Spécification du modèle de donnée PASSAGE++

mise à jour du 13 juin 2010

Gil Francopoulo / Tagmatica

1 Introduction

Pour mémoire, le modèle de donnée Passage permet de représenter l'analyse morphosyntaxique et syntaxique d'une phrase. Le modèle Passage++ est une extension de Passage afin de couvrir certains phénomènes sémantiques qui sont :

* l'unité de sens,

* la coréférence,

* la citation (ou prises de parole).

Toutes les parties sémantiques sont optionnelles, ce qui signifie qu'un document Passage valide (au sens d'XML) est d'emblée un document Passage++ valide.

2 Le modèle Passage

2.1 Historique

Le modèle Passage a été spécifié dans le cadre de la campagne d'évaluation des analyseurs syntaxiques du projet ANR-Passage, voir http://atoll.inria.fr/passage/documents.fr.html.

2.2 Architecture

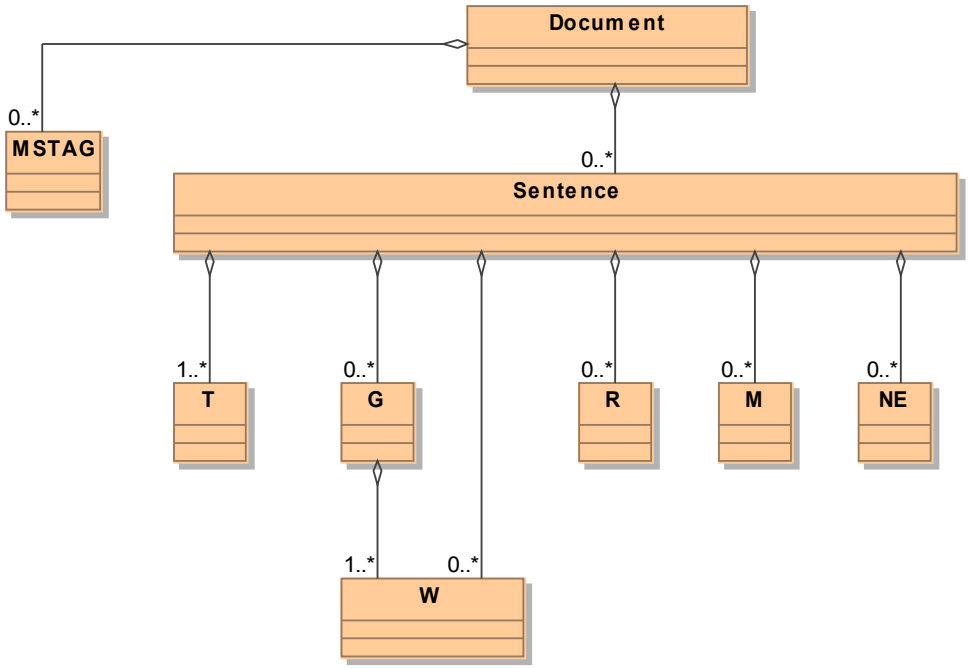
Le format est multi-niveaux de manière hiérarchique, à savoir :

- Niveau Mot Forme
- Niveau Groupe, au sens de groupe syntaxique
- Niveau Sentence
- Niveau Document

De manière non hiérarchique, nous avons :

- le token
- la relation
- la marque
- l'entité nommée
- la combinaison de traits morpho-syntaxiques

La structure est synthétisée par le diagramme de classe UML suivant [1]:



Pour simplifier la présentation, les enchâssements des relations ne sont pas dessinés, voir pour cela la DTD plus loin.

Selon les niveaux, un mécanisme de notation enchâssée (i.e. 'embedded' notation ou 'in line' notation) ou de notation déportée (i.e. 'stand off' notation) est utilisé. Le critère de choix entre les deux est un compromis entre la puissance d'expression et la lisibilité. Une notation déportée est plus puissante mais moins lisible.

Tout élément référencé doit apparaître avant son référenceur pour éviter aux développeurs qui utilisent des dispositifs d'analyse XML à la volée comme SAX (Simple API for XML) ou StAX (Streaming API for XML) d'avoir à gérer des références en avant dans le flux XML [6]. Evidemment, pour les développeurs qui utilisent des dispositifs d'analyse en mode DOM (Document Object Model) cette contrainte n'offre aucun intérêt mais ne les empêche pas d'analyser le fichier XML.

Le format respecte les deux spécifications ISO en cours d'élaboration qui sont MAF (ISO 24611) et SynAF (ISO 24615)[3][4]. Comme toutes les spécifications en cours d'élaboration au sein de l'ISO-TC37/SC4, les structures définies doivent être décorées par des catégories de données définies dans le registre de l'ISO (i.e. data category registry, i.e. DCR) [2].

2.3 Les niveaux d'annotation

2.3.1 Niveau Token

Un token est une chaîne de caractères minimale. Elle est minimale dans le sens où un éventuel découpage en fragments plus petits n'est pas réalisé car il n'offre aucun intérêt. Le token est la plus petite unité adressable via un identifiant. Le choix de l'algorithme de segmentation du texte en tokens n'est pas imposé, mais un algorithme est conseillé, voir plus loin.

Chaque token est décrit par un identificateur, un empan et un contenu. Un empan est un couple position du premier caractère / position après le dernier caractère **dans le texte d'origine**. La position est donnée à partir de zéro en nombre de caractères et non en nombre d'octets[1]. Tous les caractères sont comptés, qu'ils soient des lettres, des espaces ou des fins de ligne. Le contenu de

1 Rappelons que si du temps de l'ISO Latin, les notions de caractères et d'octets étaient confondues, ce n'est plus le cas dans le contexte actuel qui suit Unicode, voir par exemple [5]. En Unicode, un caractère est représenté à l'aide d'un ou plusieurs octets. Un octet fait toujours huit bits.

la balise est la chaîne de caractères telle qu'elle apparaît dans le texte d'origine.

Par exemple, à partir du texte d'entrée « **Les chaises** », nous aurons :

```
<T id="t0" start="0" end="3">Les</T>
<T id="t1" start="4" end="11">chaises</T>
```

2.3.2 Niveau mot forme ou forme fléchie

La forme fléchie est une annotation construite à partir de la notion de token. Une forme fléchie n'est pas toujours équivalente à un token, même si c'est fréquemment le cas. Ainsi, les tokens «aujourd», «'»et «hui» pourront ne former qu'une seule forme fléchie, de même que les tokens «afin» et «de». Inversement, un même token peut être référencé par différentes formes fléchies.

La forme fléchie est un niveau intermédiaire entre le token et le groupe. Alors que le Token est défini par un algorithme de segmentation, la forme fléchie est définie par sa présence dans le dictionnaire ou bien sa reconnaissance en tant qu'entité nommée.

Dans le format d'annotation, chaque forme fléchie est décrite par un identifiant et des références strictement consécutives à un ou plusieurs tokens.

Sur le même texte que précédemment, et en supposant que les mots figurent en début de document, nous aurons :

```
<T id="t0" start="0" end="3">Les</T>
<W id="w0" tokens="t0"/>

<T id="t1" start="4" end="11">chaises</T>
<W id="w1" tokens="t1"/>
```

De manière optionnelle, la forme fléchie peut porter la partie du discours, le lemme, la forme fléchie canonique après redressement éventuel, une combinaison de traits morphologiques et un booléen indiquant si le mot est une tête pour le groupe. La combinaison doit être déclarée en en-tête du fichier, voir plus loin, la section sur la combinaison des traits.

Dans ce cas, nous aurons :

```
<T id="t0" start="0" end="3">Les</T>
<W id="w0" tokens="t0" pos="definiteArticle" lemma="le" form="les" mstag="nP"/>

<T id="t1" start="4" end="11">chaises</T>
<W id="w1" tokens="t1" pos="commonNoun" lemma="chaise" form="chaises" mstag="nP gF" head="true"/>
```

Les unités multi-mots sont représentées par un ensemble de tokens référencés par une unique forme fléchie. Par exemple, pour "aujourd'hui", nous aurons :

```
<T id="t0" start="0" end="7">aujourd</T>
<T id="t1" start="7" end="8">'</T>
<T id="t2" start="8" end="11">hui</T>
<W id="w0" tokens="t0 t1 t2" pos="adverb" lemma="aujourd'hui" form="aujourd'hui"/>
```

Le dispositif permet de représenter une chaîne de caractères qui donne lieu à un redressement

orthographique par éclatement. Un même token est alors référencé par plusieurs formes fléchies. Ainsi, en imaginant que, par erreur, la chaîne d'entrée soit: "unetable", il est possible de représenter la décomposition suivante :

```
<T id="t0" start="0" end="8">unetable</T>
<W id="w0" tokens="t0" pos="indefiniteArticle" lemma="un" form="une"/>
<W id="w1" tokens="t0" pos="commonNoun" lemma="table" form="table"/>
```

Il est possible de représenter les agglutinés comme "au" par "à" et "le" de la manière suivante :

```
<T id="t0" start="0" end="2">au</T>
<W id="w0" tokens="t0" pos="preposition" lemma="à" form="à"/>
<W id="w1" tokens="t0" pos="indefiniteArticle" lemma="le" form="le"/>
```

Les références aux tokens doivent être strictement consécutives. Cela signifie qu'il n'est pas possible qu'un mot adresse un token qui se trouve entre deux tokens d'un autre mot. En d'autres termes, il n'est pas possible de représenter un "composé à trou" comme dans le **contre-exemple** suivant :

```
<!--Rappel: ceci est interdit-->
<T id="t0" start="0" end="2">ne</T>
<T id="t1" start="3" end="11">vraiment</T>
<T id="t2" start="12" end="15">pas</T>
<W id="w0" tokens="t0 t2"/>
<W id="w1" tokens="t1"/>
<!--Rappel: ceci est interdit-->
```

Concernant la liste des parties du discours, le registre de l'ISO contient de multiples valeurs, par exemple "bullet" ou "comma" qui sont organisées en une ontologie. Nous choisissons de ne prendre qu'une liste relativement simple et plate.

