MAF: a Morphosyntactic Annotation Framework

Article · January 2005

CITATIONS
27

READS
366

3 authors, including:

Eric De la Clergerie
National Institute for Research in Computer Science and Control
95 PUBLICATIONS **824** CITATIONS

SEE PROFILE

MAF: a Morphosyntactic Annotation Framework

Lionel Clément*, Éric Villemonte de la Clergerie†

* lionel.clement@lefff.net
† INRIA, Rocquencourt, B.P. 105, 78153 Le Chesnay (France)
Eric.De_La_Clergerie@inria.fr

Abstract

In the context of ISO Sub-Committee TC37 SC4 for the normalization of linguistic resources, we are promoting a framework for handling morpho-syntactic annotations. This paper sketches the main ideas of this proposal based on a two level structuring for tokens and word forms, ambiguity handling through lattices, use of feature structures for morpho-syntactic content, and mechanisms to define comparable tagsets.

Introduction

Morpho-Syntactic Annotations provide an important layer of linguistic information to a document. Large amount of corpora have been and are still manually annotated, while more and more annotations are now automatically produced by linguistic tools. Many NLP tasks (such as terminology extraction, information extraction, parsing, ...) rely on these morpho-syntactic annotations.

While prior efforts have already been devoted to standardize morpho-syntactic annotations, no full consensus has yet been reached, partly because of the difficulty to agree on a tagset organizing morpho-syntactic contents for all human languages. Our ambition is more modest, in the sense that we are not trying to propose a single tagset (or a family of tagsets) but rather a generic way to anchor, structure and organize annotations (with similarities with (Bird and Liberman, 2001)), completed, and this is one of the specificities of this proposal, by mechanisms to specify tagsets and annotation contents that may be compared.

Our proposal MAF (*Morpho-Syntactic Annotation Framework*) takes place in the effort done by ISO sub-committee TC37 SC4 (http://www.tc37sc4.org/) for the normalization of linguistic resources and relies on other complementary proposals initiated by that committee and on guiding principles (Ide et al., 2003). Of course, we also wish to integrate ideas from previous proposals on morpho-syntactic annotations (and more generally on annotations) and are looking forward for a large consensus and compatibility with existing annotation systems such as (Cunningham et al., 1996; Bird et al., 2000; Dybkjær and Bernsen, 2000).

1. A generic model

As many recent standardization proposals, we favor the use of XML representations, because they ensure both human readability (or, at least, human understanding) and easier machine processing. Still, these XML representations should rely on some consistent XML-independent model.

Figure 1 presents a simplified view of the proposed model for morpho-syntactic annotations. An annotated document is formed by a raw document and a set of annotations. The annotations are carried by *word forms* covering zero, one or more *tokens* of the documents.

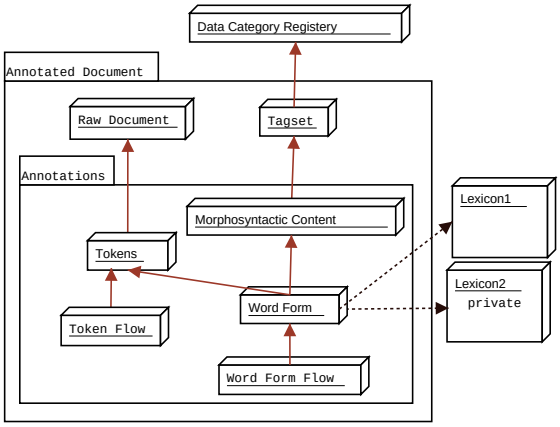

Figure 1: Simplified view of MAF model

To handle ambiguities, both word forms and tokens may be organized as flows materialized by *lattices*[1].

A word form may, in a non-mandatory way, reference a lexicon entry with, possibly, the use of more than one lexicon.[2] A word form may also only reference a morpho-syntactic property. In particular, a morpheme as a case mark or a person morphological inflexion, may be identified by a distinct word form, even if this morpheme is agglutinated in a word.

The morpho-syntactic content attached to a word form is expressed by *feature structures* following the guidelines of one or more *tagsets*. The terminology or set of *categories* (types, features, and feature values) used in tagsets are described w.r.t. *registered data categories* whose meaning has been clearly stated. Feature structures and registered data categories provide a promising direction to build tagsets that may be automatically compared, even if only approximatively.

