COLLECTION
ÆGYPTIACA LEODIENSIA

La collection *Ægyptiaca Leodiensia* — dirigée par Jean Winand, Dimitri Laboury et Stéphane Polis — a pour vocation de publier des travaux d'égyptologie dans les domaines les plus divers. Elle accueille en son sein des monographies ainsi que des volumes collectifs thématiques.

This volume represents the outcome of the meeting of the Computer Working Group of the International Association of Egyptologists (Informatique & Égyptologie) held in Liège in 2010 (6-8 July) under the auspices of the Ramses Project. The papers are based on presentations given during this meeting and have been selected in order to cover three main thematic areas of research at the intersection of Egyptology and Information Technology: (1) the construction, management and use of Ancient Egyptian annotated corpora; (2) the problems linked to hieroglyphic encoding; (3) the development of databases in the fields of art history, philology and prosopography. The contributions offer an up-to-date state of the art, discuss the most promising avenues for future research, developments and implementation, and suggest solutions to longstanding issues in the field.
Two general trends characterize the projects laid out here: the desire for online accessibility made available to the widest possible audience; and the search for standardization and interoperability. The efforts in these directions are admittedly of paramount importance for the future of Egyptological research in general. Indeed, for the present and increasingly for the future, one cannot overemphasize the (empirical and methodological) impact of a generalized access to structured data of the highest possible quality that can be browsed and exchanged without loss of information.

Stéphane POLIS is Research Associate at the National Fund for Scientific Research (Belgium). His fields of research are Ancient Egyptian linguistics and Late Egyptian philology and grammar. His work focuses on language variation and language change in Ancient Egyptian, with a special interest for the functional domain of modality. He supervises the development of the Ramses Project at the University of Liège with Jean Winand.

Jean WINAND is professor ordinarius at the University of Liège, and currently Dean of the Faculty of Philosophy and Letters. He specializes in texts and languages of ancient Egypt. His major publications include *Études de néo-égyptien. La morphologie verbale* (1992); *Grammaire raisonnée de l'Égyptien classique* (1999, with Michel Malaise); *Temps et Aspect en égyptien. Une approche sémantique* (2006). He launched the Ramses Project in 2006, which he supervises with Stéphane Polis.

PRESSES UNIVERSITAIRES DE LIÈGE

ISBN : 978-2-87562-016-3

ÆGYPTIACA LEODIENSIA

9

Texts, Languages & Information Technology in Egyptology
Stéphane POLIS – Jean WINAND

COLLECTION
ÆGYPTIACA LEODIENSIA

Texts, Languages & Information Technology in Egyptology

Stéphane POLIS — Jean WINAND
With the collaboration of Todd GILLEN

Presses Universitaires de Liège

Texts, Languages & Information

Technology in Egyptology

Dépôt légal D/2012/12.839/17
ISBN 978-2-87562-016-3

Place du 20-Août, 7
B-4000 Liège (Belgique)
http://www.presses.ulg.ac.be

Imprimé en Belgique

Collection *Ægyptiaca Leodiensia* 9

Texts, Languages & Information Technology in Egyptology

Selected papers from the meeting of the Computer Working Group
of the International Association of Egyptologists
(Informatique & Égyptologie), Liège, 6-8 July 2010

Stéphane POLIS & Jean WINAND (eds.)

With the collaboration of Todd GILLEN

Presses Universitaires de Liège
2013

Table of Contents

Building a Construction-Based Treebank of Late Egyptian*

The Syntactic Layer in Ramses

Stéphane POLIS & Serge ROSMORDUC

F.R.S.-FNRS (Université de Liège) – Conservatoire National des Arts et Métiers (Paris)

1. INTRODUCTION

The ultimate purpose of the Ramses Project is to provide scholars with a fully annotated corpus of Late Egyptian texts.[1] Unsurprisingly, the annotation of the corpus with syntactic structure came as the last significant development of the project.[2] This part of the software — called SyntaxEditor — has been (and still remains to some extent) an actual challenge in its own right; indeed, several requirements had to be handled simultaneously as regards (1) the syntactic formalism to be implemented and the related representational format, (2) the specificities of the annotation scheme to be developed, and (3) the ergonomic demands of annotation. These needs can be summarized as follows:

