ATOLL - INRIA Rocquencourt

http://atoll.inria.fr

Call for contributions on morpho-syntactic annotations
First notes

Éric de la Clergerie & Lionel Clément

Eric.De_La_Clergerie@inria.fr & Lionel.Clement@inria.fr

ISO TC37 SC4 WG1 Meeting

Sapporo – July 7th 2003

Slides available at http://atoll.inria.fr/RNIL/sapporo03.pdf

Context

- French national action NORMALANGUE about Normalization of Linguistic Resources
- ISO TC37 SC4 French Mirror (AFNOR) – http://atoll.inria.fr/RNIL/

⇝ Proposition of a new work item about Morpho-Syntax annotation
with a call for contributions and documents, such as

- tagsets
- annotation guidelines
- annotated corpus fragments

- Why morpho-syntax annotations ?
  - First and important layer of linguistic processing
  - a lot of diversity (practices, languages) but some hints of convergences
  - some promising models (Feature structures, Finite State Automata)

Objectives of the talk

- Sketch a model for morpho-syntactic annotation
- Provide elements of information about practices for morpho-syntactic annotation.
  - based on received contributions
  - way for convergence ?
- Start a discussion on morpho-syntactic annotations

Sketching a model : a few principles

- Guided by the propositions made during last ISO TC37 SC4 WG1 [LAF] meeting (Pont à Mousson, Nov. 2002)
- Use XML as representation language (emphasis on human readability)
- Use of XML Schema (or RElax NG) as specification language, instead of just DTDs
  ⇒ wider modularity
  ⇒ schema extensibility
  ⇒ familly of **related** annotation schemata, with possibility of **conversion** using XSLT
- Use of other standards (or propositions),
  - TEI, EAGLES,MULTEXT
  - Feature Structure (Drafts WG1 N17, N23, N33 ; Kyong Lee)
  - Use of Data Category Registry, (Drafts Laurent Romary)
    i.e. registering of linguistic terms and concepts to be (preferably) used.
- Representation compatible with manual or automatic annotations
- A complete model, but simpler representations for simpler cases.
  variations for current practices

Towards a two level model

Segmentation Based (not exclusively) on typography and morphology

- Follow the text stream
- Syntagmatic relations (finite set of elements, present units in the text)

Tagging Associative relations based (not exclusively) on distributional analysis, substitution relations, and morphological analysis,

- A type to classify relations (category)
- Properties (distributional property, morphological property ...)
- A relationship between the units and the meaning or the lexicon (lemma, lexemes ...)

+ an interaction level to connect segmentation and tagging

Segmentation : token

Motivation : Anchor annotations on segments of a document
we assume the document is a **stream** (text, speech, video, indexed texts ...)

This level should also be adapted as output for a **tokenizer**

2 variants :

Embedded Inclusion of lexical content

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

Note : Token content may be complex (embedding XML elements)

Reference Low level reference to a segment, for instance in a multimedia document.

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

Notes :

- The content of span attribute should rely on standards for multimedia documents (MPEG-4 ?)
- The attribute span could be replaced by attributes from and to

Segmentation : more delicate cases

Segmentation/tokenization raises some problems !

Contraction

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

```
<token form="à" id="0">auquel</token> <!-- to whom -->
<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 : What possible value of attribute type ? Really needed ?

Segmentation : questions

- Nature of the segmentation :
  - partition (no) : too much restriction
  - overlapping (yes) :

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

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

  - skipped parts (yes) : silences and noises (speech), spaces, ...
  - discontinuous segments (no) : « afin seulement de » (in order [only] to)

- Attribute form : which content ?
  useful for naming segment content (multimedia), typo corrections and variations

- Element name : token vs more neutral term segment
- Scope of Ids (case of large documents) : looking for generic mechanisms ?

Segmentation : s

May be useful, but no real need of a segmentation for sentence.

