

# Where opposites meet. A Syntactic Meta-scheme for Corpus Annotation and Parsing Evaluation

Alessandro Lenci, Simonetta Montemagni, Vito Pirrelli, Claudia Soria

Istituto di Linguistica Computazionale – CNR  
Via Alfieri 1 - Pisa 56010 - ITALY  
lenci,simo,vito,soria@ilc.pi.cnr.it

## Abstract

The paper describes the use of FAME, a functional annotation meta-scheme for comparison and evaluation of syntactic annotation schemes, i) as a flexible yardstick in multi-lingual and multi-modal parser evaluation campaigns and ii) for corpus annotation. We show that FAME complies with a variety of non-trivial methodological requirements, and has the potential for being effectively used as an “interlingua” between different syntactic representation formats.

## 1. Motivation and background

It is widely acknowledged that the output of parsing systems should be assessed in terms of the linguistic information they are intended to provide, and not for how well they meet the requirements of a given annotation scheme taken as a reference standard. At the same time, it makes a lot of sense to compare the amount of information provided by different parsers by translating their output into a common parlance. Elsewhere (Lenci et al., 1999a; Lenci et al., 1999b), we suggested that the *prima facie* incompatibility of these two requirements can be tackled by providing a formal framework specifically designed for comparison and evaluation of existing syntactic annotation practices. We called this framework an *annotation meta-scheme*.

The key-ideas underlying the design of a meta-scheme can be summarized as follows. Different theoretical frameworks appear to impose different requirements on what phenomena should be annotated in a corpus, and how this should be done. One way of tackling the issue of theoretical compatibility stems from the observation that existing schemes mostly differ in the way pieces of linguistic information are mutually implied, rather than in the intrinsic nature of this information. To give a concrete example, almost all theoretical frameworks make use of the notion of grammatical subject, but only a subset of them take it to be a primitive unit of syntax: some make this notion contingent on the structural position of a noun phrase in the tree representation of a sentence, others on verb agreement, others on predicate-argument information, yet others on lexical control etc.

A way out of this theoretical maze is to augment the expressive power of an annotation scheme both *horizontally*, i.e. by distributing the annotated information over mutually independent “coding layers”, and *vertically*, by further specifying the information conveyed by each such layer. For example, it is possible to make the repertoire of grammatical functions notationally independent of other aspects concerning their morphological, syntactic, predicative and pragmatic realization. This can be achieved by assigning each such piece of information an independent coding layer so as to make it explicit how grammatical functions relate to the linguistic information needed for their assignment both in real texts and across languages.

In this paper discussion will be focused on some interesting aspects of FAME (Lenci et al., 1999a), a meta-scheme for annotating texts at the level of grammatical relations such as subject, object, indirect object etc., explicitly designed for parsing evaluation. The reasons for choosing this particular level of annotation are not trivial and will be summarized in section 2.1. Moreover, we will touch on i) an operational instantiation of FAME as a full-fledged annotation scheme in its own right (section 4.), and ii) the compatibility of FAME with other state-of-the-art annotation schemes (section 5.). That one and the same (meta) scheme can be used successfully on both scores paves the way to the development of multi-purpose, highly compatible and portable linguistic resources.

## 2. Requirements for a meta-scheme

FAME is designed to meet the following desiderata:

- factor out linguistically independent (but possibly correlated) primitive units of functional information;
- make explicit, through annotation, information which is otherwise only indirectly accessible in the parsed text;
- provide not only a measure of coverage but also of the utility of the covered information as opposed to missing information;
- avoid controversial analysis of a given phenomenon;
- rely on grammatical aspects which have the highest degree of inter-theoretical agreement;
- avoid framework-specific representational solutions.

All these requirements serve the main purpose of making the meta-scheme open to both annotation-dependent and task-dependent parameterization. This is felt important since the definition of closeness to a standard, and the utility of an analysis that is less-than-perfect relative to a given standard can vary from task to task, and, perhaps more crucially, from annotation scheme to annotation scheme. Information about how functional relations are actually instantiated in context is thus factored out into linguistically independent coding layers. For this purpose, we formally

represent a grammatical function as a binary relation consisting of the following pieces of information, structured as coding layers:

- i. the unordered terms of the relationship (i.e. the linguistic units in text entering a given functional relationship): example `(give, Mary)`;
- ii. the order relationship between the terms considered, conveying information about the head and the dependent: example `<give, Mary>`;
- iii. the type of relationship involved: example, the functional relation of the pair `(give, Mary)` in the sentence *John gave the book to Mary* is “indirect object”;
- iv. morpho-syntactic features associated with the dependent and the head; e.g. the dependent in the pair `(give, Mary)` is “non-clausal”;
- v. the predicate-argument status of the terms involved: for example `give(John, book, Mary)` in *John gave the book to Mary*.

