ATOLL - INRIA Rocquencourt

http://atoll.inria.fr

Appel à contribution sur la morpho-syntaxe

Éric de la Clergerie

Eric.De_La_Clergerie@inria.fr

Réunion RNIL

AFNOR – Lundi 7 Avril 2003

Objectifs de la contribution

- Préparer un appel à contribution concernant les annotations morpho-syntaxiques.
- Susciter une discussion, en proposant un point de départ.

Quelques grandes orientations

- Utilisation de XML comme format de représentation

- Utilisation de XML Schema comme langage de spécification, plutôt que les DTD

  $\Rightarrow$ plus grande modularité

  $\Rightarrow$ possibilités d'extensions des schémas

  $\Rightarrow$ famille de schémas **reliés** d'annotations, avec possibilités de **conversions** par XSLT

- Ré-utilisation d'autres normes (ou propositions),
  comme les Structures de traits, TEI, EAGLES, MULTEXT

- Utilisation de repertoire de catégories,
  i.e. enregistrement de termes linguistiques à utiliser de préférence

- Format compatible avec une annotation manuelle ou automatique.

Segmentation : token

Motivation : Permettre l'ancrage des annotations sur des segments du document.

Peut également servir comme sortie d'un **segmenteur**

2 formes :

- Inclusion de contenu lexical

```
<token id="1" form="pomme">pomme</token>
<token id="2" form="de">de</token>
<token id="3" form="terre"> terre </token>
```

Note : le contenu d'un token peut être complexe.

- Réference (bas niveau) à un segment, par exemple dans un document multimédia.

```
<token id="1" form="pomme" span="20 26"/>
```

Notes :

- le contenu de l'attribut span doit reposer sur des normes pour le multimédia.
- l'attribut span remplacés par des attributs from et to

segmentation : token (suite)

La segmentation peut poser quelques problèmes.

Contraction

```
<token form="etc" id="0">etc.</token>
<token form="DOT" id="1"/>

<token form="à" id="0">auquel</token>
<token form="lequel" id="1"/>
```

Concatenation

```
<token form="Geburtstags" id="0" type="glued">Geburtstags</token>
<token form="Geschenk" id="1" type="glued">geschenk</token>
<token form="Papier" id="2">papier</token>

<token form="aujourd" id="0" type="??">aujourd'</token>
<token form="hui" id="1">hui</token>
```

Note : quels types possibles ?

Segmentation : s

Utilité mais non nécessité d'une segmentation en phrases

```
<s>
    <token id="0">il</token>
    <token id="1">mange</token>
    <token id="2">.</token>
</s>
```

Les autres segmentations de document (paragraphes, . . .) gérées par d'autres normes.

Annotations morpho-syntaxique : w

Motivation : Donner des informations morpho-syntaxiques pour un mot.

Les informations

- portent sur un ou plusieurs tokens
- s'appuient sur une structure de traits (forme pleine)

```
<token id="1" form="pomme">pomme</token>
<token id="2" form="de">de</token>
<token id="3" form="terre"> terre </token>
<w entry="pomme␣de␣terre" tokens="1␣2␣3">
    <fs>
        <f name="pos"><sym value="n"/></f>
        <f name="num"><sym value="sing"/></f>
        <f name="gen"><sym value="fem"/></f>
    </fs>
</w>
```

Annotations : formes compactes

Les bibliothèques de valeurs, traits et structures permettent des formes compactes :

```
<fvLib>
    <sym id="n" value="n"/><sym id="sing" value="sing"/><sym id="fem" value="fem"/>
</fvLib>
<fLib>
  <f id="pos@n" name="pos" fVal="n"/>
  <f id="num@sing" name="num" fVal="sing"/>
  <f id="gen@fem" name="gen" fVal="fem"/>
</fLib>
<w entry="pomme␣de␣terre" tokens="1␣2␣3">
      <fs feats ="pos@n␣num@sing␣gen@fem"/>
</w>
<w entry="pomme␣de␣terre" tokens="1␣2␣3" tag="pos@n␣num@sing␣gen@fem"/>
```

Notes :

- attribut tag renommé en feats ?
- attribut entry référence vers une entrée dans un lexique ?

Annotations : traits et valeurs de traits

Problème : D'où viennent les traits et valeurs de traits ? que signifient-ils ?

Principes :

- (minimun) Déclarer les traits et valeurs (catégories) possibles par traits grâce à des bibliothèques, avec des commentaires.
- (maximum) Utiliser des traits et valeurs enregistrés dans un repertoire centralisé de catégories
- Utiliser des namespaces pour éviter des confusions sur des catégories utilisées avec des sens différents.
- (intermédiaire) lier ses catégories aux catégories enregistrées

Annotations : lier les catégories

Motivation : proposer des mécanismes de liaisons de catégories personnelles vers des catégories enregistrées permettant des conversions (avec perte d'information).

sous-typage Déclaration d'une catégorie personnelle comme sous-type d'une catégorie enregistrée.
exemple : `advneg` sous-type de `adv`

équation Equation entre une structure de trait personelle et une batie sur des catégories enregistrées.

```
<fsmap>
    <fs>          <!-- user fs -->
      <f  name="pos"><sym value="noun"/></f>
      <f  name="kind"><sym value="numeral"/></f>
    </fs>
    <!-- registered  fs -->
    <fs> <f  name="pos"><sym value="numeral"/></f></fs>
</fsmap>
```

Prises en compte des liaisons (jusqu'à un certain point) dans un schéma XML (soustypage) et dans des transformations XSL.