(1) From a linguistic viewpoint, the syntactic formalism had to be as theory neutral as possible — i.e. free from theoretical idiosyncrasies, with the evident goal of ultimately allowing scholars from diverse backgrounds to retrieve data on Late Egyptian syntax profitably; at the same time, the generic nature of this formalism could not lead to a simplification of the syntactic annotation: the diversity of the syntactic facts found in the Late Egyptian corpus had to be handled and annotated in its complexity. Besides the traditional specification of "groups" or constructions[3] and accepted part-whole structure hierarchies of constructions, syntactic functions (or roles) — and, crucially, not abstract syntactic relations — have to be explicitly defined for any element according to the construction it belongs to.[4] Furthermore, the corpus has to be annotated not only for skeletal syntactic structure (the so-called "bracketing" task): we wanted a representational model that handles discontinuous constituents on the one hand and that allows, on the other hand, for the annotation of "horizontal" relations between

* Serge Rosmorduc is responsible for all IT conception and development to date. The theoretical principles that lie behind the implementation of the SyntaxEditor have much benefited from the expertise of Eitan Grossman who held a post-doctoral position within the Ramses project in 2009-2010. Our thanks are due to Todd Gillen for proofreading the English.

1. See Polis, Honnay & Winand in the current volume.
2. As far as the implementation of the annotating tools is concerned, at least. Indeed, the Web application that will give the community of both linguists and Egyptologists access to the corpus is still to be developed.
3. In this paper, the label "group" is understood generically as referring to any kind of construction at the lexical, idiomatic, phrasal, clausal and textual levels. As "groups", constructions can be compared to constituents in constituency-based formalisms and to a head-dependent(s) relation in dependency-based formalism.
4. On this basic principle, see *inter alii* Croft 2001: 5.

constructions or elements of constructions (the domains related to textual cohesion — e.g. co-indexation of pronouns and nouns phrase, co-reference, etc.[5] — and information structure).

(2) The SyntaxEditor software had to be abstract enough to allow for an Egyptological team to perform by itself — i.e. in a self-standing (and intelligible) definition file — different types of changes within an annotation scheme that is likely to develop considerably as new constructions are encountered. The goal is indeed not to write *a priori* a grammar in the annotation scheme, but to facilitate the later writing of a grammar based on the documented constructions in the corpus. At the same time, the annotation scheme had to be developed in a way to constrain somehow the annotating process (in order to ensure the coherence of the encoding) and to control beforehand (and to facilitate thereby) the annotators' work.

(3) As for the ergonomics, unlike in modern language corpora where the bracketing task is usually performed with (deterministic) parsers and where, ideally, the parser's output is hand-corrected by annotators in a second step,[6] no parser is available from scratch for Late Egyptian texts. If such a tool is to be part of long-term plans,[7] the ergonomics of the annotation tools had to be designed in a way that would (a) take advantage of the pre-existing annotations (part-of-speech tagging, lemmatization, morphological analysis and translation) and (b) make the chunking of sentences into constructions and the analysis of functions (or roles) of the grouped elements and constructions as quick and straightforward as possible.

This paper addresses these issues by reviewing the current state of the annotation tool, i.e. the SyntaxEditor. In a first section (§2), we argue in favor of a construction-based syntactic formalism, i.e. a formalism that is neither framed in a constituency- nor in a dependency-based model, and aims at encoding the widest varieties[8] of syntactic constructions without positing (in advance) abstract syntactic functions. In the following section (§3), we introduce the evolutionary annotation scheme: it is written in a Syntax Description Language (SDL) and can easily be modified by the annotators any time a previously unattested construction pops up in a texts (without further programming work involved). In the next section (§4), we succinctly describe the steps for manually annotating a text with syntactic structures and present the capabilities of the SyntaxEditor. Finally (§4), we broach future developments: the search engine as well as an interactive parser sensitive to mark-up data.

2. THE SYNTACTIC FORMALISMS

At present, the Ramses corpus contains a reasonable number of lemmatized and morphologically annotated texts (as of late 2011, ca. 1 400 texts for a total of 300 000 words). We may therefore proceed with the last significant step of the first phase of the project as regards the annotation procedure, i.e. providing the corpus with a full syntactic analysis.

While the software developed for encoding the lemmatised texts was designed as a one-dimensional linear system — the texts were analysed word by word, each word being assigned a spelling, a lemma and an inflexion —, the syntactic layer calls for some kind of tree editor: in any kind of approach to syntax, the elements of sentences are at some point hierarchically ordered in a two dimensional graph.

5. For the domain of coreference information in corpus linguistics, see e.g. the Potsdam Commentary Corpus (Stede 2004; other examples cited in Dipper & Götze 2005) or the Spanish CESS-ECE corpus (Recasens *et al.* 2008).