Specifying all this information can in fact be *redundant* in some cases, as information at layer  $x$  can be presupposed logically by a piece of information encoded at layer  $y$ . For example, “nominative case” is often (but not always) a unique indicator of “subjecthood”, and the same holds for grammatical agreement. Yet, there is a general consensus that redundancy should not be a primary concern in the design of a standard representation (Sanfilippo et al., 1996). By assuming that all layers are, in a sense, primitive, rather than some of them being derivative of others, one provides considerable leeway for radically different definitions of functional relations to be cast into a common core of required information. We will return to this point in section 5. of the paper.

This surfeit of descriptive power does not make the FAME scheme an empty one. We contend that the approach has the potential for shedding light on nature and degree of correspondence between coding schemes developed for different purposes and with diverse, or even conflicting, theoretical requirements. Moreover, it seems to have a useful impact on our theoretical understanding of linguistic phenomena, since it is capable of expressing the degree of correlation between layers, and ultimately between dimensions of linguistic analysis. Finally, an annotator using this framework, whatever his/her own purposes and theoretical biases, is forced to relate his/her own tag set to a constrained number of pre-defined layers (i–v above), or, if extra layers or attributes are introduced, to give reasons for them.

### 2.1. Why functional annotation

In FAME, functional relations hold between head words belonging to major lexical classes only (i.e., non-auxiliary verbs, nouns and adjectives), independently of their relation to (possibly identified) phrasal constituents. The choice of this specific type of functional annotation is largely motivated on the basis of a number of practical concerns. We contend that information about how functional relations are

actually instantiated in a text is important for the following reasons:

- it is linguistically valuable, both as an end in itself and as an intermediate linguistic resource; in fact, it is sufficiently close to semantic representations to be used as an intermediate stage of analysis in systems requiring full text understanding capabilities;
- it is likely to become a more and more heavily used information asset for NLP applications: a shift of emphasis from purely pattern matching methods operating on  $n$ -word windows to functional information about word pairs has recently been witnessed both in the context of information retrieval/filtering systems (Grefenstette, 1994) and for the purposes of word sense disambiguation (see the last SENSEVAL and ROMANSEVAL evaluation campaigns);
- it is comparatively easy and “fair” to evaluate since it overcomes some of the shortcomings of constituency-based evaluation (Carroll and Briscoe, 1996; Carroll et al., 1998; Sampson, 1998; Lin, 1998);
- it represents a very informative “lowest common ground” of a variety of different syntactic annotation schemes (Lin, 1998);
- it is naturally multi-lingual, as functional relations probably represent the most significant level of syntactic analysis at which cross-language comparability makes sense;
- it is abstract enough to permit joint evaluation of systems dealing with both spoken and written language, and annotation of data typically fraught with cases of disfluency, anacoluthon, syntactic incompleteness and any sort of non-canonical syntactic structure (Antoine, 1995; Klein et al., 1998);
- it is “lexical” enough in character to make provision for *partial* and *focused* annotation: since a functional relation always involves only two lexical heads at a time, it is comparatively easy to evaluate an annotated text only relative to a subset of *relevant headwords*.

### 3. A brief survey of proposed standards

In this section we briefly survey existing standards for functional annotation to assess their suitability as a basis for the design of a functional annotation meta-scheme. The syntactic annotation practices considered here are limited to those put forward in the framework of the European projects EAGLES, SPARKLE and MATE. Actually, the EAGLES Recommendations and Guidelines for the Syntactic Annotation of Corpora (Leech et al., 1996) do not cover dependency-based models: however, reference is made to the Helsinki Constraint Grammar (Karlsson et al., 1995) as a key exemplar. Other functional annotation schemes were proposed in the framework of the European projects SPARKLE (LE-2111) and MATE (LE4-8370). We will refer to them respectively as the SPARKLE Scheme and the MATE Scheme, the latter being partially derived from the former.

Briscoe et al. (1998), largely drawing on (Carroll et al., 1996), emphasize the importance of the SPARKLE Scheme for NLP evaluation, to propose it as a standard representation format. Successful adoption of the SPARKLE Scheme in the MATE project certainly shows suitability for dialogue annotation (Mengel et al., 1999). Nevertheless, in our view, the SPARKLE Scheme falls short of satisfying all desiderata in section 2.

For instance, SPARKLE adopts a multi-stratal approach to syntactic analysis, whereby deep grammatical relations are distinguished from surface ones, in order to deal with displacement and/or movement phenomena. In particular, this assumption affects the representation of passive constructions in a crucial way, as shown by the following example:

```
subj(employ, John, obj) John was employed by IBM
```

The problem is that such an assumption is not shared by all syntactic frameworks, and therefore can distort parsing evaluation to a considerable extent when applied to parsing schemes that do not make a distinction between deep and surface subjects.