Les parties du discours sont les suivantes :

identifiant ISO	nom français

adverb	adverbe
commonNoun	nom commun
coordinatingConjunction	conjonction de coordination
definiteArticle	article défini
demonstrativeDeterminer	déterminant démonstratif
demonstrativePronoun	pronom démonstratif
exclamativeDeterminer	déterminant exclamatif
foreignText	texte étranger
formula	formule
fusedPrepositionPronoun	préposition pronom fusionnés
impersonalPronoun	pronom impersonnel
indefiniteDeterminer	déterminant indéfini
indefinitePronoun	pronom indéfini
interjection	interjection
interrogativePronoun	pronom interrogatif
letter	lettre
mainPunctuation	ponctuation principale
negativeParticle	particule négative
numeral	numéral
ordinalAdjective	adjectif ordinal
partitiveArticle	article partitif
personalPronoun	pronom personnel
possessiveDeterminer	déterminant possessif
possessivePronoun	pronom possessif
preposition	préposition
properNoun	nom propre
qualifierAdjective	adjectif qualificatif
reflexivePersonnalPronoun	pronom réflexif personnel
relativePronoun	pronom relatif
residual	résiduel
secondaryPunctuation	ponctuation secondaire
subordinatingConjunction	conjonction de subordination
verb	verbe

2.4 Niveau Groupe

Il s'agit de représenter un groupe syntaxique. Ce groupe peut être un groupe non récursif ou bien un syntagme récursif.

Un groupe est constitué d'un identifiant, d'un type syntaxique, d’une suite de formes fléchies ou de groupes. Un groupe possède au moins une forme fléchie ou un groupe, autrement dit, un groupe ne peut pas être vide.

```
<T id="t0" start="0" end="3">Les</T>
<T id="t1" start="4" end="11">chaises</T>
<G id="g0" type="GN">
        <W id="w0" tokens="t0"/>
        <W id="w1" tokens="t1"/>
</G>
```

L'attribut 'type' est valué par une des constante de la liste donnée ci-dessous. Chaque valeur trouve un correspondant dans le registre de catégorie de données de l'ISO (DCR). Dans le tableau qui suit, la colonne de gauche est la valeur utilisée dans le projet Easy/Passage, la colonne du milieu, le nom français dans le DCR et la colonne de droite est l'identifiant dans le DCR.

nom Easy	nom ISO français	identifiant ISO
NV	noyau verbal	verbNucleus
GN	groupe nominal	nounPhrase
GP	groupe prépositionnel	prepositionalPhrase
GA	groupe adjectival	adjectivePhrase
GR	groupe adverbial	adverbPhrase
PV	groupe verbal prépositionnel	prepositionalVerbPhrase

De plus, les valeurs GV (pour groupe verbal), GD (pour groupe déterminant) et CL (pour clause, ou proposition) peuvent être utilisées. Notons que ces valeurs n'existaient pas dans la campagne Easy puisque cette dernière ne permettait que l'annotation en chunks (i.e. groupes non-récursifs) sans aucune possibilité pour un groupe d'avoir des sous-groupes.

De manière optionnelle, le groupe peut porter une combinaison de traits morpho-syntaxiques comme par exemple :

```
<T id="t0" start="0" end="3">Les</T>
<T id="t1" start="4" end="11">chaises</T>
<G id="g0" type="GN" mstag="nP gF">
        <W id="w0" tokens="t0"/>
        <W id="w1" tokens="t1"/>
</G>
```

Notons que lors de la campagne Technolangue/Easy, les groupes étaient enchâssés dans une balise "groupes". Mais lors de la première campagne de Passage, cette balise a disparu, nous avons fait de même.

2.5 Niveau Relation

C'est une notation déportée (i.e. 'stand-off'). Pour des raisons de compatibilité avec les outils de la campagne Easy, le format général des relations est conservé.

Les arguments référencés sont aussi bien des formes fléchies que des groupes, mais au contraire du format Easy, les références sont de véritables références XML. En effet, dans le format Easy, les références avaient été déclarés par erreur en type CDATA optionnel, ce qui faisait que les outils XML ne pouvaient pas vérifier le typage des éléments.

Par rapport à la DTD Easy, les éléments "adjectif" et "complementeur" ont été ajoutés car ils étaient mentionnés dans le contenu de l'élément "relation" sans être définis. Pour détecter automatiquement ce type de problème, un validateur XML[2] a été utilisé.

En supposant que l'on veuille définir une relation sujet entre le groupe g0 et g1, nous aurons la structure suivante :

2 Bonfire Studio en l'occurrence

```
<R id=”r0" type="SUJ_V">
        <sujet  ref="g0"/>
        <verbe ref="g1"/>
</R>
```

L’attribut ‘type’ est valué par une des constantes de la liste donnée ci-dessous. Chaque valeur trouve un correspondant dans le registre de catégorie de données de l’ISO (DCR). Dans le tableau qui suit, la colonne de gauche est la valeur utilisée dans le projet Easy/Passage, la colonne du milieu, le nom français dans le DCR et la colonne de droite est l’identifiant dans le DCR.

nom Easy	nom ISO français	identifiant ISO
SUJ_V	sujet	subject
AUX_V	auxiliaire	auxiliary
COD_V	objet direct	directObject
CPL_V	complément verbal	verbComplement
MOD_V	modifieur de verbe	verbModifier
COMP	complémenteur	complementizer
ATB_SO	attribut	attribute
MOD_N	modifieur de nom	nounModifier
MOD_A	modifieur d'adjectif	adjectiveModifier
MOD_R	modifieur d'adverbe	adverbModifier
MOD_P	modifieur de préposition	prepositionModifier
COORD	coordination	coordination
APP	apposition	apposition
JUXT	juxtaposition	juxtaposition

Le rôle est indiqué par l'une des valeurs figurant ci-dessous :

nom Easy	nom ISO français	identifiant ISO
adjectif	adjectif	adjective
adverbe	adverbe	adverb
appose	apposé	apposed
attribut	attribut	attribute
auxiliaire	auxiliaire	auxiliary
cod	objet direct	directObject
complement	complément verbal	verbalComplement
complementeur	complémenteur	complementizer
coord-d	coordonné droit	rightCoordinated
coord-g	coordonné gauche	leftCoordinated
coordonnant	coordonnant	coordinator
modifieur	modifieur	modifier
nom	nom	noun
premier	premier	first
preposition	préposition	preposition
suivant	suivant	next
sujet	sujet	subject
verbe	verbe	verb

2.6 La pose de repères

Un repère est un dispositif qui permet à un annotateur humain d'enregistrer un commentaire qui lui permettra de localiser un fragment du texte afin d'y revenir par la suite. Un repère peut aussi être enregistré automatiquement par un programme. Un repère référence soit un empan, soit un ou plusieurs éléments du fichier via les identifiants.

Nous aurons par exemple :

```
<M id=”m0” start="0" end="3" objs="">à valider avec Robert<M/>
```

2.7 Les entités nommées

En terme de format, une entité nommée est une annotation qui possède a minima un identifiant, un type et une liste de références de mot-formes. Ces références ne sont pas nécessairement continues mais leur portée ne doit pas excéder la phrase.