The different components of the model may interact in more or less complex ways. A guiding principle for our proposal is to provide a gentle learning curve, with the

[1] Lattices may also be seen either as a restricted kind of *finite state automata* or as an extension of Directed Acyclic Graphs (DAGs).

[2] In particular, the use of document specific "lexica" was suggested, for collecting and organizing the named entities found in a document.

key idea that simple things should be simply represented. Therefore, we try to provide simplified alternate representations, relying on XML technology (for instance, XML schema and XSL transformations) to move from one level of representation to another one.

The current paper provides more information on the various components and variants of the model.

2. Tokens

We assume that a document presents a linear dimension and that it may be broken into *tokens* that identify non-empty continuous parts of the document. These tokens generally result from applying a tokenizer on a document. They are used to anchor linguistic units but need not be defined in a linguistic way. Actually, they are often defined by typographic rules (space separated sequences of characters for instance), by characters (for Asian languages), by phonemes (for oral documents), etc. Nevertheless, lexicon information is also a possible method to identify tokens.

The material covered by a token can be either embedded inside tokens or identified by a pair of *document positions*. These positions depend on the kind of documents being annotated. A non-exhaustive list of document position schema may include simple byte offsets, Unicode character offsets, time durations for speech, frames for video, etc. It should be noted that the embedded notation is only to be used for very simple documents and that the standoff notation, as suggested by the TIPSTER architecture (Grishman, 1997), is definitely a more robust option when dealing with more complex kinds of documents whose own structure may interact with the annotations.

A token may be completed by additional information (represented using XML attributes), for instance for transcriptions, transliteration, orthographic standardization, spelling correction, ... For instance, in modern Greek, the idiomatic expression "καλόκαγαθος" in Figure 2 (which means *good and brave*) may be segmented in three agglutinated segments "καλός", "και", and "αγαθος". The surface "surface graphical flow" is completed through the use of attribute `value` by additional information for each segment.

```
<token value="καλός" id="t0">
    καλο
</token>
<token value="και" id="t1">
    κ
</token>
<token value="αγαθός" id="t2">
    αγαθος
</token>
```

Figure 2: Use of attribute `value`

A yet to be fully formalized notion of *glue* has been suggested for specifying how two contiguous tokens are separated or joined (a space, nothing, a dash, an apostrophe, ...). We favor an interpretation of glues as a property of tokens (and represented by an XML attribute, Figure 3) but maybe they should have their own proper life and be represented as a special kind of (possibly empty) tokens. It should be noted that punctuation marks (such as dots or commas) have no reason to be understood as glues.

```
<token value="aujourd" id="t0">
    aujourd
</token>
<token value="hui" id="t1" glue="'">
    hui
</token>
```

Figure 3: Glue

3. Word Forms

A word form is a linguistic unit identified by its morpho-syntactic properties. Generally, this linguistic unit refers to some lexicon entry (materialized by the XML attribute `entry`). However, it should be noted that this reference is not mandatory, in particular for unknown words, neologisms or named entities. Furthermore, the linguistic units do not necessarily refer to lexicological descriptions but, instead, to morpho-syntactic descriptions of the words. In this case, the word forms are only identified through their morphological properties, for instance a case marker or an agglutinated verb clitic in Spanish.

Word forms are anchored by tokens but there is no one-to-one correspondence between tokens and word forms. A word form may cover several tokens (which may even be non contiguous) and, conversely, several word forms may be anchored by a same token. Furthermore, it may be noted that a same sequence of word forms may be differently anchored by tokens, depending on the granularity of the tokenization process. For instance, in French, the morphological agglutination of *auquel* («to whom») may have two distinct but equivalent representations, illustrated by Figure 4: a coarse tokenization where *auquel* is not decomposed but covered by a single `token`, with two word forms covering this segment, or a fine-grained tokenization identifying two agglutinated parts materialized by two tokens, each of them anchoring a word form.

```
<token id="t0">auquel</token>
<wordForm entry="à" tokens="t0"/>
<wordForm entry="lequel" tokens="t0"/>
```

```
<token value="à" id="t0">auquel</token
    >
<token value="lequel" id="t1"/>
<wordForm entry="à" tokens="t0"/>
<wordForm entry="lequel" tokens="t1"/>
```

