :*

Le nom des souverains ont des formats aussi divers que variés.Ils comportent des majuscules, des chiffres romains... Il serait difficile d'utiliser une expression régulière pour espérer annoter cette information. Il est plus judicieux

d'utiliser un gazetier. Il sera construit à partir du début du numéraire.

■

■ *Formalisation :*

```
Pépin le Bref:valeur=Pépin le Bref:periode=752-768
Adalbert Lothaire:valeur=Adalbert Lothaire:periode=954-986
Amoul roi de Germanie:valeur=Amoul roi de Germanie:periode=887-899
Bérenger I:valeur=Bérenger I:periode=888-924
Bérenger II:valeur=Bérenger II:periode=950-961
```

■

IV Deuxième recherche d'entités nommées avec les règles JAPE

Périodes

■ *Vulgarisation :*

Période : intervalle de deux dates séparées par un tiret.
Exemple :
757/8-786

Une période est un intervalle entre deux dates. Dans notre travail, les dates sont constituées de trois et seulement trois chiffres. En cas d'ambiguïté,Chacune d'elle peut être suivie d'un « / » et d'un chiffre traduisant l'indétermination de la date.
Exemple :
757/8

■

■ *Formalisation :*
Date

```
([0-9]{3}\/?[0-9]?)
```

Période

```
([0-9]{3}(\/[0-9])?)-([0-9]{3}(\/[0-9])?)
```

Exemple :
" *Type de 771-793/4 : Charlemagne (768-814),...*"

Group #1 : 771
Group #2 : 793/4

Group #1 : 768
Group #2 : 814

■

■ *Règle JAPE :*

```
     // Regle JAPE
   Macro: TROIS_NOMBRES
({Token.kind==number,Token.length == 3})

Macro: UN_NOMBRE
({Token.kind==number,Token.length == 1})

Macro:SLASH
({Token.string=="/"})

Macro:DATE_PRECISE
(TROIS_NOMBRES)

Macro:DATE_IMPRECISE
(TROIS_NOMBRES SLASH UN_NOMBRE)

Macro:DATE
(DATE_PRECISE | DATE_IMPRECISE)

Rule: PeriodeRule
(
   (DATE):d1({Token.string =="-"})(DATE):d2
    ):Periode -->
:Periode{/*Code java pour extraire les extremites de l'intervalle*/}
```

■

Périodes d'émission

■ *Vulgarisation :*

Une période d'émission a la forme suivante : « Type de {Période} : » (Période étant l'annotation définie précédemment).
Il faut sécuriser la capture de la période d'émission en ajoutant la contrainte précédée de "Type de" afin de ne pas récupérer toutes les périodes du documents.
■

■ *Formalisation :*

`Type de : ([0-9]{3}(\/[0-9])?)-([0-9]{3}(\/[0-9])?)`

Exemple :
" Type de 771-793/4 : Charlemagne (768-814)..."
Group #1 : 771
Group #2 : 793/4

■

■ *Règle JAPE :*

```
   // Regle JAPE
   Macro: CHAINE_DEBUT
(
    ({Token.string =="Type"})({SpaceToken})
    ({Token.string =="de"})({SpaceToken})
)

Rule: PeriodeEmissionRule
(
    CHAINE_DEBUT ({Periode}):p
):PeriodeEmission
-->
:PeriodeEmission.PeriodeEmission = { Kind = "PeriodeEmission" ,D1 =
    :p.Periode.D1, D2 = :p.Periode.D2}
```

■

Périodes de règne

■ *Vulgarisation :*

Une période de règne a la forme suivante : « Nom_souverain ({Période}) : ». Il faut sécuriser la capture de la période de règne en ajoutant la contrainte précédée de "Souverain" afin de ne pas récupérer toutes les périodes du documents.
Exemple :
" *Type de 771-793/4 : Charlemagne (768-814)...*"

■

■ *Formalisation :*

```
Nom_souverain ([0-9]{3}(\/[0-9])?)-([0-9]{3}(\/[0-9])?)
```