Les types possibles sont les suivants:

identifiant	définition	note	exemples
individual	nom d'être vivant	C'est en général un nom de personne mais peut désigner un animal ou une plante. L'être vivant peut être réel ou imaginaire.	Jacques Chirac Chirac
organization	société ou institution	La société peut être privée, publique ou mixte (ex GIE).	Danone FMI
location	nom de lieu ou de l'entité géopolitique qui lui est associée	On inclut aussi les continents, les rivières et les planètes.	Dijon Suisse Grande-Bretagne
dateTime	dates et heures	On ne distingue pas les dates des heures. Les dates et heures peuvent être agrégées, ex: 5 février 14h. On n'a alors qu'une seule entité. Les fêtes du calendrier sont considérées comme des dates et non des événements.	mardi 5 février 5h 10 Noël
URLetc	nom de fichier ou d'URL		www.afnor.org
measure	toutes les mesures du moment qu'il y a la mention d'une unité	On inclut aussi les sommes d'argent et les pourcentages.	2,66 GHz $3 million
mark	nom de marque, de protocole ou de format	C'est un peu une catégorie fourre tout. Les marques commerciales comme Chanel sont considérées comme des organisations, puisqu'il est impossible de les distinguer. En définitive, c'est un nom de marque quand ce n'est pas aussi le nom d'une organisation. La situation se complique quand c'est aussi le nom d'un individu comme "Armani".	PDF
event	nom d'événement		Tour de France 2010 championnat de France
work	nom d'oeuvre		le film "Avatar"
unnamed	nom commun	Ce n'est pas vraiment une entité nommée stricto sensus mais le mot a en général un référent, ce qui le distingue d'un adjectif ou d'un adverbe.	chaise

C'est une partition dans la mesure où un même empan ne peut être qualifié de deux types différents.

De manière optionnelle, l'entité nommée possède une combinaison de traits morpho-syntaxiques comme le genre ou le nombre.

De manière optionnelle, l'entité nommée peut porter une information décrivant plus en détail le type de l'entité. C'est l'activité qui peut prendre une ou plusieurs valeurs parmi les suivantes, pour l'instant :

pour le domaine sportif	athlete

	golfer
	rugbyPlayer
	soccerPlayer
	swimmer
	tennisPlayer
pour le domaine politique	politician
pour l'information	journalist

Notons que la détermination d'une liste de types d'entités nommées est une entreprise délicate. Le niveau de détail est très variable d'un logiciel à l'autre. Et, contrairement aux autres listes de valeurs, il n'existe pas de liste fixée par l'ISO: il y a bien un travail en cours pour la représentation des entités nommées mais ce dernier n'est pas assez avancé pour fixer une liste de valeurs. Le choix est pris de fixer une liste de niveau relativement général, sur lequel tout le monde semble à peu près d'accord.

L'entité nommée n'est pas un élément inclus dans un mot ou dans un groupe. Il n'est pas dans le 'flux' mais figure en quelque sorte de manière 'parallèle' aux groupes. Il est un élément d'annotation de 'première classe'.

Notons que la définition linguistique de ce qu'est une entité nommée et quelles sont les consignes pour la construire ne fait pas l'objet du présent document mais doit être défini par un guide d'annotation.

Si nous avons coïncidence entre le groupe et l'entité nommée, nous aurons sur "Jacques Chirac" :

```
<T id="t0" start="0" end="7">Jacques</T>
<T id="t1" start="8" end="13">Chirac</T>
<G id="g0" type="GN">
        <W id="w0" tokens="t0"/>
        <W id="w1" tokens="t1"/>
</G>
<NE id="e0" type="individual" lst="w0 w1"/>
```

Mais nous pouvons ne pas avoir une coïncidence exacte entre le groupe et l'entité nommée. En supposant que le guide d'annotation spécifie que l'article ne fait pas partie de l'entité nommée, nous aurons une entité nommée plus petite que le groupe. Ainsi sur "l'Hôtel Crillon" par exemple, nous aurons :

```
<T id="t0" start="0" end="2">l'</T>
<T id="t1" start="2" end="7">Hôtel</T>
<T id="t2" start="8" end="15">Crillon</T>
<G id="g0" type="GN">
        <W id="w0" tokens="t0"/>
        <W id="w1" tokens="t1"/>
        <W id="w2" tokens="t2"/>
</G>
<NE id="e0" type="location" lst="w1 w2"/>
```

Inversement, le format d'annotation permet de construire un empan plus grand que le groupe avec, par exemple sur "le jour de la Toussaint" avec l'entité nommée « jour de la Toussaint »:

```
<T id="t0" start="0" end="2">le</T>
<T id="t1" start="3" end="7">jour</T>
<T id="t2" start="8" end="10">de</T>
<T id="t3" start="11" end="13">la</T>
<T id="t4" start="14" end="23">Toussaint</T>
<G id="g0" type="GN">
        <W id="w0" tokens="t0"/>
        <W id="w1" tokens="t1"/>
</G>
<G id="g1" type="GP">
        <W id="w2" tokens="t2"/>
        <W id="w3" tokens="t3"/>
        <W id="w4" tokens="t4"/>
</G>
<NE id="e0" type="dateTime" lst="w1 w2 w3 w4"/>
```

Le format permet aussi de gérer des phénomènes plus complexes comme la factorisation d'une partie de l'entité qui intervient fréquemment dans les coordinations et les intervalles. Par exemple, en supposant l'entrée: "Bill et Hillary Clinton", il est possible de produire d'un côté "Bill Clinton" et de l'autre "Hillary Clinton". Ce serait une erreur de représenter d'un coté "Bill" et d'un autre coté "Hillary Clinton", car s'il s'agit de mémoriser "Bill" dans un index, plus aucune distinction n'est faite pour indiquer qu'il s'agit d'un Bill particulier qui est le mari d'Hillary et non de n'importe quel Bill. Une autre solution serait d'extraire "Bill et Hillary Clinton" d'un seul tenant mais cette solution est encore pire, car elle n'autorise plus les comparaisons d'entités nommées de manière fiable.

Les références des entités vers les mots-formes ne sont, dans le cas particulier de la factorisation, alors pas continues.

Le codage sera le suivant:

```
<T id="t0" start="0" end="4">Bill</T>
<T id="t1" start="5" end="7">et</T>
<T id="t2" start="8" end="15">Hillary</T>
<T id="t3" start="16" end="23">Clinton</T>
<G id="g0" type="GN">
        <W id="w0" tokens="t0"/>
</G>
        <W id="w1" tokens="t1"/>
<G id="g1" type="GN">
        <W id="w2" tokens="t2"/>
        <W id="w3" tokens="t3"/>
</G>
<NE id="e0" type="individual" lst="w0 w3"/>
<NE id="e1" type="individual" lst="w2 w3"/>
```

Notons que la coordination ne se limite pas aux entités de personnes mais concerne aussi les autres types comme les dates, dans "Il restera le 3 et le 4 avril". La factorisation d'une partie de l'entité concerne aussi les intervalles comme dans: "Le chiffre d'affaires sera compris entre 3 et 4 millions d'euros".

Un autre mécanisme couvert par le format est le recouvrement d'entités qui sont typées de manière différente. Ainsi, dans "la ville de Michelin" pour désigner Clermont Ferrand, nous avons une entité géographique "ville de Michelin" mais nous avons aussi la mention de la société "Michelin"[3]. Nous aurons donc:

3 Encore une fois, rappelons que le présent document se contente de définir le format. La consigne de représenter ou non le recouvrement doit être défini dans un guide d'annotation.

```
<T id="t0" start="0" end="2">la</T>
<T id="t1" start="3" end="8">ville</T>
<T id="t2" start="9" end="11">de</T>
<T id="t3" start="11" end="20">Michelin</T>
<G id="g0" type="GN">
        <W id="w0" tokens="t0"/>
        <W id="w1" tokens="t1"/>
</G>
<G id="g1" type="GP">
        <W id="w2" tokens="t2"/>
        <W id="w3" tokens="t3"/>
</G>
<NE id="e0" type="location" lst="w1 w2 w3"/>
<NE id="e1" type="organization" lst="w3"/>
```

2.8 La définition des combinaisons de traits morpho-syntaxiques

En début de fichier, de manière optionnelle, les combinaisons des traits morpho-syntaxiques (i.e. tagsets en anglais) pourront être définies.

Une combinaison porte un nom et un ensemble de traits (i.e. features) qui en forme le vocabulaire. Chacun de ces traits doit être une valeur du DCR. La syntaxe est celle de la spécification ISO pour les structures de traits [7].

Nous aurons, par exemple :

```
<MSTAG id="nP">
        <fs>
                <f name="grammaticalNumber">
                        <symbol value="plural"/>
                </f>
        </fs>
</MSTAG>
```

Les traits morpho-syntaxiques pour le français sont les suivants :

catégorie de donnée utilisée comme attribut	domaine conceptuel
grammaticalNumber	singular
	plural
grammaticalGender	masculine
	feminine
person	firstPerson
	secondPerson
	thirdPerson
verbFormMood	indicative
	conditional
	subjunctive
	imperative
	participle
	infinitive
grammaticalTense	present
	imperfect
	past
	future
ownerPerson	firstPerson
	secondPerson
	thirdPerson
ownerNumber	singular
	plural
ownedGender	masculine
	feminine
ownedNumber	singular
	plural

Il est possible de spécifier des disjonctions, par exemple dans les situations où l'on ne désire pas statuer sur un trait précis et conserver l'ambiguïté, comme dans le fragment XML suivant:

```
<MSTAG id="mI.S">
        <fs>
                <f name="grammaticalMood">
                        <vAlt>
                                <symbol value="indicative"/>
                                <symbol value="subjunctive"/>
                        </vAlt>
                </f>
        </fs>
</MSTAG>
```

2.9 Niveau Sentence ou phrase

C’est une notation enchâssée. Le choix de l’enchâssement plutôt que la notation déportée est motivé par l’intérêt de rassembler toutes les balises d’une même phrase dans la même partie du fichier, et ainsi en faciliter la lecture.