```
<s>
    <token id="0">She</token>
    <token id="1">eats</token>
    <token id="2">.</token>
</s>
```

Other level of segmentation (paragraphs, . . .) to be handled by other standards (TEI).

Morpho-syntactic annotations : w

Motivation : Provide morpho-syntactic information about a word (pos, lemma, morphology, ...)

The pieces of information :

- may address one or more tokens
- should rely on feature structures
- may embed sub-components

```
<token id="1">prime</token>
<token id="2">minister</token>
```

```
<!-- Full notation -->
<w entry="prime minister" tokens="1 2">
    <fs>
        <f name="pos"><sym value="n"/></f>
        <f name="num"><sym value="sing"/></f>
    </fs>
    <w entry="prime">...</w>
    <w entry="minister">...</w>
</w>
```

Alternate embedded forms for w

- Embedding of tokens inside w or in its components

```
<w entry="prime minister">
  <token id="1">prime</token>
  <token id="2">minister</token>
  <fs> ... </fs>
  <w entry="prime">...</w>
  <w entry="minister">...</w>
</w>
```

```
<w entry="prime minister">
  <fs> ... </fs>
  <w entry="prime">
      <token id="1">prime</token>
      ...
  </w>
  <w entry="minister">
      <token id="2">minister</token>
      ...
  </w>
</w>
```

Notes :

- Embedded w may have no reference to tokens (but topmost one must have a reference)
- attribute tag renamed into feats ?
- attribute entry reference to an entry in a lexicon ?
  $\Rightarrow$ use of some addressing notation (Xpointers, namespace)

Annotations : compact notations

Current practices favors tagsets (e.g. MULTEXT) for compactness

Libraries for values, features and structures proposed in FS draft provide such compact notations :

```
<fvLib> <!-- Value library -->
    <sym id="n" value="n"/>
    <sym id="sing" value="sing"/>
    <sym id="fem" value="fem"/>
</fvLib>
<fLib> <!-- Feature-Value library -->
  <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"> <!-- potato -->
     <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"/>
```

Part of speech as FS type

Part of speech (POS) has generally a special status, governing the properties that may be attached :

```
<w entry="pomme de terre" tokens="1 2 3">
    <fs type="noun" feats="num@sing gen@fem" />
</w>
```

Slight problem for very compact notation : need of an attribute pos (or type) ?

```
<w entry="pomme de terre" tokens="1 2 3" pos="noun" tag="num@sing gen@fem" />
```

Annotations : features and values

Question : Where values and features come from ? What do they mean ?

Principles :

- (minimal) Use FS libraries and declaration mechanisms to specify
  - types
  - features, for a given type
  - allowed values, for a given feature (and a given type)

  Add commentaries !

- (maximal) Use registered feature and values from the Data Category Registry.

- (intermediary) link private categories to registered categories

- Use of namespaces to avoid conflicts and confusions between categories used with different meaning.

Annotations : linking categories

Motivation : Proposition of mechanisms to link private categories to registered categories allowing conversion (with information losses).

sub-typing Declaration of a private category as a sub-category of a registered one.
example : `advneg` subtype of `adv`

equation Equation between a private FS and another build using registered categories

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

Such a mapping may be handled (up to some point) by :

- XML schemata (subtyping)
- XSL transformations.

Fine vs coarse segmentation

Depending of the granularity of the segmentation, more than one way to represent equivalent information :

- Fine segmentation

```
<token form="etcetera" id="0">etc.</token>
<token form="DOT" id="1"/>
<w tokens="0" entry="etcetera" ... >
<w tokens="1" entry="punct:dot" tag="pos@punct">
```

- Coarse segmentation

```
<token form="etc." id="0">etc.</token>
<w tokens="0" entry="etcetera" ... >
<w tokens="0" entry="punct:dot" tag="pos@punct">
```

Similar remark for compound words (german)

Segmentation and annotation interaction

For simple non-deterministic case (manual annotations),
token and w are enough :

```
<token id="0">He</token>
<w entry="mylex#cl" tokens="0" tag="pos@clitic case@nom pers@3 gen@masc"/>
<token id="1">eats</token>
<w entry="mylex#eat" 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#eat">
        <token>eats</token>
        <fs> ... </fs>
    </w>
```

Dealing with ambiguities

Motivations :

- problems of ambiguities (automatic annotations)
- compatibility with word lattices (vocal recognition)
- Parser input entry (word lattice or FSA)

$\Rightarrow$ use of a representation of Finite State Automata (FSA), possibly with weights.

```
<!-- baked potatoes -->
<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 declaration -->
<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 : Elements state not really informative.