6. As for example in the Penn Treebank (see Marcus *et al.* 1993: 313-314).

7. See §5 for an alternative view.

8. It is worth noting that the syntactic encoding of syntax in Ramsès is *emic* in the sense that it does not take into account allographies and allomorphies that are dealt with in the Text/LexiconEditors, see Polis, Honnay & Winand in the current volume.

2.1. *Dependency-based grammars and phrase structure grammars*

Basically, we had the choice between two overwhelmingly dominant families of formalism in corpus linguistics: dependency grammars[9] and the phrase structure grammars (or constituency grammars).[10]

In dependency grammars — that go back, in modern times, to the *Éléments de syntaxe structurale* of Lucien Tesnière[11] —, there are no phrasal nodes. Everything is modelled as asymmetrical dependency relationships between words: heads (governors of each structure) and dependents. Accordingly, a sentence such as  *ḏd=w m rꜣ wꜥ* "they said with one mouth" could be represented as:

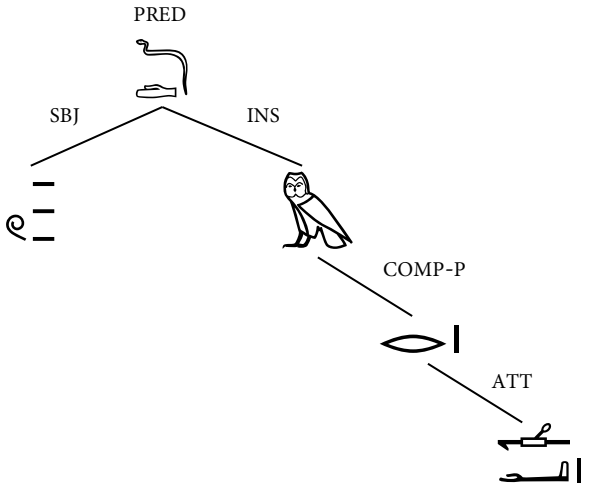

Figure 1. Basic dependency analysis

In our opinion, two obvious advantages of the dependency-based formalism are (1) that, unlike in constituency formalism, syntactic functions are always spelled out explicitly through the specification of the relation types, and (2) that valency patterns or argument structures (especially of verbal predication) are more directly retrievable.

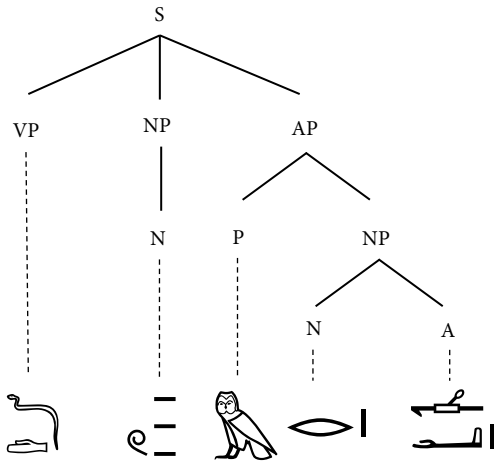

Figure 2. Basic phrase structure analysis

The phrase structure grammar, on the other hand, would group the words in phrases, themselves grouped in higher level phrases, up to the sentence level. Instead of a one-to-one relationship between

9. An example is the Prague Dependency Treebank for Czech (see Hajic 1999), admittedly a pioneer in this domain.
10. Much of the work on Treebanks focuses primarily on modern languages. Treebanks of ancient text languages remain rather uncommon; see however McGillivray *et al.* (2009) and Haug *et al.* (2009) as well as references 8-14 cited in Bamman & Crane 2011.
11. More recently, see inter alii Mel'čuk 1988; Polguère & Mel'čuk 2009.

the words of a sentence and the nodes in the syntactic tree, there is a one-to-one-or-more correspondence between elements of a sentence and syntactic nodes, as exemplified in Fig. 2.

A definite advantage of constituency-based formalisms — at least for an ancient language like Late Egyptian whose texts are often fragmentary — is that it is easy to define a group without describing its entire structure (for instance, to say that *m r3 wꜥ* "with one mouth" in Fig. 2 is an adverbial phrase without analyzing further its constituency). In dependency grammars, the equivalent would be to create temporary unlabelled links between the three words, which is obviously less convenient, especially for complex phrases.