Une phrase est composée d'une suite non vide de groupes et de formes fléchies, suivie d'une suite éventuellement vide de relations, suivie d'une suite éventuellement vide de repères.

L'élément "Sentence" possède un attribut optionnel qui s'appelle "trust" qui permet d'indiquer le degré de confiance que le producteur de la donnée accorde à la qualité de l'analyse. La valeur de l'attribut varie de 0 (pas fiable) à 100 (fiable).

Nous aurons par exemple :

```
<Sentence id="e0" trust="100">
        <G>…</G>
        <G>…</G>
        <W>…</W>
        <G>…</G>
        <W>…</W>
        <R>…</R>
        <R>…</R>
        <M>…</M>
        <M>…</M>
</Sentence>
```

2.10 Niveau Document

Le niveau document se résume à une balise englobante des phrases du texte en question. La balise de document possède l'attribut suivant :

- un attribut qui est le nom de fichier. Ce nom n'est pas le nom avec le chemin d'accès au sens du système d'exploitation, mais au contraire, la forme relative.

```
<Document file="monfichier.xml">
        <Sentence …>…</Sentence>
        <Sentence …>…</Sentence>
</Document>
```

3 Le modèle Passage++

3.1 L'unité de sens

3.1.1 aspect linguistique

Il s'agit de mentionner le sens du mot. C'est un des sens du mot tel qu'il est décrit dans le dictionnaire interne de l'analyseur.

Un mot peut se trouver dans l'un des trois cas de figure suivants:

- polysèmie. Cela signifie que le mot possède plusieurs sens mais que l'analyseur a choisi l'un de ces sens. Il s'agit par exemple de la graphie "avocat" pris en tant que "fruit" comparativement avec la profession d'avocat.
- monosèmie. Cela signifie que le mot ne possède qu'un seul sens.
- autre. Cela signifie qu'il n'y a pas de description de l'unité de sens dans le dictionnaire.

L'unité de sens peut être qualifiée comme appartenant à un ou plusieurs domaines techniques. La liste des domaines n'est pas fixée dans la présente spécification de format.

L'unité de sens peut porter un attribut pour désigner le synset Wordnet.

Enfin, la définition de l'unité de sens est exprimée par un texte.

3.1.2 modèle de donnée

Nous aurons donc un élément XML nommé "U" qui sera enchâssé dans un mot. Cet élément est optionnel, ce qui correspond au cas de figure "autre" du paragraphe précédent.

L'élément comporte trois attributs optionnels:

- l'attribut monovalué "p" pour la polysémie qui peut prendre l'une des deux valeurs "poly" ou "mono".
- l'attribut optionnel multivalué "d" qui comporte la liste des domaines.

- la référence à un synset WordNet via l'attribut "wn".

Nous aurons par exemple à partir du mot "avocat" qui est polysémique (ce qui explique la valeur "poly") et dont le sens retenu est celui du fruit (ce qui explique le domaine "AGRICULTURE") les balises XML suivantes :

```
<W id="E0W3" tokens="E0W3T0" pos="commonNoun" lemma="avocat" form="avocat" mstag="nS gM">
    <U p="poly" d="AGRICULTURE" wn="w35">pear-shaped tropical fruit with green or blackish skin and rich yellowish pulp enclosing a single large seed</U>
</W>
```

3.2 La coréférence

3.2.1 aspect linguistique

Il s'agit de relier ensemble des éléments dont on fait l'hypothèse qu'ils désignent la même réalité. La structure s'appelle la chaîne de coréférence.

Les phénomènes linguistiques concernés sont les suivants:

* les références pronominales,
* les variantes d'entités nommées,
* les variantes fondées sur des noms communs,
* l'apposition,
* la métaphore,
* le lien avec une entrée d'une base de connaissance.

Une référence pronominale est un lien d'un pronom vers un antécédent. Dans la phrase: "Robert arrive puis il repart", le pronom "il" fait référence à "Robert". On appelle ce mécanisme, l'anaphore qui est une reprise par un pronom d'un antécédent, donc de la droite vers la gauche. On peut rencontrer (assez rarement) une cataphore qui est une référence à une entité nommée à venir, que l'on appelle alors conséquent. C'est donc une référence de la gauche vers la droite. Ce sera par exemple: "Tu la vois Mireille ce week-end ?" dans lequel "la" est une référence au conséquent "Mireille". Notons que certains pronoms peuvent ne pas avoir de référent, c'est le cas des pronoms impersonnels, comme dans "il pleut" ou tout simplement des pronoms dont l'auteur n'a pas jugé bon de mentionner le référent, en se fondant sur le fait que le contexte d'interprétation par un lecteur humain permettra de le déduire.

Le traitement des variantes d'entités nommées consiste à regrouper dans un même chaînage des entités nommées qui réfèrent à la même réalité (ou supposée telle) mais qui sont soient identiques, soient exprimées de manière légèrement différente. Ainsi, par exemple, les différentes occurrences de "Jacques Chirac" feront partie de la même chaîne de coréférence. Un autre exemple est la présence de "Jacques Chirac" et "J. Chirac" qui seront regroupés dans la même chaîne de coréférence car ce sont des variantes.

Le traitement des variantes fondées sur des noms communs consiste à regrouper des variantes qui réfèrent à la même réalité mais qui sont soient identiques, soient exprimées de manière légèrement différente. Par exemple "Le nouveau secrétaire ... Le secrétaire ...". Il ne s'agit pas traiter tous les noms communs mais seulement une petite liste, peut-être réduite à la liste des fonctions.

L'apposition consiste à associer une EN avec un groupe nominal fondé sur un nom commun, avec bien souvent la mention d'une fonction. Ce sera par exemple: "Yves Tavernier, le député du Sud-Ouest ...". L'apposition est un phénomène très informatif qui est relativement fréquent dans le style

journalistique. Ainsi, dans l'exemple, sa détection automatique permettra d'inférer sans aucune base de connaissance qu'Yves Tavernier est un député au moment de l'énonciation.

La métaphore consiste à employer un mot imagé pour un autre au travers d'une analogie. Par exemple: "Bruges, la Venise du nord". On se limitera aux métaphores enregistrées dans une base de connaissance sans se préoccuper des algorithmes qui seraient capables de les découvrir dynamiquement.

Ces éléments même s'ils sont de différente nature peuvent se combiner dans la même chaîne de coréférence. On pourra avoir par exemple deux entités nommées et un pronom dans la même chaîne: "Jacques Chirac ... J. Chirac ... il déclare alors ...".

L'unité de traitement n'est pas la phrase mais le document. Car même si dans certaines circonstances, l'unité de traitement peut se limiter à la phrase comme dans "Robert accélère puis il tourne à droite", les coréférences dépassent bien souvent la portée de la phrase comme dans "Robert accélère. Puis il tourne à droite". Pour avoir un dispositif homogène quelles que soient les circonstances, l'unité de traitement sera donc pour nous le document.

3.2.2 modèle de donnée

La représentation des coréférence est fondée sur la notion de Coref. Un Coref est une référence vers soit une entité nommée, soit un pronom. Le type est donc nécessairement hétérogène.

La chaine de coréférence est un élément comprenant:

* la liste des Coref de la chaîne. Cette liste conserve l'ordre d'apparition des éléments dans le document d'entrée. La liste comporte au moins deux Coref sans aucune limite supérieure de leur nombre.

Un élément ne peut pas appartenir à deux chaînes de coréférence différentes.

Quand une entité nommée n'apparait qu'une seule fois, il n'y a pas construction d'une chaîne de coréférence.

Prenons l'exemple de ces trois phrases: « Robert entre. Il dit ‘bonjour’. Max dort»,.