Another related issue is the treatment of pro-drop phenomena in sentences like Italian *ha dormito* ‘(he) slept’. The SPARKLE Scheme proposes to deal with pro-drop in terms of a subject relation between the verbal head and an empty element, so-called “pro”, as in `subj(dormire, pro)`. This solution is again likely to introduce a theoretical bias into parsing evaluation. Moreover, if “pro” elements are taken to have an independent existence at the level of surface realization of a sentence, then one would have to mark them in the body of a corpus to be annotated. In an Italian spoken corpus, where sentences with elliptical subjects represent the vast majority of attested sentences, this could easily result in a very laborious pre-processing of the material to annotate.

As another example, let us consider the taxonomies of functional tags in SPARKLE and in Karlsson’s Constraint Grammar (Karlsson et al., 1995). Both taxonomies collapse different information types into one annotation layer only, for reasons ranging from a theoretical bias towards a maximally economic description of the phenomena in question, or a particular view of the way syntactic phenomena are mutually implied from a logical standpoint, to choices chiefly motivated by the intended application. A typical example of this is the tag `xcomp` in the SPARKLE scheme, which, following Lexical Functional Grammar (Bresnan, 1982), covers *all subcategorized open predicates*: namely, traditional predicative complements (whether subject or object predicative), and unsaturated clausal complements, such as embedded infinitival and participial clauses (as opposed to, e.g., *that*-clauses). In Constraint Grammar, predicative nominal and adjectival phrases are tagged as “subject complement” or “object complement”, while, say, controlled infinitive clauses, as in *Mary wants to read*, are marked functionally as an “object” of the main verb. Any context-free attempt to map SPARKLE `xcomp` onto a Constraint Grammar tag, would inevitably be one-to-many and not necessarily information-preserving. Clearly, these aspects make it very hard to provide any sort of fair base-

line for comparing a SPARKLE annotated text against the same text tagged with Constraint Grammar labels.

On the basis of these observations, in (Lenci et al., 1999a) we illustrated a revision of the SPARKLE Scheme, for it to better comply with the methodological desiderata of section 2. In the remainder of this paper, we discuss how FAME can be used for purposes of annotation (section 4.), and how well it functions as an *interlingua* between different, and in some cases mutually incompatible, syntactic representation formats (section 5.).

## 4. Using FAME for annotation

To date, FAME covers coding layers i–iv of section 2. only. A concrete instantiation of FAME was developed in the framework of SITAL for tagging Italian texts at the level of functional annotation (Autori vari, 2000). This instantiation is based on the set of FAME dependency relations illustrated in detail in section 4.1. Some adjustments were required to make FAME suitable for annotation of unrestricted texts, e.g. introduction of further relation types to deal with coordination phenomena, relative clauses and the like. These extensions are illustrated in section 4.2.

### 4.1. Dependency relations

The building blocks of the proposed annotation scheme are functional relations, where a functional relation is an asymmetric binary relation between a word called HEAD and another word called DEPENDENT. We assume only relations holding between lexical or full words. Therefore, we exclude functional relations involving grammatical elements such as determiners, auxiliaries, complementizers, prepositions, etc. The information concerning these elements is conveyed through features, as described below in section 4.1.3.

Each functional relation is expressed as follows:

```
dep_type (lex_head.<head_features>,
          dependent.<dep_features>)
```

`Dep_type` specifies the relationship holding between the lexical head (`lex_head`) and its dependent (`dependent`). The head and the dependent of the relation are further specified through a (possibly empty) list of valued features (respectively `head_features` and `dep_features`), which complement functional information.

#### 4.1.1. The hierarchy of functions

`Dep_types` are hierarchically structured to make provision for underspecified representations of highly ambiguous functional analyses (see further below). The hierarchy of relations is given in figure 1 below. In the hierarchy, the function `subj` (for “subject”) is opposed to other grammatical relations by being assigned a higher prominence in the taxonomy, as customary in contemporary grammar theories (e.g. HPSG, GB). Moreover, modifiers and arguments are subsumed under the same `comp` node (mnemonic for complement), allowing for the possibility of leaving underspecified the distinction between an adjunct and a subcategorized argument in those cases where the distinction is

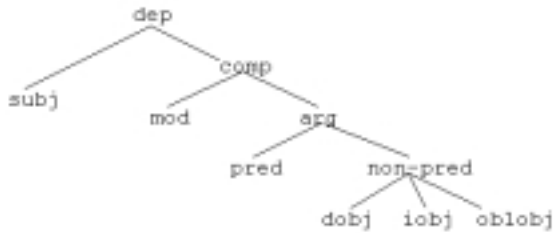


Figure 1: Hierarchy of functional relations

difficult to draw in practice. In turn, the node `arg` (for argument) is split into `pred`, subsuming all and only classical predicative complements, and `non-pred`, further specified into `dobj` (for direct objects), `iobj` (for indirect objects) and `oblobj` (for oblique arguments).