Articulation simple entre segmentation et annotations

Pour les cas simples sans ambiguïtés (annotation manuelle),
token et **w** suffisent :

```
<token id="0">il</token>
<w entry="mylex#cl" tokens="0" tag="pos@clitic case@nominatif pers@3 gen@masc"/>
<token id="1">mange</token>
<w entry="mylex#manger" tokens="1" tag="pos@v pers@3 tense@pres mode@ind"/>
token id="2">.</token>
<w entry="mylex#dot" tokens="2" tag="pos@punct"/>
```

Alternative

```
<w entry="mylex#manger">
    <token>mange</token>
    <fs> ... </fs>
</w>
```

Articulation complexe

Motivations :

- problèmes des ambiguïtés (annotations automatiques)
- compatibilité avec les treillis de mots (reconnaissance vocale)

$\Rightarrow$ s'appuyer sur une représentation d'automates à états finis (FSA), éventuellement pondérés.

```
<token form="pomme" id="0">Pomme</token>
<token form="de" id="1">de</token>
<token form="terre" id="2">terre</token>
<token form="cuite" id="3">cuite</token>
<state id="S0" type="initial"/>
<state id="S2"/>
<state id="S3"/>
<state id="S4"/>
<state id="S5" type="final"/>
<transition source="S0" target="S4">
    <w entry="pomme_de_terre" tokens="0_1_2" tag="..."/>
</transition>
<transition source="S4" target="S5">
    <w entry="cuite" tokens="1" tag="..."/>
</transition>
```

```
< transition   source="S0"  target ="S1">
        <w entry="pomme" tokens="0" tag=" ..." />
</ transition >
< transition   source="S1"  target ="S2">
        <w entry="de"  tokens="1"  tag=" ..." />
</ transition >
< transition   source="S3"  target ="S5">
        <w entry=" terre ␣cuite"  tokens="1"  tag=" ..." />
</ transition >
< transition   source="S3"  target ="S4">
        <w entry=" terre "  tokens="1"  tag=" ..." />
</ transition >
```

Note : Balises state pas réellement nécessaires

Articulation complexe (suite)

```
<token id="0">des</token>
<state id="S0"/>
<state id="S1"/>
<state id="S2"/>
< transition source="S0" target ="S1">
  <w entry="un" tokens="0" tag="pos&art"/>
</ transition >
< transition source="S0" target ="S1">
  <w entry="de" tokens="0" tag="pos&prep␣.."/>
</ transition >
< transition source="S1" target ="S1">
  <w entry="le" tokens="0" tag="pos&art␣.."/>
</ transition >
```

Ambiguïtés simples

Peuvent être gérées grâce aux structures de traits :

```
<w entry="manger">
    <token>mange</token>
    <fs>
        <f name="pos" fVal="v"/>
        <f name="pers"> <vAlt><sym value="1"><sym value="3"></vAlt></f>
        <f name="tense" fVal="pres"/>
        <f name="mode" fVal="ind"/>
    </fs>
</w>
```

Notes

- utilisation des bibliothèques de traits

```
<fLib>
  <f id="pers@1.3" name="pers"> <vAlt><sym value="1"><sym value="3"></vAlt></f>
</fLib>
<w entry="manger" tokens="0" tag="pos@v␣pers@1.3␣tense@pres␣mode@ind"/>
```

- existence d'un forme plus complexe avec alt et join pour gérer la disjunctions pour le mode impératif.

Ordonnancement

Motivations : grande liberté d'ordre des balises mais faciliter les traitements

- les tokens forment un flux ordonnés
- Ne pas référencer un objet non encore défini :
  token < state < transition < w
- Favoriser la localité des informations (pour du streaming)
  mais rien n'interdit de repousser les annotations w (state et transition) en fin de phrase ou de document.

```
<token id="0">il</token>
<w entry="mylex#cl" tag=" .." />
<token id="1">mange</token>
<w entry="mylex#manger" tag=".."/>
<token id="2">.</token>
<w entry="mylex#dot" tag=" .." />
```

```
<token id="0">il</token>
<token id="1">mange</token>
<token id="2">.</token>
<w entry="mylex#cl" tag=" .." />
<w entry="mylex#manger" tag=".."/>
<w entry="mylex#dot" tag=" .." />
```

Quelques catégories

Parties du discours et sous-catégorisation

noun sous-cat : proper, common

adjective

Adverb

Verb sous-cat : aux, main, raising, semi-aux or modal

Pronoun

Clitic sous-cat : enclitic, proclitic

Conjunction sous-cat : coord, subord

Determiner sous-cat : article

Ad-position sous-cat : prepos, postpos

Interjection

Punctuation sous-cat : weak, strong

Complementizer

Prederterminer

Residual

Specifier

Quelques catégories (suite)

gender

number

person

mood

tense

voice

case

aspect

Programme de travail

- Rédiger un document plus complet s'appuyant a priori sur le contenu de cette présentation
  $\Rightarrow$ mise sous forme de schéma XML

- Examen des schémas actuels d'annotation dans cette optique
  $\Rightarrow$ examen de fragments de corpus annotés

- Discussions pour améliorer, amender ou **rejeter**