Avec la convention que seules les balises nécessaires à la compréhension de l'exemple sont présentes (voir plus loin, pour un exemple plus complet), les données XML relatives à la coréférence seront comme suit, :

```
<W id= “w1“ form=“Robert“/>                    <!--le premier mot-->
<NE id=”e1” type=”individual” lst=”w1”/>       <!--l'entité nommée qui fait référence au premier mot-->
<W id=“w4“ form=“il“/>                         <!--le deuxième mot-->
<CorefChain id=”c1”>                           <!--La chaîne de coréférence pour relier Robert à il-->
        <Coref ref=“e1“/>
        <Coref ref=“w4“/>
</CorefChain>
```

Maintenant, imaginons qu'une entité nommée soit reconnue comme étant une entrée d'une base de connaissance, il faut préciser l'identifiant et le nom de la base de connaissance. Sur l'exemple: "Staline a poursuivi la stratégie ...", nous aurons la structure suivante:

```
<W id="w50" form="Staline"/>
<NE id="e50" type="individual" lst="w50"/>
<CorefChain id="c50">
        <Coref ref="e50"/>
        <Coref ref="X66" base="histoire.kb"/>
</CorefChain>
```

Mais le calcul des coréférences peut très bien ne pas être parfait et s'il existe des possibilités pour affiner le calcul ou demander à l'utilisateur de choisir, il est nécessaire de représenter les hypothèses plausibles au regard des contraintes syntaxiques. L'élément HypoCorefChain est destiné à cet usage. Il est associé à un poids numérique de confiance qui est une valeur à prendre entre zéro et 100. L'élément CorefChain est en fait le meilleur des choix parmi les différentes hypothèses. L'élément CorefChain est le seul élément qui sera testé par les tests de non régression. En supposant que deux références possibles depuis un mot w4 soient possibles (soit vers e1 ou soit vers e2), nous aurons la structure suivante:

```
<HypoCorefChain id="h1" trust="4">
        <Coref ref="e1"/>
        <Coref ref="w4"/>
</HypoCorefChain>
<HypoCorefChain id="h2" trust="50">
        <Coref ref="e2"/>
        <Coref ref="w4"/>
</HypoCorefChain>
```

3.3 Les références au monde externe

Il s'agit de lister de manière unique les références au monde extérieur au document.

C'est la liste :

1) des entités nommées qui appartiennent à des chaînes de coréférence distinctes,

2) et des pronoms dont la référence n'a pas été résolue.

Pour le premier point, ces informations seront très importantes si l'on désire alimenter automatiquement (ou semi-automatiquement) une base de connaissances.

Quand une chaîne de référence est mentionnée, le WordRef porte une référence particulière d'un des éléments de la liste. C'est l'élément qui est choisi si un affichage doit être effectué.

Les données XML seront comme suit:

```
<W id="w1" form="Robert"/>
<W id="w10" form="Max"/>
<NE id="e1" type="individual" lst="w1"/>
<NE id="e2" type="individual" lst="w10"/>
        <!--Fait référence à la chaîne de coréférence du paragraphe précédent-->
<WorldRef id="wr1" lib="Robert" type="individual" ref="e1" chain="c1"/>
<WorldRef id="wr2" lib="Max" type="individual" ref="e2"/>
```

3.4 Les citations (ou prises de parole)

3.4.1 aspect linguistique

Par convention terminologique, on définit le terme "citation" comme un élément structuré comportant un fragment de texte rapporté. C'est le seul élément obligatoire. De manière optionnelle, la citation comporte un locuteur (ou source[4]) et un relateur [8],[9],[10].

Dans le texte d'entrée, la citation comporte la notion de relateur. Le relateur est un groupe qui peut être soit un noyau verbal NV (il déclare), un GP (D'après X) ou un GN (La réponse de X). Il peut également être absent. Quand un relateur est un verbe, il est bien souvent nécessaire de savoir si le verbe est au conditionnel ou à l'indicatif.

Pour être un peu plus concret, l'algorithme de détection permet de calculer la réponse aux

4 On évitera d'utiliser le terme "source" pour éviter l'ambiguïté avec la notion de source (ou origine) de document.

requêtes suivantes:

1) déterminer ce qui a été dit que le locuteur (ou le relateur) soit ou non connu,

2) étant donné un locuteur (et/ou un relateur), déterminer ce qu'il a dit,

3) étant donnés des fragments de ce qui a été dit, déterminer quel est le locuteur (et/ou le relateur).

3.4.2 modèle de donnée

Une citation est définie par le triplet : locuteur / relateur / discours.

Le locuteur (ou source) est une référence vers un WordRef. Il peut être absent. On peut en avoir plusieurs comme dans l'exemple : « Jean et Marie s'écrient en cœur : 'touché' ». Notons que les représentations des locuteurs ne peuvent pas être des mots car les locuteurs peuvent être des entités nommées, avec par exemple: « Jean Dupont et Marie Durand s'écrient en coeur : 'touché' ».

Le relateur est une référence vers un mot. Il peut être absent.

Le discours peut être encadré par des guillemets en totalité, partiellement (discours à composantes, îlots textuels) ou en dehors de tout guillemet. Un discours est défini par une liste de fragments. Chaque fragment un empan qui est un couple de mot de début et mot de fin.

Les citations dans le document conservent leur ordre d'apparition dans la mesure où les fragments sont ordonnés dans la liste.

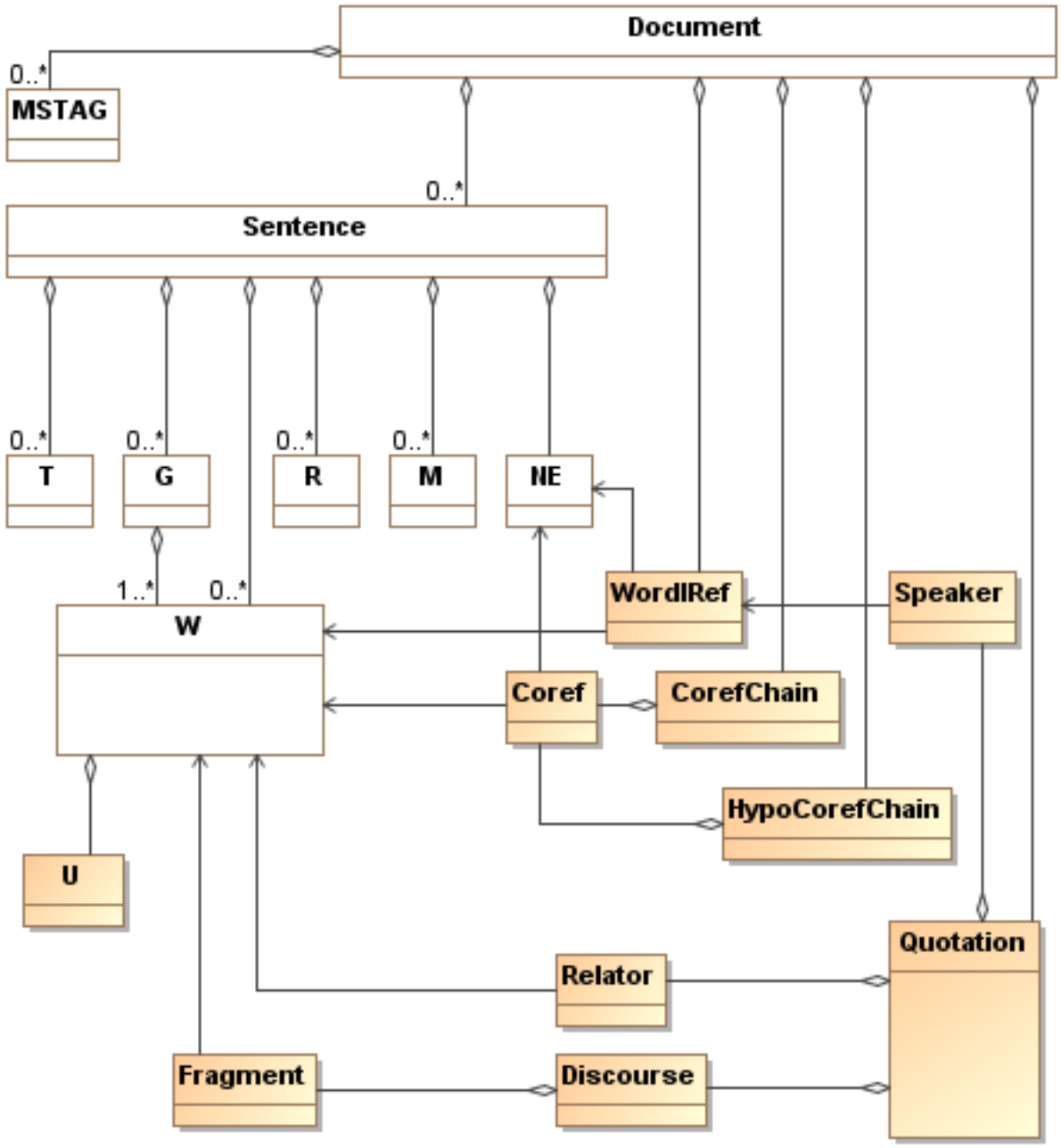
Sur "Robert entre. il dit "bonjour".", les balises XML de la citation seront comme suit:

```
<W id="w4" form="il"/>
<W id="w5" form= "dit"/>
<W id="w6" form="' "/>
<W id="w7" form="bonjour"/>
<W id="w8" form="' "/>
<W id="w9" form="."/>
<NE id="e1" type="individual" lst="w1"/>
<WorldRef id="wr1" ref="e1" chain="c1"/>
<Quotation id="q1">
        <Speaker refs="wr1"/>
        <Relator ref="w5"/>
        <Discourse>
                <Fragment id="f1" start="w6" end="w8"/>
        </Discourse>
</Quotation>
```

4 Le modèle de donnée Passage++

Les chaînes de coréférence, les WordlRef et les citations apparaissent au niveau du document. En revanche, l'unité de sens est locale à un mot (l'élément W).