The hierarchy of figure 4.1.1. is a revision of the SPARKLE functional hierarchy (Carroll et al., 1996), in the light of the methodological points raised in sections 3. and 2. The main point of departure can be found under the node `comp`, which, in SPARKLE, dominates the nodes `obj` and `clausal`, thus reflecting a view of predicative complements as small clauses, to be assimilated with other unsaturated clausal constructions such as infinitival and participial clauses. This is in clear conflict with another grammatical tradition that marks clausal complements with the functional relations also assigned to non clausal complements, when the latter appear to be in a parallel distribution with the former, as in *I accept his position* and *I accept that he leaves*, where both *his position* and *that he leaves* are tagged as objects (Karlsson et al., 1995). This is a typical example of how functions may differ due to a difference in the levels of the linguistic information taken to be criterial for tag assignment. As we will see in more detail in section 4.1.2., the FAME hierarchy circumvents the problem by assigning all non-subject clausal complements the tag `arg`, which subsumes both traditional predicatives (`pred`) and non clausal arguments (`non-pred`), thus granting sentential complements a kind of ambivalent (underspecified) functional status.

#### 4.1.2. The typology of functions

In what follows we sketchily define each functional relation; examples are provided for non generic nodes of the hierarchy only.

`dep(head, dependent)` is the most generic relation between a head and a dependent, subsuming the distinction between a subject and a complement. It is particularly useful to handle ambiguous constructions for which no reliable bias is available for disambiguation. For example, both *Gianni* and *Mario*, *non l'ha ancora visto, Gianni* 'Mario has not seen Gianni yet' / 'Gianni has not seen Mario yet'. In this case, we can resort to an underspecified representation, namely `dep(vedere, Mario)` and `dep(vedere, Gianni)`.

`subj(head, dependent)` is the relation between a

predicate and its subject:

`subj(arrive, John)` *John arrived in Paris*  
`subj(employ, Paul)` *Paul was employed by IBM*

`subj` refers to the superficial subject of a verb, regardless of the latter being used in the active or passive voice. Also clausal subjects are marked as `subj`:

`subj(mean, leave)` *that Mary left meant she was sick*  
`subj(require, win)` *to win the America's Cup requires heaps of cash*

With pro-drop languages such as Italian, when the subject is not overtly realised the annotation is partial, as specified below:

`subj(arrivare, )` *arrivai in ritardo* 'I arrived late'

where i) the dependent slot is left unspecified and ii) the morphosyntactic features, which indicate person, number and gender of the subject, can be recovered from the inflectional features associated with the head (see section 4.1.3.).

Moreover, `subj` can also be used to mark subject control relations and, possibly, raising to object/subject phenomena, as exemplified below:

`subj(leave, John)` *John promised Mary to leave*  
`subj(leave, Mary)` *John ordered Mary to leave*  
`subj(be, her)` *John believes her to be intelligent*  
`subj(be, John)` *John seems to be intelligent*

Finally, `subj` is also used to mark the subject of predicative complements, as in:

`subj(intelligent, John)` *John is intelligent*

With these last two cases FAME goes beyond surface syntax proper and annotates information relative to the predicate-argument structure, corresponding to level v. in the information layers of the proposed meta-scheme.

`comp(head, dependent)` is the most generic relation between a head and a complement, whether a modifier or a subcategorized argument. This underspecified functional relation is particularly useful for those cases where it is difficult to draw a line between adjuncts and subcategorized elements. This is a crucial issue if one considers the wide range of variability in the subcategorization information contained in lexical resources. For example, given the sentence *John pushed the car to the station*, the following representation:

`comp(push, station)`

is compatible with both possible analyses, whether *to the station* has to be considered as a modifier or as an argument. Other cases in which the underspecified representation in terms of `comp` is particularly useful are in the annotation of i) the relation holding between a noun head and a prepositional complement, and ii) the relation between a head and a semantic argument syntactically realised as a modifier (as in the case of the agent as expressed in the passive construction), e.g.:

`mod(professor, Physics)` *the professor of Physics*  
`mod(kill, Brutus)` *he was killed by Brutus*

`mod(head, dependent)` holds between a head and its

modifier, whether clausal or non-clausal; e.g.

```
mod(flag, red) a red flag
mod(walk, slowly) walk slowly
mod(Picasso, painter) Picasso the painter
mod(walk, talk) walk while talking
```

`arg(head, dependent)` is the most generic relation between a head and a subcategorized argument; it is typically used to tag the syntactic relation between a verbal head and a non-subject clausal argument:

```
arg(say, accept) He said that he will accept the job
arg(promise, leave) John promised Mary to leave
arg(intend, leave) Paul intends to leave IBM
```

This choice was discussed in section 4.1.1. and motivated as a way to circumvent the theoretical issue of whether the functional relations of clauses should be defined on the basis of their predicative status, or, alternatively, of their syntactic distribution.