2.2. *A third way: A construction-based Treebank*

Both approaches have obvious pros and cons depending, first, on each scholar's theoretical assumptions regarding syntactic structures, of course, and — more practically — on the language being syntactically annotated.[12]

Given the fact that our basic requirements were (a) to be as theory neutral as possible while taking into account the diversity of the syntactic facts found in the Late Egyptian corpus, (b) to make the annotation of functions explicit[13] in each individual syntactic environment, and (c) to allow for annotations of horizontal relations between elements (graph relations), we tried to combine the advantages of both dependency-based grammars and phrase structure grammars and to overcome what we consider to be their respective shortcomings by developing a simple and intuitive *construction-based* formalism.[14] It should be stressed here that this is much in agreement with the practice in Egyptological linguistics that has traditionally been "Construction Grammar" *avant la lettre*, e.g. with the identification of numerous *patterns*. By doing so, we do not exclude output formats such as dependency or phrase structure graphs, but we envision them as two possible export formats of a more generic formalism[15] that allows describing the syntactic structures in their complexity at the level of surface forms, i.e. not at the level of posited deep structure.[16]

Based on some fundamental tenets of Construction Grammar[17] (CxG) and after a close look at innovative tools such as EMDROS[18] and Notabene,[19] we decided to use the following formalism: the analysis will consist of a set of syntactic constructions, called "groups" in the annotation scheme. A group represents any syntactic construct, from complex words,[20] idioms and simple phrases (like noun or adjectival phrases), to sentences (with various types of predications — including, crucially for a language like Late Egyptian, non-verbal predication patterns — and argument structure schemata) and even paragraphs or entire texts. A group in the formalism (i.e. a construction from a linguistic point of view) has the following properties:

12. Dependency grammars, for example, prove to be easier to implement for languages involving a relatively free word order.
13. In this respect, see the remarks in Blache 2000: 85.
14. It turned out that the annotation scheme of the TIGER Treebank for German (see Brants *et al.* 2002) is actually close to principles advocated for in the present paper.
15. On the links and transformations between dependency-based and constituency-based graphs, see Robinson 1970 and Mazziotta 2010b: 144, which was our prime source of inspiration in this respect.
16. Therefore, we do not have to posit (frightening) null elements in the tagset.
17. See especially Croft 2001; Goldberg 1995, 2006 (with the literature cited p. 18-19). The surface generalizations or the "what you see is what you get" approach to syntactic form that is adopted in CxG is particularly worth mentioning this context, for it does not derive one construction from another and it avoids positing zeroes in the syntactic analysis.
18. See Petersen 2004.
19. See Mazziotta 2010b.
20. The morphemic and word levels, although an obvious part of any analyzable construction is dealt with at the level of the LexiconEditor.

(1) A group may contain one or several basic elements and/or other groups. The functions of children elements and groups depend on the type of parent construction. We see, for example, no point in assuming categories like 'subject' across clause types: such categories must be the object of study based on the annotated data. Therefore, each construction type has its own features and syntactic function. For instance, the S function or role, i.e. 'intransitive subject' is only possible for an element or construction that is part of a higher level construction identified as an intransitive construction.

(2) A group can have various attributes (meant to capture and annotate the combination of different constructions in a single group For instance, an example such as *in wn di=f is.t ḫ3rw* "does he have a crew of Syrians" [*LES* 67,3-4] is an existential-possessive construction that has the attribute 'question construction').

(3) A group need not be continuous; it can be discontinuous.

Moreover, not all links between groups are of a hierarchical parent/child nature: other "horizontal" links are possible in the formalism (graph and not tree type) in order to represent inter alia the phenomena related to textual cohesion and information structure, such as anaphoric relations.

This formalism is admittedly very loose. The specification of the function of a phrasal construction in a sentence, for instance, is not mandatory. That way, partial analysis can be built, which — as already stated above — was a basic requirement for a text language in which many documents are fragmentary. Ideally, we would subsequently run some checking software on the analysis so as to detect what has been left under-specified.