Figure 4: Coarse vs fine-grained tokenizations

Tokens may be either embedded inside word forms or, better, referred to by a sequence of token identifiers (standoff notation with XML attribute `tokens`). A word form

like "*prime minister*" has an internal structure which may be materialized by embedding word forms for "*prime*" and "*minister*" (Figure 5). More generally, such internal structuring may be used to represent derivational morphology.

```
<wordForm entry="prime minister"
    tokens="t1 t2">
  <wordForm entry="prime">...
  </wordForm>
  <wordForm entry="minister">...
  </wordForm>
  ...
</wordForm>
```

Figure 5: Compound words

4. Morphosyntactic contents and tagsets

```
<token id="t0">mange</token>
<wordForm entry="manger" tokens="t0">
  <fs>
   <f name="mode">
     <symbol value="imperative"/>
   </f>
   <f name="number">
    <symbol value="singular"/>
   </f>
   ...
  </fs>
</wordForm>
```

```
<wordForm entry="lex:manger" tokens="
    t0" tag="mode@imp num@sing ..."/>
```

Figure 6: Word Form with morphological contents

Morphological information (including part of speech) is attached to word forms and expressed by feature structures, which are, roughly speaking, sets of feature-value pairs, where values may be atomic or (recursively) feature structures. The representation of these feature structures relies on the joint TEI-ISO proposal for "*Feature Structure Representation*" (FSR) (Lee et al., 2004) that covers many useful extensions such as alternations of values, lists of values, or sets of values.

Another (standard) extension provided by FSR is the possibility to assign a type to a feature structure. In particular, the part-of-speech may be seen as the value of a feature (say `pos`) but is more generally perceived as a type, because it selects a set of pertinent features (verbs and nouns do not select the same sets of features). The possibility to associate conditions to types is discussed below but is not covered by FSR.

Feature structures provide a very powerful and generic way to express partial information about the morphological properties of a word form. They can easily be understood by humans and processed by programs. However, feature structures tend to be rather verbose while current practices favor compact notations through `tags` (e.g. MULTEXT tags (Ide et al., 1996)). Fortunately, FSR provides the possibility to build libraries (`vLib`) of uniquely identified values (atomic or not) and libraries (`fvLib`) of uniquely identified feature-value pairs. These identifiers may be used in a way very similar to usual tags, providing a more compact notation, with the added advantage that they can be easily expanded in order to compare their contents.

Using feature structures is a first step toward a more uniform representation and processing of morpho-syntactic content but it does not ensure that everybody will use the same set of features or of values in a consistent way, or, in other words, with identical meaning.

By mapping types, features, and atomic values to data categories defined and registered in some global repository as encouraged by the proposal on "*Data Category Registries*" (DCR) (Ide and Romary, 2004), a greater compatibility may be achieved. A registered data category C (say *mode*) provides a textual definition for the linguistic concept (*verbal mode*) and possibly mention a *conceptual domain* as a list of other data categories (*indicative*, *subjunctive*, ...) that may be used as values for C. The name of the data category, its definition and its conceptual domain can be further refined on a language basis, for instance for French. We consider the mapping to registered data categories to be a very important step, but, still, it will not be mandatory to provide such a mapping and ways are being investigated to state simple partial mappings (for instance to declare a part-of-speech value `advneg` as a subkind of registered value `adv`).

Another possibility to improve understanding, ensure automatic processing, and automatically detect some errors is to specify the set of valid feature structures to be used in the annotations. A first solution is to use feature structure libraries to list, in an extensional way, all possible values and feature-values combinations. However, a more elegant and compact solution should be offered by a future companion proposal for FSR, namely "*Feature System Declaration*" (FSD). While not yet available, FSD should (at least) provide ways to specify the allowed set of features attached to a type and the set of possible values for a given feature in the context of a given type, following Carpenter's type hierarchies (Carpenter, 1992). Hopefully, such a system declaration should be expressible as an XML schema, in order to use standard XML technologies to validate annotations.