`pred(head, dependent)` is the relation which holds between a head and a predicative complement, be it subject or object predicative, e.g.

```
pred(be, intelligent) John is intelligent
pred(consider, genius) John considers Mary a genius
```

`nonpred(head, dependent)` is the relation which holds between a head and a non predicative complement.

`dobj(head, dependent)` is the relation between a predicate and its direct object (always non-clausal), e.g.:

```
dobj(read, book) John read many books
```

`iobj(head, dependent)` is the relation between a predicate and the indirect object, i.e. the complement expressing the recipient or beneficiary of the action expressed by the verb, e.g.

```
iobj(speak, Mary) John speaks to Mary
iobj(give, Mary) John gave Mary the contract
iobj(give, Mary) John gave the contract to Mary
```

`oblobj(head, dependent)` is the relation between a predicate and a non-direct non clausal complement, e.g.

```
oblobj(live, Rome) John lives in Rome
oblobj(inform, run) John informed me of his run
```

### 4.1.3. Feature specification

In FAME, a crucial role is played by the features associated with both elements of the relation.

`Dep(endent)_features` are as follows:

- `intro(ducer)`: it refers to the grammatical word (a preposition, a conjunction etc.) which possibly introduces the dependent in a given functional relation, e.g.  

```
iobj(give, Mary.<intro=to>) give to Mary
```

```
arg(say, accept.<intro=that>) Paul said that he accepts his offer
```

- `Case`: it encodes the case of the dependent, e.g.

```
iobj(dare, gli.<case=DAT>) dargli 'give to him'
```

- `status`: this feature is associated with clausal dependents to distinguish open predicative functions (controlled from outside) from closed predicative functions (already saturated). Possible values of this feature are:

- `open`: a subcategorized clausal argument or modifier containing an empty argument position which must have an external controller, e.g.

```
arg(decide, leave.<status=open>) John decided to leave
```

- `closed`: a subcategorized clausal argument or modifier which requires no control by a constituent outside it, e.g.

```
arg(say, leave.<status=closed>) He ate the cake because he was hungry
```

- `mood`: the feature specifies the mood of a verbal head, e.g.:

```
mod(decide, leave.<mood=infinitive>) John decided to leave
```

`Head_features` are as follows:

- `diath(esis)`: it specifies the diathesis of a verbal head. Possible values are *active*, *passive*, and *middle*, e.g.

```
subj(employ.<diath=passive>, Paul) Paul was employed by IBM
```

```
subj(employ.<diath=active>, IBM) IBM employed Paul
```

- `pers(on)`: it specifies the person of a verbal head, e.g.

```
subj(eat.<pers=3>, he) he eats a pizza
```

- `num(ber)`: it specifies the number of a verbal head. e.g.

```
subj(eat.<num=sing>, he) he eats a pizza
```

- `gen(der)`: it specifies the gender of a head, e.g.

```
subj(arrivare.<gen=fem>, Maria) Maria è arrivata 'Maria has come'
```

For nouns, whether heads or dependents, the following additional features are specified:

- `quant(ifier)`: the value is represented by a quantifier: 

```
dobj(see, cat.<quant=many>) I've seen many cats
```

- `card(inality)`: the value is represented by a number:

```
dobj(see, cat.<card=two>) I've seen two cats
```

- `def( initeness )`: possible values are +, – and 0:  
`dobj( see, cat.<def=+> ) I've seen the cat`  
`dobj( see, cat.<def=-> ) I've seen a cat`  
`dobj( see, cat.<def=0> ) I've seen cats`

## 4.2. Other relation types

### 4.2.1. Coordination

In order to represent conjunctions and disjunctions, FAME avails itself of the two symmetric relations `conj` and `disj`, lying outside the dependency hierarchy. Consider, for instance, the sentence *John and Mary arrived* containing a conjoined subject whose FAME representation is reported below:

```
subj( arrive, John )
subj( arrive, Mary )
conj( John, Mary )
```

The FAME representation of the sentence *John or Mary arrived* differs from the previous one only in the type of relation holding *John* and *Mary*: namely, `disj( John, Mary )`.