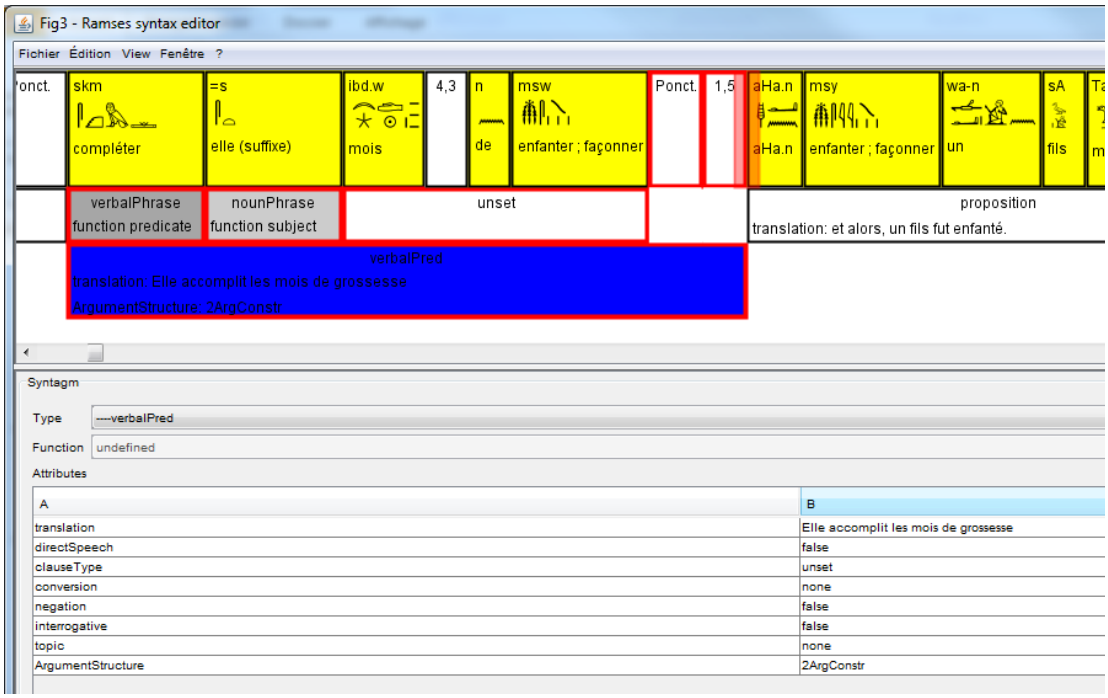

Figure 3. Example of simplified annotation

Fig. 3 shows the current interface of the SyntaxEditor when editing the sentence *skm=s ibd.w n msw* litt. "she passed the months of pregnancy" from the *Doomed Prince*. It features a number of interesting capabilities of the system. One can notice that the content of the noun phrase construction *ibd.w n msw.t* "months of pregnancy" has not yet been analyzed. This can be achieved at any point: it is possible to build a bottom-up analysis by grouping the words in larger constructions, and combining phrasal constructions into complex clausal constructions, then into whole sentences and possibly larger units. At the same time, a top-down analysis is also possible, first chunking the sentence-level

constructions, then the main clausal and phrasal constructions, down to individual words. Actual practices of annotators are of course likely to mix the two approaches.

It is worth noticing however that the function of any element or construction directly depends on the larger construction to which it belongs: the annotation of the functions or roles of individual elements or constructions can only be achieved top-down. As shown Fig. 4, the function of the imperative *imy* "give" can be annotated only after the independent main clause type has been defined as a verbal predication and the only available option, in this case, is to annotate it with the "predicate" function.