Pour faciliter la lecture, nous adoptons la convention que les classes du modèle Passage sont en blanc et les classes qui sont ajoutées par le modèle Passage++ sont colorées. Ce qui nous donne le diagramme UML suivant:



5 Dépendance entre les données

Les structures Passage++ reposent sur les données Passage, donc il faut que les données Passage soient calculées avant les structures Passage++.

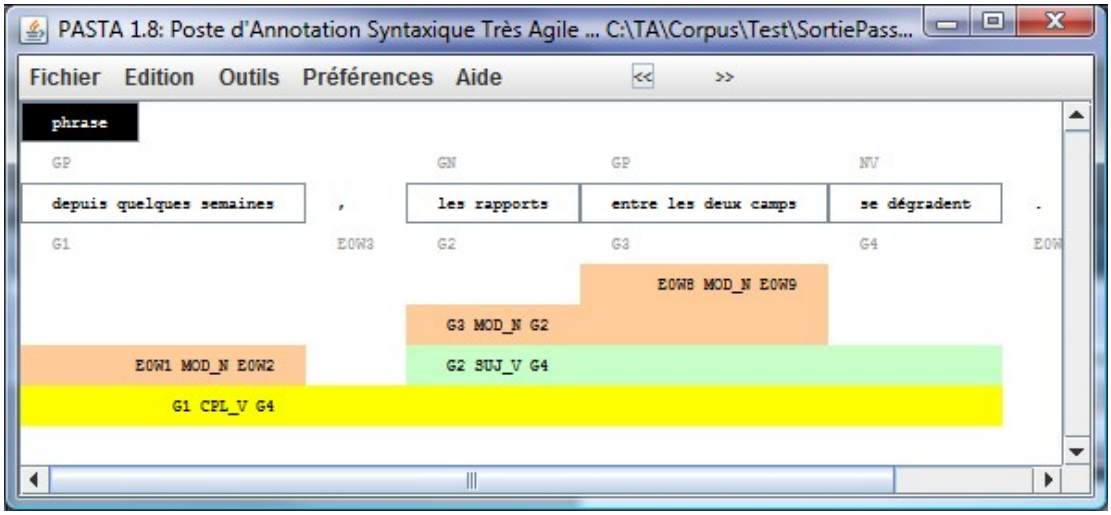
Au sein des structures Passage++, il existe des dépendances. La coréférence doit être calculée en premier, puis les WordRef puis les citations.

6 Exemples

6.1 Exemple#1 de Passage

Sur la phrase suivante, située en début de fichier:

« Depuis quelques semaines, les rapports entre les deux camps se dégradent. »



Nous aurons le balisage XML suivant:

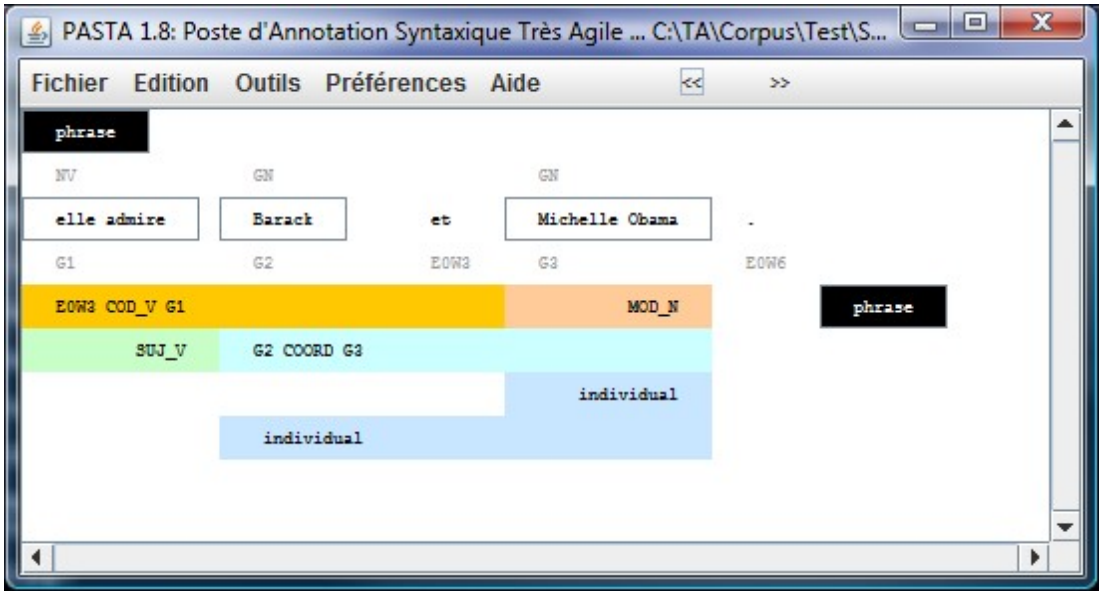
```
<?xml version="1.0" encoding="UTF-8"?>
<Document dtdVersion="2.1" file="test.txt">
<MSTAG id="nS"><fs><f name="grammaticalNumber"><symbol value="singular"/></f></fs></MSTAG>
<MSTAG id="nP"><fs><f name="grammaticalNumber"><symbol value="plural"/></f></fs></MSTAG>
<MSTAG id="nX"><fs><f name="grammaticalNumber"><vAlt><symbol value="singular"/><symbol
value="plural"/></vAlt></f></fs></MSTAG>
<MSTAG id="gM"><fs><f name="grammaticalGender"><symbol value="masculine"/></f></fs></MSTAG>
<MSTAG id="gF"><fs><f name="grammaticalGender"><symbol value="feminine"/></f></fs></MSTAG>
<MSTAG id="gE"><fs><f name="grammaticalGender"><vAlt><symbol value="masculine"/><symbol
value="feminine"/></vAlt></f></fs></MSTAG>
<MSTAG id="mI"><fs><f name="verbFormMood"><symbol value="infinitive"/></f></fs></MSTAG>
<MSTAG id="mP"><fs><f name="verbFormMood"><symbol value="participle"/></f></fs></MSTAG>
<MSTAG id="mC"><fs><f name="verbFormMood"><symbol value="conjugated"/></f></fs></MSTAG>
<MSTAG id="mU"><fs><f name="verbFormMood"><symbol value="unknown"/></f></fs></MSTAG>
<Sentence id="E0">
   <T id="E0W0T0" start="0" end="6">depuis</T>
   <T id="E0W1T0" start="7" end="15">quelques</T>
   <T id="E0W2T0" start="16" end="24">semaines</T>
 <G id="E0G1" type="GP">
   <W id="E0W0" tokens="E0W0T0" pos="preposition" lemma="depuis" form="depuis"/>
   <W id="E0W1" tokens="E0W1T0" pos="indefiniteDeterminer" lemma="quelque" form="quelques"/>
   <W id="E0W2" tokens="E0W2T0" pos="commonNoun" lemma="semaine" form="semaines" mstag="nP gF"/>
 </G>
   <T id="E0W3T0" start="24" end="25">,</T>
   <W id="E0W3" tokens="E0W3T0" pos="secondaryPunctuation" lemma="," form=","/>
   <T id="E0W4T0" start="26" end="29">les</T>
   <T id="E0W5T0" start="30" end="38">rapports</T>
 <G id="E0G2" type="GN">
   <W id="E0W4" tokens="E0W4T0" pos="definiteArticle" lemma="le" form="les"/>
   <W id="E0W5" tokens="E0W5T0" pos="commonNoun" lemma="rapport" form="rapports" mstag="nP gM"/>
 </G>
   <T id="E0W6T0" start="39" end="44">entre</T>
   <T id="E0W7T0" start="45" end="48">les</T>
   <T id="E0W8T0" start="49" end="53">deux</T>
   <T id="E0W9T0" start="54" end="59">camps</T>
 <G id="E0G3" type="GP">
```

```
  <W id="E0W6" tokens="E0W6T0" pos="preposition" lemma="entre" form="entre"/>
  <W id="E0W7" tokens="E0W7T0" pos="definiteArticle" lemma="le" form="les"/>
  <W id="E0W8" tokens="E0W8T0" pos="numeral" lemma="deux" form="deux"/>
  <W id="E0W9" tokens="E0W9T0" pos="commonNoun" lemma="camp" form="camps" mstag="nP gM"/>
</G>
  <T id="E0W10T0" start="60" end="62">se</T>
  <T id="E0W11T0" start="63" end="72">dégradent</T>
<G id="E0G4" type="NV">
  <W id="E0W10" tokens="E0W10T0" pos="personalPronoun" lemma="se" form="se"/>
  <W id="E0W11" tokens="E0W11T0" pos="verb" lemma="dégrader" form="dégradent" mstag="mC"/>
</G>
  <T id="E0W12T0" start="72" end="73">.</T>
  <W id="E0W12" tokens="E0W12T0" pos="mainPunctuation" lemma="." form="."/>
<R id="E0R0" type="SUJ-V">
  <sujet ref="E0G2"/>
  <verbe ref="E0G4"/>
</R>
<R id="E0R1" type="CPL-V">
  <complement ref="E0G1"/>
  <verbe ref="E0G4"/>
</R>
<R id="E0R2" type="MOD-N">
  <modifieur ref="E0W1"/>
  <nom ref="E0W2"/>
</R>
<R id="E0R3" type="MOD-N">
  <modifieur ref="E0G3"/>
  <nom ref="E0G2"/>
</R>
<R id="E0R4" type="MOD-N">
  <modifieur ref="E0W8"/>
  <nom ref="E0W9"/>
</R>
</Sentence>
```