A *tagset* is therefore composed by (a) a selection of data categories, (b) feature structure declarations identifying valid morpho-syntactic content, and (c) feature structure libraries naming most common morpho-syntactic contents. A tagset may be specific to a document but, of course, we hope that a few largely used tagsets will progressively emerge. Preliminary investigations to express current tagsets such as those covered by MULTEXT have shown no major difficulties.

```
<vLib name="mode">
 <symbol value="imperative" id="imp"/>
 <symbol value="indicative" id="ind"/>
 <symbol value="subjonctive" id="subj"
     />
   ...
</vLib>
<fvLib name="fv_mode">
 <f fVal="imp" name="mode" id="
     mode@imp"/>
 <f fVal="ind|subj" name="mode" id="
     mode@ind|subj">
   <vAlt>
    <symbol value="indicative"/>
    <symbol value="subjonctive"/>
   </vAlt>
 </f>
 ...
</fvLib>
```

Figure 7: A tagset fragment

5. Handling Ambiguities

For most manually annotated documents, annotations can be simply represented by listing, in linear order, tokens and word forms. However, ambiguities may arise, especially in the context of automatic processing. We propose a very generic solution to capture ambiguities through a lattice of possibilities. Still, before presenting this solution, we also propose simpler solutions for handling simpler cases of ambiguities. Figure 9 shows an example of word form lattice for "*mange des pommes afin de grandir*" (*eat apples to grow*) illustrating different kinds of ambiguities.

5.1. Morphological ambiguities

Many morphological ambiguities can be straightforwardly handled by using alternations (`vALt`) inside feature structures. Compact tag notations still work by listing in libraries the most common cases of such ambiguities (cf. Figure 7, mode@ind|subj). Note that mutually dependent alternations cannot be elegantly represented by FSR (for instance, in French, an ambiguity present for many verbs between `2,imperative` or `1|3,indicative|subjunctive`).[3]

5.2. Lexical amnbiguities

Ambiguities between different lexical entries (or complex morphological ambiguities) may be handled by alternations on word forms (using XML element `wfAlt`).

5.3. Structural ambiguities

The remaining ambiguities are structural ones corresponding to distinct coverage of the tokens by word forms, or, more exceptionally, as distinct coverage of the input

[3]It should also be noted that we are not aware of any implementation of feature structures that can handle such mutually dependent alternations.

```
<token id="t0">mange</token>
<wfAlt>
 <wordForm entry="lex:manger" tokens="
     t0" tag="mode@imp ..."/>
 <wordForm entry="lex:manger" tokens="
     t0" tag="mode@ind|subj ..."/>
</wfAlt>
```

Figure 8: Alternation on word forms

document by tokens (for instance, in the case of automatic segmentation of speech documents). Both kinds of structural ambiguities can be represented by lattices, that may be seen as a slight extension of DAGs (requiring to have a single initial node and a single terminal node) or as a slight restriction of Finite State Automata (no looping paths).[4] For sake of simplicity, we do not plan to provide ways to explicitly specify interactions between the token and word form lattices[5], but rather plan to rely on the following implicit coherence constraint:

> the tokens covered by word forms along a path of the word form lattice belong to some path in the token lattice.

It is yet to be examined if this constraint can be easily checked using standard XML technology.

As a simplified variant and because ambiguities are generally localized, MAF provides the possibility to switch locally between the linear notation and the lattice-based notation as illustrated by Figure 10.

Structural ambiguities could have been alternatively described by "regular" expressions over word forms or tokens, using an operator for alternations and an operator for sequences. However, we believe lattices to be more readable for complex cases and more immediately processable. It is also easier to extend lattices to handle probabilities or metadata, for instance by adding attributes on edges.

6. Metadata

Metadata are clearly needed, for instance for specifying the author (or tool) of a set of annotations, the date, the confidence, ... However, we do not plan to provide a specific mechanism to handle metadata but rather to rely on other proposals such as (Wittenburg et al., 2000).

7. Conclusion

A demonstrator covering most of the features presented in this paper can be tried for French at `http://atoll.inria.fr/mafdemo` (and was used to produce Figure 9). In coordination with other experts involved in the development of MAF, we hope to see the

[4]The current XML representation is based of FSA terminology, with elements `transitions`, `state`, and `fsm`. However, this choice may be revised.

[5]This interaction could however be represented by moving to simplified chart structures, where an edge can list the edges from which it is derived.