### 4.2.2. Linking

In FAME, a `link` relation is used to annotate other clause-internal non-functional and co-referential bonds. In particular, a `link` relation is used to relate:

- a relative pronoun and its antecedent. In the Italian sentence *Il ragazzo che Maria ha incontrato è Giovanni* ('The boy Mary met is John'), the relative clause 'che Maria ha incontrato' is represented as follows:

```
subj( incontrare, Maria )
dobj( incontrare, che )
mod( incontrare, ieri )
link( che, ragazzo )
```

- a clitic and its intended co-referent in dislocated constructions. For instance, the Italian sentence *Il dolce Maria lo porta domani* 'The cake, Mary will bring it tomorrow' is represented as follows:

```
subj( portare, Maria )
dobj( portare, lo )
mod( portare, domani )
link( lo, dolce )
```

## 5. Using FAME for evaluation

**Theory-neutrality** Theory-neutrality is an often emphasised requirement for reference annotation schemata to be used in evaluation campaigns (see GRACE, Adda et al. (1998)). The problem with theory neutrality in this context is that, although some agreement can be found on a set of basic *labels*, problems arise as soon as the *definition* of these labels comes in. For example, the definition of "subject" as a noun constituent marked with nominative case is not entirely satisfactory, since a system might want to analyse the accusative pronoun in *John believes her to be intelligent* as the subject of the verb heading the embedded infinitival clause (as customary in some linguistic

analyses of this type of complements). Even agreement, often invoked as a criterial property for subject identification, may be equally tricky and too theory-loaded for purposes of parser comparison and evaluation.

As we saw, the approach of FAME to this problem is to separate the repertoire of functional relation types (labels), from the set of morpho-syntactic features associated with the head and dependent, as shown in the examples below:

```
subj( be, she.<case=accusative> ) John believes
her to be intelligent
subj( be, she.<case=nominative> ) She seems to
be intelligent
```

By doing this way, emphasis is shifted from theory-neutrality (an almost unattainable goal) to *modularity* of representation: a functional representation is articulated into different information levels, each factoring out different but possibly inter-related linguistic facets of functional annotation.

**Intertranslatability** A comparative evaluation campaign has to take into account that participant systems may include parsers based on rather different approaches to syntax (e.g. dependency-based, constituency-based, HPSG-like, LFG-like, etc.) and applied to different languages and test corpora. For a comparative evaluation to be possible, it is therefore necessary to make room for the specificity of a system, while at the same time guaranteeing the feasibility and effectiveness of mapping a system output onto the reference annotation scheme. It is important to bear in mind at this stage that:

- most broad-coverage parsers are constituency-based;
- the largest syntactic databases (treebanks) use constituency-based representations.

It is then crucial to make it sure that constituency-based representations, or any other variants thereof, be mappable onto the functional reference annotation meta-scheme. The same point is convincingly argued for by Lin (1998), who also provides an algorithm for mapping a constituency-based representation onto a dependency-based format. To show that the requirement of intertranslatability is satisfied by FAME, we consider here four different analyses for the sentence *John tried to open the window* together with their translation equivalent in the FAME format:

1. ANLT Parser (Briscoe & Carroll, 1995) - traditional PSG representation:

```
(Tp
 (V2 (N2 (N1 (N0 John_NP1)))
  (V1 (V0 tried_VVD)
   (V1 (V0 to_TO)
    (V1 (V0 open_VV0)
     (N2 (DT the_AT)(N1 (N0 window_NN1))
    )))))).
```

FAME equivalent:

```
subj( try, John )
arg( try, open.<intro=to> )
dobj( open, window )
```

## 2. Fast Partial Parser (Grefenstette, 1994):

```
SUBJ(try,John)
DOBJ(open>window)
SUBJ(open,John)
MODIF(open,try).
```

FAME equivalent:

```
subj(try,John)
dobj(open>window)
subj(open,John)
mod(open,try)
```

## 3. Finite State Constraint Grammar Parser (Karlsson et al., 1995):

```
John N SUBJ
tried V MVMAINC^
to INFMARK open V_INF MV OBJ^
the DET window NOBJ.
```

FAME equivalent:

```
subj(try,John)
arg(try,open.<intro=to,
      status=open>)
dobj(open>window)
```

## 4. PENN Predicate Argument structure (Marcus et al., 1994):

```
want(try(John,open(John,window))).
```

FAME equivalent:

```
subj(try,John)
arg(try,open)
subj(open,John)
dobj(open>window)
```

Let us suppose now that the reference analysis for the evaluation of the same sentence in FAME is as follows:

```
subj(try,John)
arg(try,open.<intro=to,status=open>)
subj(open,John)
dobj(open>window)
```

Notice that this representation differs from the output of the ANLT Parser and of the Finite State Constraint Grammar Parser mainly because they both give no explicit indication of the control relationship between the verb in the infinitive clause and the matrix subject. This information is marked in the output of both the Fast Partial Parser and the PENN predicate–argument tagging. Note further that the Fast Partial Parser gives a different interpretation of the infinitival complement, which is marked as being modified by *try*, rather than being interpreted as a direct object of *try*.