6.2 Exemple#2 de Passage

Sur la phrase suivante, située en début de fichier:

« Elle admire Barack et Michelle Obama. »



Nous aurons le balisage XML suivant, avec deux entités nommées « Barack Obama » et :

« Michelle Obama » en fin de fichier.

```
<?xml version="1.0" encoding="UTF-8"?>
<Document dtdVersion="2.1" file="test.txt">
<MSTAG id="nS"><fs><f name="grammaticalNumber"><symbol value="singular"/></f></fs></MSTAG>
<MSTAG id="nP"><fs><f name="grammaticalNumber"><symbol value="plural"/></f></fs></MSTAG>
<MSTAG id="nX"><fs><f name="grammaticalNumber"><vAlt><symbol value="singular"/><symbol
value="plural"/></vAlt></f></fs></MSTAG>
<MSTAG id="gM"><fs><f name="grammaticalGender"><symbol value="masculine"/></f></fs></MSTAG>
<MSTAG id="gF"><fs><f name="grammaticalGender"><symbol value="feminine"/></f></fs></MSTAG>
<MSTAG id="gE"><fs><f name="grammaticalGender"><vAlt><symbol value="masculine"/><symbol
value="feminine"/></vAlt></f></fs></MSTAG>
<MSTAG id="mI"><fs><f name="verbFormMood"><symbol value="infinitive"/></f></fs></MSTAG>
<MSTAG id="mP"><fs><f name="verbFormMood"><symbol value="participle"/></f></fs></MSTAG>
<MSTAG id="mC"><fs><f name="verbFormMood"><symbol value="conjugated"/></f></fs></MSTAG>
<MSTAG id="mU"><fs><f name="verbFormMood"><symbol value="unknown"/></f></fs></MSTAG>
<Sentence id="E0">
   <T id="E0W0T0" start="0" end="4">elle</T>
   <T id="E0W1T0" start="5" end="11">admire</T>
 <G id="E0G1" type="NV">
   <W id="E0W0" tokens="E0W0T0" pos="personalPronoun" lemma="il" form="elle" mstag="nS gM"/>
   <W id="E0W1" tokens="E0W1T0" pos="verb" lemma="admirer" form="admire" mstag="mC"/>
 </G>
   <T id="E0W2T0" start="12" end="18">Barack</T>
 <G id="E0G2" type="GN">
   <W id="E0W2" tokens="E0W2T0" pos="properNoun" lemma="Barack" form="Barack" mstag="gM"/>
 </G>
   <T id="E0W3T0" start="19" end="21">et</T>
   <W id="E0W3" tokens="E0W3T0" pos="coordinatingConjunction" lemma="et" form="et"/>
   <T id="E0W4T0" start="22" end="30">Michelle</T>
   <T id="E0W5T0" start="31" end="36">Obama</T>
 <G id="E0G3" type="GN">
   <W id="E0W4" tokens="E0W4T0" pos="properNoun" lemma="Michelle" form="Michelle" mstag="gF"/>
   <W id="E0W5" tokens="E0W5T0" pos="properNoun" lemma="Obama" form="Obama"/>
 </G>
   <T id="E0W6T0" start="36" end="37">.</T>
   <W id="E0W6" tokens="E0W6T0" pos="mainPunctuation" lemma="." form="."/>
 <R id="E0R0" type="SUJ-V">
   <sujet ref="E0W0"/>
   <verbe ref="E0W1"/>
 </R>
 <R id="E0R1" type="COD-V">
   <cod ref="E0W3"/>
   <verbe ref="E0G1"/>
 </R>
 <R id="E0R2" type="MOD-N">
   <modifieur ref="E0W4"/>
   <nom ref="E0W5"/>
 </R>
 <R id="E0R3" type="COORD">
   <coordonnant ref="E0W3"/>
   <coord-g ref="E0G2"/>
   <coord-d ref="E0G3"/>
 </R>
 <NE id="E0E0" type="individual" lst="E0W2 E0W5"/>
 <NE id="E0E1" type="individual" lst="E0W4 E0W5"/>
</Sentence>
</Document>
```

6.3 Exemple de Passage++

Sur l'exemple : « Robert entre. Il dit 'bonjour'. Max dort», nous aurons :

```
<Document dtdVersion="2.1" file="examplePPP.txt">
< !—début de section avec balises partielles->
<Sentence>
```

```
        <W id="w1" form="Robert"/>
        <W id="w2" form="entre"/>
        <W id="w3" form="."/>
</Sentence>
<Sentence>
        <W id="w4" form="il"/>
        <W id="w5" form= "dit"/>
        <W id="w6" form="' "/>
        <W id="w7" form="bonjour"/>
        <W id="w8" form="' "/>
        <W id="w9" form="."/>
<NE id="e1" type="individual" lst="w1"/>
</Sentence>
<Sentence>
        <W id="w10" form="Max"/>
        <W id="w11" form="dort"/>
        <W id="w12" form="."/>
<NE id="e2" type="individual" lst="w10"/>
</Sentence>
<!—fin de section avec balises partielles->
<CorefChain id="c1">
<Coref ref="e1"/>
        <Coref ref="w4"/>
</CorefChain>
<WorldRef id="wr1" lib="Robert" type="individual" ref="e1" chain="c1"/>
<WorldRef id="wr2" lib="Max" type="individual" ref="e2"/>
<Quotation id="q1">
        <Speaker refs="wr1"/>
        <Relator ref="w5"/>
        <Discourse>
                <Fragment id="f1" start="w6" end="w8"/>
        </Discourse>
</Quotation>
</Document>
```

7 Segmentation en tokens

L'algorithme est très simple et se fonde sur les propriétés des caractères telles que définies par Unicode dans la table General_Category. Rappelons que les caractères sont définis par une hiérarchie à deux niveaux des propriétés. Le premier niveau est soit l'une des 7 valeurs suivantes : Other, Letter, Mark, Number, Punctuation, Symbol et Separator. Notons de plus que le deuxième niveau est par exemple Lowercase_Letter qui est une valeur plus spécifique de Letter, mais nous n'avons pas besoin de ce deuxième niveau.

Pour ce qui nous concerne, nous distinguons :

- les caractères informatifs qui vont constituer l'essentiel du contenu textuel. Ce sont les caractères de type Letter et Number.
- les caractères séparateurs. Ce sont les caractères de type Separator qui comprennent par exemple, l'espace ou le saut de ligne.
- les caractères autonomes. Ce sont les caractères de type Other, Mark, Punctuation, Symbol. La virgule par exemple sera de type autonome.

La segmentation en tokens est régie par les principes suivants:

Principe#1

Un token est défini de manière exclusive :

- soit comme une suite contiguë de caractères de type Letter ou Number, à l'exclusion de tout autre caractère. Par exemple, les suites "la", "34", "X56" formeront chacun un token.
- soit comme un caractère autonome. Par exemple, la virgule formera un token à elle toute

seule, et ceci quels que soient les caractères qui l'entourent.

Principe#2

Un caractère séparateur ne figure pas dans l'annotation. Par exemple, le caractère espace même s'il provoque une segmentation, ne figure pas dans le résultat de l'annotation.

Notons que la conséquence de ces deux principes est qu'un token ne peut pas être vide.