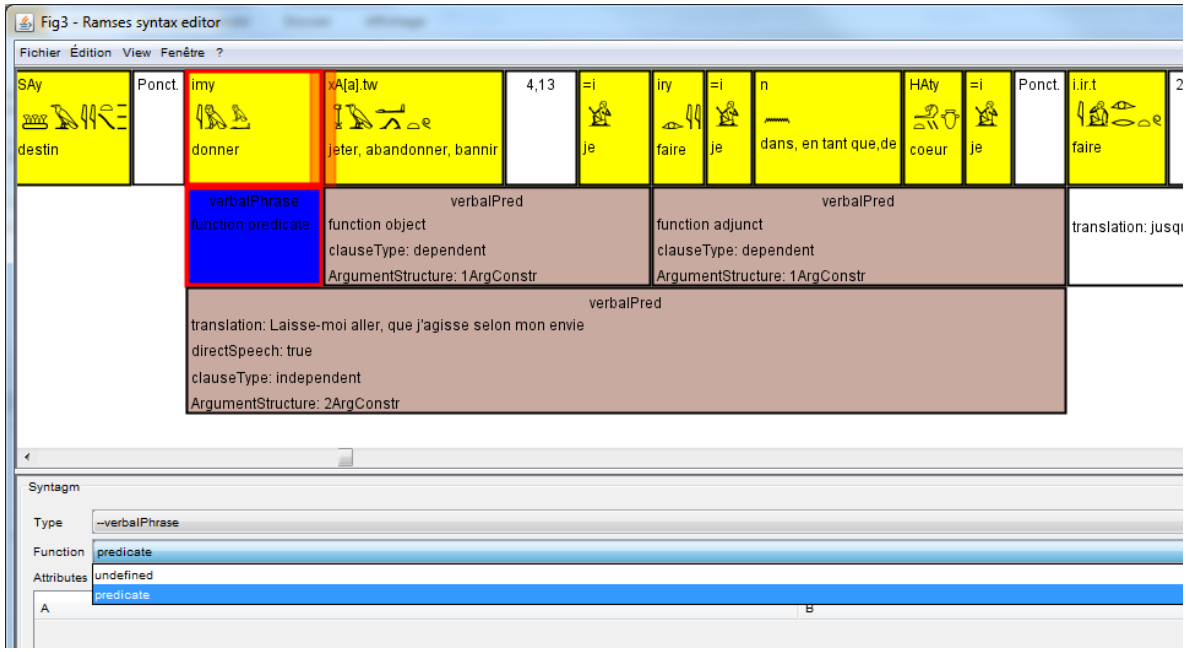

Figure 4. Annotation of function for a verbal phrase

Coming back to Fig. 3, it also features a variety of attributes attached to the groups. The currently selected construction, a verbal predication, is outlined in red and specifies its children elements. The construction itself has attributes of different kinds: the translation is a simple free-text attribute ("elle accomplit les mois de grossesse"). Then, we have a number of boolean attributes (usually with three possible values, 'true', 'false'/'none', and 'unset'). And finally, the ArgumentStructure attribute has a value assigned from a list (here, '2ArgConstr').

In Fig. 5, one can see a discontinuous phrase.[21] The negation *bn* … *iwn3* is considered as one element only. The groups are actually considered to be sets of elements, not spans of text (although, of course, this is the most frequent case).

21. This example of discontinuous phrase is meant as an illustration and admittedly disputable; *bn* and *iwn3* are — most probably — still two independent phrases in Late Egyptian, see Winand 1997.

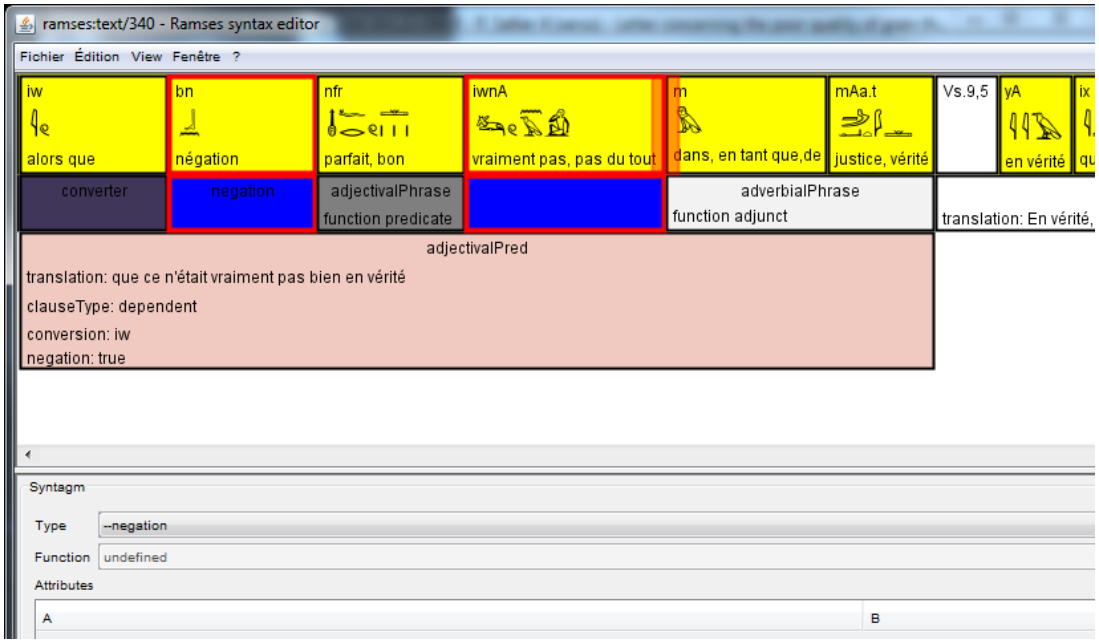

Figure 5. Discontinuous negation

3. EVOLUTIONARY ANNOTATION SCHEME AND SYNTAX DESCRIPTION LANGUAGE

3.1. *A brief look in the rear-view mirror*

When the project was started in 2006, the immediate priority was to allow text encoding, even on an elementary level. Hence, the first versions of the software (a "TextEditor" coupled to a "LexiconEditor") were developed to allow the quick creation of a fully lemmatized corpus, with complete information about word spellings and inflexions, but without any syntactic grouping.