Complex interactions (Cont'd)

```
<token id="0">des</token> <!-- 'pl␣undef␣det' or 'of␣pl␣def␣det' -->
<state id="S0"/>
<state id="S1"/>
<state id="S2"/>
<transition source="S0" target="S1">
  <w entry="un" tokens="0" tag="pos@det"/>
</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@det␣.."/>
</transition>
```

Simple ambiguities (for instance morphology)

May be handled by Feature Structures

```
<w entry="eat">
    <token>eat</token>
    <fs>
        <f name="pos" fVal="v"/>
        <f name="pers"> <vAlt><sym value="1"><sym value="2"></vAlt></f>
        <f name="tense" fVal="pres"/>
        <f name="mode" fVal="ind"/>
    </fs>
</w>
```

Notes

- Again, use of FS libraries for frequent cases

```
<fLib>
  <f id="pers@1.2" name="pers"> <vAlt> ... </vAlt></f>
</fLib>
<w entry="eat" tokens="0" tag="pos@v pers@1.2 tense@pres mode@ind"/>
```

Simple ambiguities (for instance on lexical entries)

Several lexical entries for a same word :

```
<token id="0">doctor</token>
<alt>
  <w tokens="0" entry="mylex#med#doctor" ...>
  <w tokens="0" entry="mylex#edu#doctor" ...>
</alt>
```

Local complex ambiguities

How to switch locally between simple and full notations ?

```
<token id="0">il</token>
<token id="1">sort</token>
<token id="2">les</token>
<token id="3">pommes</token>
<token id="4">de</token>
<token id="5">terre</token>
<token id="6">.</token>
```

```
<w tokens="0" entry="he" tag="pos@cl ..."/>
<w tokens="1" entry="dig_out" tag="pos@v ..."/>
<w tokens="2" entry="the" tag="pos@det ..."/>
<fsm>
    <state id="s1" type="init"/>
    <state id="s2"/><state id="s3"/>
    <state id="s4" type="final"/>
    <transition source="s1" target="s4">
        <w tokens="3 4 5" entry="potato" .../>
    </transition>
    <transition source="s1" target="s2">
        <w tokens="3" entry="apple" .../>
    </transition>
    <transition source="s2" target="s3">
        <w tokens="4" entry="from" .../>
    </transition>
    <transition source="s3" target="s4">
        <w tokens="5" entry="earth".../>
    </transition>
</fsm>
<w tokens="6" entry="dot" tag="pos@punct ..."/>
```

Rule :

- implicit state before entering fsm merged with init state
- implicit state after exiting fsm merge with final state

Problematic : Handling discontinuous words

Problematic wrt interaction segmentation and tagging

afin seulement de

Handling of discontinuity

- at segmentation level : Rejected ?

```
    <token form="afin de" span="0 1 2 3" />
    <token form="seulement" span="1 2" />
```

- using interaction : still delicate : depends on further use

```
<token id="1" />afin</token>
<token id="2" />seulement</token>
<token id="3" />de</token>
```

```
<w tokens="1 3" entry="afin de" />
<w tokens="2" entry="seulement" />
```

```
<w tokens="1" entry="afin de/1" />
<w tokens="2" entry="seulement" />
<w tokens="3" entry="afin de/2" />
```

```
<w tokens="1 3" entry="afin de/1" />
<w tokens="2" entry="seulement" />
<w tokens="1 3" entry="afin de/2" />
```

Element ordering

Motivations : flexibility to order elements, but processing should be kept easy.

- tokens should be organized as an ordered stream
- avoid referencing to still undefined objects
  token < state < transition < w
- Favor information locality (for streaming)
  but still possible to move annotations w (state and transition) at end of sentences or documents.
  Primary documents, segmentation and morpho-syntactic annotations may even be in separate documents.

```
<token id="0">He</token>
<w entry="cl" tokens="0" .../>
<token id="1">eats</token>
<w entry="eat" tokens="1" .../>
<token id="2">.</token>
<w entry="dot" tokens="2" .../>
```

```
<token id="0">He</token>
<token id="1">eats</token>
<token id="2">.</token>
<w entry="cl" token="0" .../>
<w entry="eat" token="1" .../>
<w entry="dot" token="2" .../>
```

Note : Also possibility of token embedding in w

Some received contributions

Following our call for contributions, we received :

- Answers
  - STTS (Stuttgart-Tuebingen Tagset)
  - CSAI University of Malta (Albert Gatt)
  - Japan
- Tagsets
  - STTS
  - ELM-DE
  - TIGER
  - Verbmobil
  - Multext
  - Eagle
  - Paris 7 Tree bank
- Segmentation
  - STTS
  - Paris 7 Tree bank