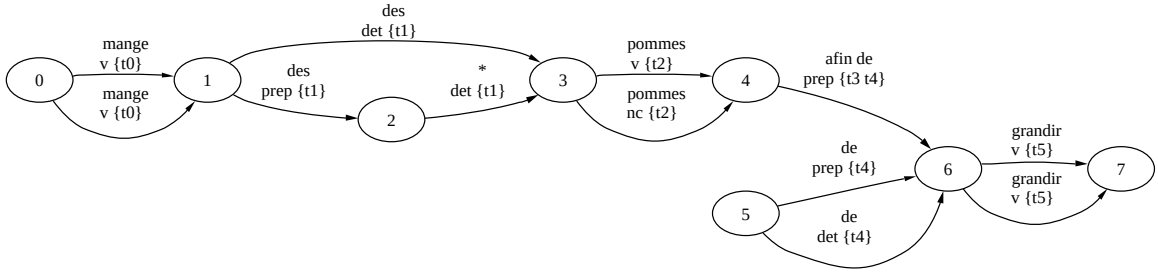

Figure 9: Ambiguities represented by a lattice

```
<maf language="fr" date="2005−03−21" author="mafd" addressing="byte">
 ...
 <token to="9" value="des" from="6" id="t1"/>
 <fsm final="3" init="1">
   <transition source="1" target="3">
     <wordForm tokens="t1" tag="def@− det@plus number@pl pos@det"/>
   </transition>
   <transition source="1" target="2">
     <wordForm entry="lex:de" tokens="t1" tag="pcas@de pos@prep"/>
   </transition>
   <transition source="2" target="3">
     <wordForm tokens="t1" tag="def@plus det@plus number@pl pos@det"/>
   </transition>
 </fsm>
 ...
</maf>
```

Figure 10: Local representation of ambiguities for “mange **des** pommes . . . ” (*eat apples* . . .)

fast emergence of other demonstrators for other languages and associated to various tagsets.

The MAF proposal has passed the first level of ISO evaluation process. We believe a large consensus should be reached before going further and hope this document will help.

8. References

Bird, S., D. Day, J. Garofolo, J. Henderson, C. Laprun, and M. Liberman, 2000. ATLAS: A flexible and extensible architecture for linguistic annotation. In *Proceedings of the Second Language Resources and Evaluation Conference (LREC)*. Athens.

Bird, S. and M. Liberman, 2001. A formal framework for linguistic annotation. *Speech Communication*, 33(1,2):23–60.

Carpenter, B., 1992. *The Logic of Typed Feature Structures with Applications to Unification Grammars, Logic Programs and Constraint Resolution*. Number ISBN 0-521-41932. Cambridge University Press.

Clément, L. and É. Villemonte de la Clergerie, 2004. Terminology and other language resources – morpho-syntactic annotation framework (MAF). ISO TC37SC4 WG2 Working Draft 24611.

Cunningham, H., Y. Wilks, and R. Gaizauskas, 1996. Gate – a general architecture for text engineering. In *Proceedings of the 16th Conference on Computational Linguistics (COLING-96)*.

Dybkjær, L. and N.O. Bernsen, 2000. The MATE workbench. In *Proc. of LREC 2000 Workshop*. Athens.

Genelex 93, 1993. *Projet Eureka Genelex - Rapport sur la couche Syntaxique - Rapport sur la couche morphologique*. Consortium Genelex.

Grishman, R., 1997. TIPSTER architecture design document version 2.3. Technical report, DARPA.

Ide, N. and L. Romary, 2004. A registry of standard data categories for linguistic annotation. In *Proceedings of the Fourth Language Resources and Evaluation Conference (LREC)*. Lisbon.

Ide, N., L. Romary, and E. Villemonte de la Clergerie, 2003. International standard for a linguistic annotation framework. In *Proceedings of HLT-NAACL'03 Workshop on The Software Engineering and Architecture of Language Technology*.

Ide, N., J. Véronis, and G. Priest-Dorman, 1996. Corpus encoding standard. Technical report, EAGLES/MULTEXT.

Lee, Kiyong, H. Bunt, S. Bauman, L. Burnard, L. Clément, É. de la Clergerie, T. Declerck, L. Romary, A. Roussanaly, and C. Roux, 2004. Towards an international standard on feature structure representation. In *proc. of LREC'04*.

Wittenburg, P., D. Broeder, and B. Sloman, 2000. EAGLES/ISLE: A proposal for a meta description standard for language resources, white paper. In *Proceedings of LREC 2000 Workshop*. Athens.