Due to the high variability of Late Egyptian orthography, and to the time-consuming nature of hieroglyphic text encoding, particular attention was paid to providing an efficient interface. As a result, we started our work by designing the lexicon structure and the corresponding software (LexiconEditor). The text database itself (TextEditor), containing the lemmatised texts, was created a little later.

The lexicon contains three kinds of entities: lemma, inflexions, and spellings. A lemma corresponds more or less to a main dictionary entry. The types of possible inflexions depend on grammatical categories or parts-of-speech that are defined at the level of the lemma; these inflexions are hierarchically subordinated to the lemmata in the LexiconEditor. For instance, we have a lemma for the verb  *ḥtp*, "to be satisfied"; this lemma is attested in the corpus under different verbal inflexions (verbal morphology): e.g. infinitive, imperative, perfective, subjunctive, participle, etc. Due to the non-normative and defective nature of the Ancient Egyptian graphemic system, the spellings can be shared between one or more inflexions of different lemmata. For instance, the spelling  *bꜣk* is connected with several inflections of the verb *bꜣk* "to work" (infinitive, old perfective, subjunctive and participle) as well as with the etymologically related substantive *bꜣkw* "task, labour".

The various problems we experienced when developing the LexiconEditor (and, to a lesser extent, the TextEditor) point out a number of desirable features for the kind of software we were developing. The main problem we faced is that, besides its general structure, such a database tends to change a lot in its early beginning with respect to its tagset (labels used, number and structure of the parts-of-speech tags, types and values of the attributes for annotating the lemmata and inflexions). Accordingly, the changes and evolutions needed ranged from small modifications in category labels to significant structural changes (removal or creation of new types of inflexions or parts-of-speech for

example). Consequently, we had to write ad-hoc software on several occasions in order to carry out those modifications.

For the syntactic analysis, we wanted to avoid both falling into the same traps as well as writing the grammar before annotating the corpus, as this part of the project is intrinsically an experimental enterprise (we do not know for sure what categories will be used in the final version of the database). At the same time, we could take for granted that important changes would also occur in the syntactic tagset during the annotation of texts. Those changes would undoubtedly involve the addition of new analytical categories and the subtraction of some existing ones. Furthermore, when this occurs, the annotation on the previously analysed texts must not be lost, i.e. the software must be able to handle multiple annotation schemes.

For all those reasons, it was quite obvious that a great level of abstraction in software would be a major improvement for the SyntaxEditor annotation tool: the various categories which can be used in a syntactic analysis had to be explicitly defined in a self-standing annotation scheme that would allow the system users, i.e. the members of the Egyptological team, to perform changes within the tagset and basic grammar on their own — without any programming work — by using an intelligible Syntax Description Language (SDL) that they could easily modify and update.

3.2. *Control over the analysis structure*

Ramses is a collective, but centralized enterprise. Accordingly, it wouldn't be reasonable to allow each annotator to create the categories and attributes he needs on the fly. It would otherwise result in a complete inconsistency of the annotated data. Thus it is necessary to have some control over what constructions are available, and what attributes they can have. On the other hand, as stated in §3.1, it is equally inevitable that new categories will be needed at some point, and that significant changes, such as splitting a category in two, or merging two categories into one, will occur.

Faced with this problem, we decided to build on ideas present in Mazziota's *Notabene* (see 2010a and 2010b), and inspired by the semantic web, where the frame of each analysis is explicitly described. Instead of using Web Ontologies, we decided however, to create an ad-hoc language, specialised in the description of syntactic formalism, a Syntax Description Language.

3.3. *The Syntax Description Language (SDL)*

The formalism, built in the "annotation scheme", describes a kind of very simple "loose grammar" of the language, based on the basic principle that the grammar of any language is made up of taxonomic networks of families of constructions. The resulting description is saved in the database, and is identified by a name. An analysis is created using one particular annotation scheme (although it is possible to change it afterwards). Figure 6 illustrates a simplified and highly incomplete annotation system.

```
ANNOTATION SCHEME "Scheme_CxG_Test"

TYPE definiteness ENUM unset defined undefined doubtful ENDTYPE

GROUP construction
   ATTR comment TEXT * ENDATTR
ENDGROUP

// Phrases

GROUP phrasalConstr EXTENDS construction
ENDGROUP

GROUP nounPhrase EXTENDS phrasalConstr
   ATTR defined definiteness ONE unset ENDATTR
ENDGROUP
```

```
// Clauses
GROUP clausalConstr EXTENDS construction
   CHILD adjunct CHILDTYPE adverbialPhrase ENDCHILD
ENDGROUP
```

Figure 6. Simplified annotation scheme

Each annotation scheme name is given at the beginning of the file, in the present example `Scheme_CxG_Test` (it can be more or less anything).