Appropriateness with contributions : segmentation

Segmentation

- Generally a simple sequence
  $\text{token}_1/\text{tag}_1\ \text{token}_2/\text{tag}_2\ \ldots$
- Some times :
  - structural organization
    $\text{compounds} < \{\text{component}_1\ \text{component}_2\ \ldots\text{component}_k\}$
  - Splitting of a segment in two tags $\text{token}_1\$\text{tag}_1\ \text{token}_1\$\text{tag}_2$ ou $\text{token}_1/\text{tag}_1+\text{tag}_2$
  - Several *segments* for the same *tag*
    $\text{token}_1\$\text{tag}_1\ \text{token}_2\$\text{tag}_2\ \ldots\text{token}_k\hat{}\text{tag}_1$
  - Several *segments* which are recombined
    $\text{token}_1\%\text{tag}_1\ \text{token}_2/\text{tag}_1$

Appropriateness with contributions : tagging

- Tagging : the needs
  1. Part Of Speech (Partitioning of the words)

     More or less classical part of speech list or distributional categories.

     Heterogeneous categories (numbers, punctuation, foreign words, abbreviations, residuals, single member class ...)
  2. Sub-categorization (semantic, morphology ...)

     Very various and large sub-categorization (distributional properties, meaning features, valence ...)
  3. Morphological features (number, gender, mood, case ...)

     Common subsets but large diversity
  4. Lemma, Lexeme, ...

     Uncommon information
  - Typology for each language and project

    $\Rightarrow$ No consensus on definitions
  - Hierarchical structuration : POS $>$ sub-type $>$ morphological features.

    $\Rightarrow$ The **FS** recommendations are enough for this encoding

Parts of Speech from several projects

Multext	Eagles	STTS	Malte	Paris 7
adj	adj	adj	adj	adj
adposition	adposition	adposition	adposition	
adv	adv	adv	adv	adv
art	art	art	art	
conj	conj	conj	conj	conj
det	det			det
int	int	int	int	int
noun	noun	noun	noun	noun
num		card	num	
pro	pro	pro		pro
resid	resid			
single	single		single/non ass.	
verb	verb	verb	verb	verb
abrev.				
		foreign		foreign
		part		
		pref		
		sign		
		punct	punct	punct
			pro/det	
				prep
				clitic

Notes :

- convergences on main categories, but a lot a diversity for other ones
- maybe need for hierarchical organization (subtyping)

Appropriateness with contributions : interaction

Segmentation : Needs (based on Paris 7 Corpus & STTS)

+ Several *tags* for 1 *word* (axe syntagm.) (STTS, P7)
+ Several *words* for 1 *tag* (STTS, P7)
+ Correction of the *word* (well-formed *word*) (STTS)
+ Empty sequence (P7)
+ Alternatives simple word / compound (P7)
+ Alternatives compound / component sequence (P7)
+ Alternatives of *tag*
? Discontinuity of the *word* (P7– graph representation)
? Segmentation correction (STTS)

Program

If acceptation of a new work item for morpho-syntactic annotations :

- Redaction of a more complete document based on this presentation
  ⇒ development of some XML schema
  ⇒ development of conversion scripts (XSL) for simplified variants

- Comparisons with existing schemata and practices for a larger language diversity

- Discussions to improve, modify or reject the current proposition

An application : tokenizer

Available at http://atoll.inria.fr/RNIL/tools-fr.html

1. *Raw* text → XML Document
2. Text segmentation using regular expressions
   - Allow overlapping ***etc.*** = ***etc.*** + **.**
   - Allow non-segmented parts (spaces, separators . . .)
   - Allow empty segments ***auquel***= ***auquel*** + ∅

```
<token id="t1" form="etc.">etc.</token>
<token id="t2" form="."/>
```

3. morpho-syntactic units (typographical corrections)
   - **-t-il** → **il**, **l'on** → **on**

```
<token form="on" id="t1">-t-on</token>
```

4. DAG construction
   - 1 transition = 1 lexical entry (*tokens* concatenation)

Other tools

Also available from ATOLL :

- Perl scripts to establish XML-based linguistic pipelines, wrapped around non XML tools
- Several tools for morpho-syntactic annotations
- A morpho-syntactic chain for French based on this linguistic pipeline
  Output format close from the proposed one
  Should be soon accessible as a WEB service