Exemple :
" *..Charlemagne (768-814)...*"
Correspondance : Charlemagne
Group #1 : 768
Group #2 : 814

■

Nature de la monnaie

■ *Vulgarisation :*

La nature de la monnaie est toujours un élément de l'ensemble {Denier, Obole, Monnaie D'Or, Faux obole, Monnaies de type indéterminé} suivi du nom d'un souverain. Il suffit donc d'utiliser une expression composée de tous les mots de l'ensemble.
Exemple :
" *Obole de Charles le Chauve*"

■

■ *Formalisation :*

```
[\s]{2,}(Denier|Obole|Monnaie d'or|Faux Obole|
Monnaies de type indéterminé)(.*)? Nom_Souverain
```

Exemple :
" *Obole de Charles le Chauve*"
Correspondance : Charles le Chauve

Group #1 : Obole

■

Légende

■ *Vulgarisation :*

La légende est toujours placée sous la ligne de la nature de la monnaie. La légende du droit est située au début de cette ligne et commence par zéro ou un caractère +. Ensuite, vient une suite d'espaces. Enfin, la légende du revers vient se placer après zéro ou un caractère +.
Exemple :
"Denier de Charlemagne .....
+ CARLO 45E CROIX SIMPLE"

■

■ *Formalisation :*

1. Se positionner à la ligne qui suit la nature de la pièce

   `(?:Denier|Obole|Monnaie d'or).*\n`

2. Capturer l'ensemble des caractères entre 0 ou 1 fois le symbole + et 2 espaces ou plus. C'est la légende du droit.

   `\+?\s?(.*)[ ]{2,}`

3. Capturer l'ensemble des caractères entre 0 ou 1 fois le symbole + et la fin de ligne. C'est la légende du revers.

   `\+?\s?(.*)\n`

On obtient une expression comme suit :

```
(?:Denier|Obole|Monnaie d'or).*\n \s*\+?\s?(.*)[ ]{2,} \+?\s?(.*)\n
```

Exemple :
"Denier de Charlemagne .....
+ CARLO 45E CROIX SIMPLE"
Group #1 : CARLO 45E
Group #2 : CROIX SIMPLE

■

Types monétaire

■ *Vulgarisation :*

Le type monétaire est la concaténation de "Type de" avec la période d'émission. Cette annotation est similaire à la période d'émission mais appartient à la dimension thématique.
Exemple :
" Type de 771-793/4 : Charlemagne (768-814),..."

■
■ *Formalisation :*

```
(Type de [0-9]{3}\/?[0-9]?-[0-9]{3}\/?[0-9]?)
```

Exemple :
"Type de 771-793/4 : Charlemagne (768-814),...
Group #1 : Type de 771-793/4

■

Collections, Trésors, Trouvailles

■ *Vulgarisation :*

Les collections, trésors et trouvailles sont chacun des ensembles d'informations à annoter séparément. Les *mots* Collections, Trésors et Trouvailles sont chacun suivis de deux points. Ensuite, vient le contenus concernant ces mots. Le contenu s'arrête lorsqu'on rencontre un point suivi d'un retour à la ligne ou bien un autre *mot* suivi de deux points.
■
■ *Formalisation :*

```
Mot_a_trouver:((?:.|\n)+?)(?:\.\s?\n|\w:)
```

Exemple :
"...Collections : Berlin 1,77, 1,70, 1,59, 1,55 ; MEC 853 (1,78) ; Monnaie de Paris 105 (1,63) ; Prou 584 (1,58), 585 (1,69), 586 (1,79), 587 (1,72), 588 (1,77). Trésors :..."
Group #1 : Berlin 1,77, 1,70, 1,59, 1,55 ; MEC 853 (1,78) ; Monnaie de Paris 105 (1,63) ; Prou 584 (1,58), 585 (1,69), 586 (1,79), 587 (1,72), 588 (1,77).
Trésors :

Cinquième partie

Conclusion