8 DTD XML du format Passage

```
<?xml version='1.0' encoding="UTF-8"?>
<!ELEMENT Document (MSTAG*, Sentence*)>
<!ATTLIST Document      dtdVersion      CDATA #FIXED "2.1"
                        file            CDATA #IMPLIED>
<!-- ================================-->
<!ELEMENT MSTAG (fs)>
<!ATTLIST MSTAG         id              ID #REQUIRED>
<!ELEMENT fs (f+)>
<!ELEMENT f (symbol | vAlt)+>
<!ATTLIST f             name            CDATA #REQUIRED>
<!ELEMENT symbol EMPTY>
<!ATTLIST symbol        value           CDATA #REQUIRED>
<!ELEMENT vAlt (symbol+)>
<!-- ================================-->
<!ELEMENT Sentence ( (T|G|W)+, R*, M* ,NE*,CorefChain*,HypoCorefChain*,WorldRef*,Quotation*)>
<!ATTLIST Sentence      id              ID #REQUIRED
                        trust           CDATA #IMPLIED>
<!-- ================================-->
<!ELEMENT T (#PCDATA)>
<!ATTLIST T             id              ID #REQUIRED
                        start           CDATA #REQUIRED
                        end             CDATA #REQUIRED>
<!-- ================================-->
<!ELEMENT G (W+)>
<!ATTLIST G             id              ID #REQUIRED
                        type            (NV|GN|GP|GA|GR|PV)  #REQUIRED
                        mstag           IDREFS #IMPLIED>
<!-- ================================-->
<!ELEMENT W (U?)>
<!ATTLIST W             id              ID #REQUIRED
                        tokens          IDREFS #REQUIRED
                        pos             (adverb | commonNoun |
                                        coordinatingConjunction | definiteArticle |
                                        demonstrativeDeterminer |    demonstrativePronoun |
                                        exclamativeDeterminer |
                                        foreignText | formula | impersonalPronoun |
                                        indefiniteDeterminer |
                                        indefinitePronoun | interjection |
                                        interrogativePronoun |
                                        letter | mainPunctuation |
                                        negativeParticle | numeral | ordinalAdjective |
                                        partitiveArticle |
                                        personalPronoun | possessiveDeterminer |
                                        possessivePronoun | preposition | properNoun |
                                        qualifierAdjective |
                                        relativePronoun | residual | reflexivePronoun |
                                        secondaryPunctuation | subordinatingConjunction |
                                        verb) #IMPLIED
                        lemma           CDATA #IMPLIED
                        form            CDATA #IMPLIED
                        mstag           IDREFS #IMPLIED
                        head          (true | false) #IMPLIED>
<!-- ================================-->
<!ELEMENT R (adjectif | adverbe | appose | attribut | auxiliaire | cod | complement | complementeur |coord-d | coord-g |
coordonnant | modifieur | nom | premier | preposition | s-o | suivant | sujet | verbe )+ >
```

```
<!ATTLIST R                 id                  ID #REQUIRED
                            type                (APPOS|ATB-SO|AUX-V|COMP|COD-V|COORD|
CPL-V|JUXT|MOD-A|MOD-N|MOD-P|MOD-R|MOD-V|SUJ-V) #REQUIRED >

<!ELEMENT adjectif EMPTY >
<!ATTLIST adjectif          ref                 IDREF #REQUIRED>

<!ELEMENT adverbe EMPTY >
<!ATTLIST adverbe           ref                 IDREF #REQUIRED>

<!ELEMENT appose EMPTY >
<!ATTLIST appose            ref                 IDREF #REQUIRED>

<!ELEMENT attribut EMPTY >
<!ATTLIST attribut          ref                 IDREF #REQUIRED>

<!ELEMENT auxiliaire EMPTY >
<!ATTLIST auxiliaire        ref                 IDREF #REQUIRED>

<!ELEMENT cod EMPTY >
<!ATTLIST cod               ref                 IDREF #REQUIRED>

<!ELEMENT complement EMPTY >
<!ATTLIST complement        ref                 IDREF #REQUIRED>

<!ELEMENT complementeur EMPTY >
<!ATTLIST complementeur ref         IDREF #REQUIRED>

<!ELEMENT coord-d EMPTY >
<!ATTLIST  coord-d          ref                 IDREF #REQUIRED>

<!ELEMENT coord-g EMPTY >
<!ATTLIST  coord-g          ref                 IDREF #IMPLIED>

<!ELEMENT coordonnant EMPTY >
<!ATTLIST  coordonnant      ref                 IDREF #REQUIRED>

<!ELEMENT modifieur EMPTY >
<!ATTLIST  modifieur        ref                 IDREF #REQUIRED>

<!ELEMENT nom EMPTY >
<!ATTLIST  nom              ref                 IDREF #REQUIRED>

<!ELEMENT premier EMPTY >
<!ATTLIST  premier          ref                 IDREF #REQUIRED>

<!ELEMENT preposition EMPTY >
<!ATTLIST  preposition      ref                 IDREF #REQUIRED>

<!ELEMENT s-o EMPTY >
<!ATTLIST s-o      valeur ( sujet | objet | ind ) #REQUIRED >

<!ELEMENT suivant EMPTY >
<!ATTLIST  suivant          ref                 IDREF #REQUIRED>

<!ELEMENT sujet EMPTY >
<!ATTLIST  sujet            ref                 IDREF #REQUIRED>

<!ELEMENT verbe EMPTY >
<!ATTLIST  verbe            ref                 IDREF #REQUIRED>
<!-- ================================-->
<!ELEMENT M (#PCDATA)>
<!ATTLIST M                 id                  ID #REQUIRED
                            start               CDATA #IMPLIED
                            end                 CDATA #IMPLIED
                            objs                IDREFS #IMPLIED>
<!-- ================================-->
```

```
<!ELEMENT NE EMPTY>
<!ATTLIST NE              id          ID #REQUIRED
                          type        (individual | organization | location | dateTime |
                                      URLetc | measure | mark | event | work | unnamed) #REQUIRED
                          act         (athlete,golfer,rugbyPlayer,soccerPlayer,swimmer,tennisPlayer,
                                      politician,
                                      journalist) #IMPLIED
                          lst         IDREFS #REQUIRED
                          mstag       IDREFS #IMPLIED>
<!-- ================================-->
<!ELEMENT U (#PCDATA)>
<!ATTLIST U               p           (poly,mono) #REQUIRED
                          d           CDATA #IMPLIED
                          wn          CDATA #IMPLIED>
<!-- ================================-->
<!ELEMENT CorefChain (Coref+)>
<!ATTLIST CorefChain      id          ID #REQUIRED>

<!ELEMENT HypoCorefChain (Coref+)>
<!ATTLIST HypoCorefChain id           ID #REQUIRED
                          trust       CDATA #IMPLIED>

<!ELEMENT Coref EMPTY>
<!ATTLIST Coref           ref         IDREF #REQUIRED
                          base        CDATA #IMPLIED>
<!-- ================================-->
<!ELEMENT WorldRef EMPTY>
<!ATTLIST WorldRef        id          ID #REQUIRED
                          lib         CDATA #REQUIRED
                          type        (individual | organization | location | dateTime |
                                      URLetc | measure | mark | event | work | unnamed)
                                          #REQUIRED
                          ref         IDREF #REQUIRED
                          chain       IDREF #IMPLIED>
<!-- ================================-->
<!ELEMENT QUOTATION (Speaker?,Relator?,Discourse)>
<!ATTLIST QUOTATION       id          ID #REQUIRED>
<!ELEMENT Speaker EMPTY>
<!ATTLIST Speaker         refs        IDREFS #REQUIRED>
<!ELEMENT Relator EMPTY>
<!ATTLIST Relator         ref         IDREF #REQUIRED>
<!ELEMENT Discourse (Fragment+)>
<!ELEMENT Fragment EMPTY>
<!ATTLIST Fragment        id          ID #REQUIRED
                          start       CDATA #REQUIRED
                          end         CDATA #REQUIRED>
```

9 Références

[1] Rumbaugh J., Jacobson I., Booch G. 2005 The Unified Modeling language reference manual, 2nd Edition, Addison Wesley, Boston MA

[2] Registre de catégories de données http://www.isocat.org

[3] MAF, voir http://lirics.loria.fr/doc_pub/maf.pdf

[4] SynAF, voir http://lirics.loria.fr/doc_pub/N421_SynAF_CD_ISO_24615.pdf

[5] Desgraupes B. 2005 Passeport pour Unicode, Vuibert, Paris

[6] Harold & Means 2004 XML in a nutshell 3rd Edition, O'Reilly, Sebastopol, CA

[7] Language Resources Management - Feature structures - Part 1: feature structure representation ISO 24610-1

[8] Rosier Laurence 2008 Le discours rapporté en français, Orphrys

[9] Groupe de travail sur le discours rapporté : “Ci-Dit”: http://www.ulb.ac.be/philo/serlifra/ci-dit/index.html

[10] Cours sur le discours rapporté : http://www.etudes-litteraires.com/discours-rapporte.php