FAME does justice to these subtle differences as follows. First, it should be reminded that the FAME equivalents given above are in fact shorthand representations. Full representations are distributed over four coding layers, and precision and recall are to be gauged jointly relative to all such layers. To be concrete, let us first show a full version of the FAME standard representation for the sentence *John tried to open the window* (cf. Section 2.2):

```
i. (try,John)
ii. <try,John>
iii. subj
    i. (try,open)
    ii. <try,open>
iii. arg
    iv. open.<introducer=to,status=open>
        i. (open,John)
        ii. <open,John>
iii. subj
    i. (open>window)
    ii. <open>window>
iii. dobj
```

Note first that information about the unsaturated clausal complement *to open* is separately encoded as *status=open* in the standard representation. Failure to explicitly annotate this piece of information, incurred by ANLT and the Constraint Grammar Parser, will then be penalised in terms of *recall*, but would eventually not affect *precision*. By the same token, the subject control relation between *John* and *open* is recalled only by the Fast Partial Parser and PENN, and left untagged in the remaining schemes, thus lowering recall. The somewhat unorthodox functional dependency between *try* and *open* proposed by the Fast Partial Parser will receive the following full-blown FAME translation:

```
mod
(try,open)
<open,try>
```

When compared with the standard representation, this translation is a hit at the level of identification of the unordered dependency pair (*try,open*), although both the order of elements in the pair (*<open,try>*) and their functional dependency (*mod*) fail to match the standard. On this specific dependency, thus, recall will be  $\frac{1}{3}$ . As a more charitable alternative to this, it can be suggested that the difference between the FAME standard and the Fast Partial Parser output is the consequence of theory internal assumptions concerning the analysis of subject-control structures, and that this difference should eventually be leveled out in the translation into FAME. This may yield a fairer evaluation, but has the drawback, in our view, of obscuring an important difference between the two representations.

## 6. Conclusions

In this paper we described the use of FAME, a functional annotation meta–scheme, as a fair basis for the construction of linguistic resources such as annotated corpora and for parsing evaluation.

FAME main features are i) its distributed coding architecture and ii) the hierarchical organization of functional relations. The distributed architecture of FAME has an obvious impact on the definition of syntactic tags. FAME separates the repertoire of functional relation types (labels),

from the set of morpho-syntactic features associated with the terms of the relation (head and dependent), instead of making the former depending on the latter, as often the case in extant annotation schemata. By doing this way, emphasis is shifted on representation modularity: a functional representation is articulated over different information layers, each factoring out different, but possibly inter-related, linguistic facets of functional annotation.

The distributed information structure of FAME is conducive to incremental evaluation, ranging from a base evaluation level (involving sheer identification of the terms in a syntactic relationship and/or their order), to deeper levels, including morpho-syntactic information, dependency type, and ultimately predicate-argument structure. The evaluation of a text annotated for functional information can then be conceived of as a function of estimating precision and recall for each of the independent coding layers. Evaluation results obtained for the different layers can eventually be combined together or, for particular purposes, assessed in their own right (e.g. for IR applications a basic evaluation at layer *i* could be sufficient). Thanks to such a distributed structure, FAME can easily be extended through addition of still further coding layers.

FAME hierarchical organization of functional relations makes it possible to resort to underspecified tags for notoriously hard cases of functional disambiguation, while providing, at the same time, an upper constraint on the range of available choices for scheme design and actual annotation. For evaluation purposes, underspecification guarantees a more flexible and balanced assessment of a system output, especially with relation to those constructions whose syntactic analysis is still controversial.

Finally, we showed how FAME was instantiated as a full-fledged annotation scheme in its own right. The scheme proves to be able to guarantee a) flexibility in actual coding, b) customizability, so as to be open to revision and usage by researchers with different theoretical orientations, and c) a high degree of portability as a theory neutral interlingua, translatable into many different syntactic idiolects. Provision of multi-purpose annotated texts of this kind is likely to represent, in our view of things, an interesting step forward in the design of new data architectures for linguistic resources, and an instrumental way to sharpen up our theoretical understanding of the annotated linguistic phenomena and their relationship to levels of linguistic analysis.

## 7. References

- Autori Vari 2000. SITAL: Manuale Operativo. *Deliverable 1.1*, Italy.
- Adda, G., Mariani, J., Lecomte, J., Paroubek, P. and M. Rajman. 1998. The GRACE French Part-of-speech tagging evaluation task. In *Proceedings of the First LREC*, pp. 443-441, Granada, Spain.
- Antoine, J. Y. 1995. Conception de Dessin et CHM. In K. Zreik et J. Caelen. *Le communicationnel pour concevoir*, pp. 161-184. Europia, Paris, France.
- Bresnan, J. 1982. *The Mental Representation of Grammatical Relations*. MIT, Cambridge, Massachusetts.
- Briscoe, T. and J. Carroll. 1995. Developing and evaluating a probabilistic LR parser of part-of-speech and punctuation labels. In *Proceedings of the Fourth ACL/SIGPARSE International Workshop on parsing technologies*, pp. 48-58, Prague, Czech Republic.
- Carroll, J., Briscoe, T., Calzolari, N., Federici, S., Montemagni, S., Pirrelli, V., Grefenstette, G., Sanfilippo, A., Carroll, G. and M. Rooth. 1996. Specification of Phrasal Parsing. *SPARKLE Deliverable 1*.
- Carroll, J. and T. Briscoe. 1996. Apportioning development effort in a probabilistic LR parsing system through evaluation. In *Proceedings of the ACL/SIGDAT Conference on Empirical Methods in NLP*, pp. 92-100.
- Carroll, J., Briscoe, T. and A. Sanfilippo. 1998. Parser evaluation: a survey and a new proposal. In *Proceedings of the First LREC*, pp. 447-454, Granada, Spain.
- Fillmore, C. J. 1968. The case for case. In E. Bach and R. Harms (Eds.), *Universals in Linguistic Theory*, pp. 1-88, Holt, Rinehart & Winston, New York, USA.
- Fillmore, C. J. 1985. Frames and the semantics of understanding. *Quaderni di Semantica*, 6:222-255.
- Giachin, E. and S. McGlashan. 1997. Spoken Language Dialogue Systems. In Steve Young and Gerrit Bloothoof (Eds.) *Corpus-Based Methods in Language and Speech Processing*, pp. 69-117, Kluwer, Dordrecht, The Netherlands.
- Grefenstette, G. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer, Dordrecht, The Netherlands.
- Hudson, R. 1984. *Word Grammar*. Blackwell, Oxford, England.
- Karlsson, F., Voutilainen, A., Heikkilä, J. and A. Anttila. 1995. *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*. de Gruyter, Berlin, Germany.
- Klein, M., Bernsen, N. O., Davies, S., Dybkjaer, L., Garrido, J., Kasch, H., Mengel, A., Pirrelli, V., Poesio, M., Quazza, S. and C. Soria. 1998. Supported Coding Schemes. MATE Technical Report D1.1.
- Mengel, A., Dybkjaer, L., Garrido, J., Heid, U., Klein, M., Pirrelli, V., Poesio, M., Quazza, S., Schiffrin, A., and C. Soria. 1999. MATE Dialogue Annotation Guidelines. MATE Technical Report D2.1.
- Lenci, A., Montemagni, S., Pirrelli, V. and Soria C. 1999. FAME: a Functional Annotation Meta-scheme for Multimodal and Multi-lingual Parsing Evaluation. *Proceeding of the ACL99 Workshop on Computer-Mediated Language Assessment and Evaluation in NLP, Univ. of Maryland, June 1999*.
- Lenci, A., Montemagni, S., Pirrelli, V., Soria C., Netter K. and M. Rajman. 1999. Corpora for Evaluation. *ELSE Deliverable D5*.
- Lin, D. 1998. A dependency based method for evaluating broad-coverage parsers. *Natural Language Engineering*, 4(2):97-114.
- Marcus, M., Kim, G., Marcinkiewicz, M. A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K. and B. Schasberger. 1994. The Penn Treebank: annotating predicate argument structure. *Proceedings of DARPA 1994*.
- Peckam, J. 1991. Speech understanding and dialogue over the telephone, An overview of the ESPRIT SUNDIAL project. In *Proceedings DARPA Speech and Natural Language Workshop*.
- Sampson, G. 1998. A proposal for improving the measurement of parse accuracy. Unpublished manuscript.
- Sanfilippo, A., Barnett, R., Calzolari, N., Flores, S., Hellwig, P., Kahrel, P., Leech, G., Melero, M., Montemagni, S., Odijk, J., Pirrelli, V., Teufel, S., Villegas, M. and Zaysser, L. 1996. Subcategorization Standards. Report of the EAGLES Lexicon/Syntax Group. *SHARP Laboratories of Europe, Oxford (UK)*.