Then, there are two kinds of elements in the SDL. First, one can describe `types`, which are used as attribute values. Here, we define the `definiteness` type, with four possible values: `unset`, `defined`, `undefined` and `doubtful`. The most important kind of element is the `group`. It describes what syntactic construction may exist in our description: any `group` is described by a name (which will be its label in the SyntaxEditor), and by its possible attributes. For instance, groups of type `nounPhrase` have an attribute called `defined`, of type `definiteness` (the `type` created just above). Other possible types are BOOLEAN for yes/no attributes, and TEXT for attributes whose value is a free text, like translations.

One can further specify whether an attribute is mandatory for a particular group, and indicate how many times (once or repeatedly) the attribute can be found. For instance, the `comment` attribute in `group` can be repeated (that is the meaning of the asterisk * character).

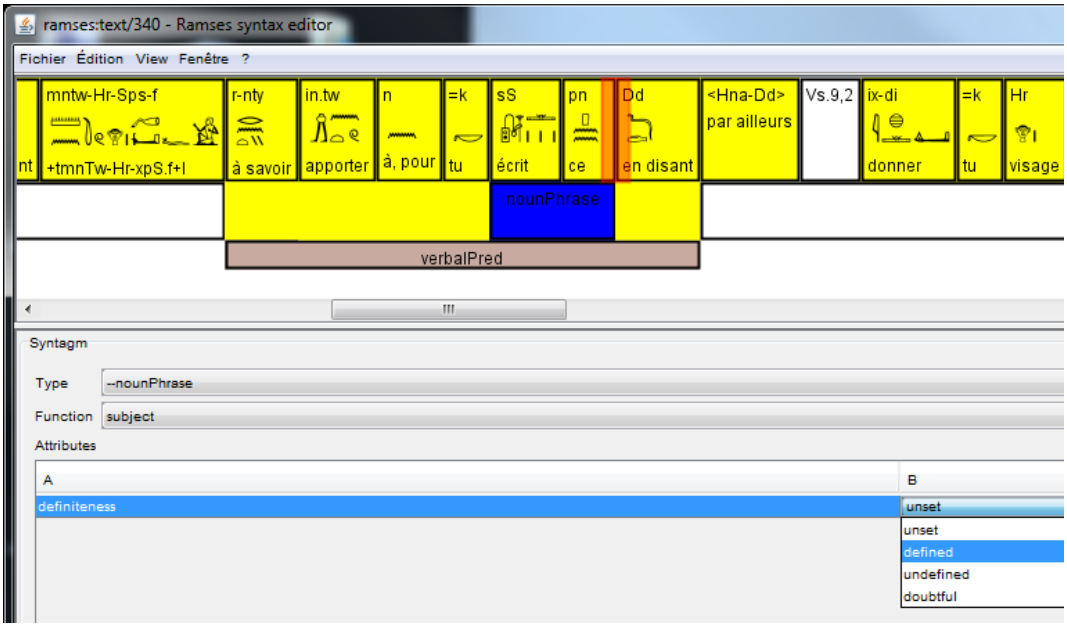

Figure 7. Use of attributes

As it is likely that some constructions will have common attributes, we have implemented an "inheritance" system between groups. A group type can be declared to "extend" another group, in which case it will receive the same attributes (and possibly more). For instance, `clausalConstr` extends `construction` in Fig. 6, which means that all clausal constructions inherit the `comment` attribute defined for `group`. If we were to define sub-types of clauses, they would all receive the characteristics of `clausalConstr`, in addition to their own.

As exemplified in Fig. 7, the attributes definition is used to fill the list of possible attributes values in the SyntaxEditor.

3.4. *Children and relations*

In order to define explicitly the functions that any element can fulfil in the construction it belongs to, the SDL allows possible syntactic subordinates of a group to be declared as `children`. `Children` describe both a syntactic function (for instance, one of the children of a verbal predication can be its subject), and the kind(s) of group(s) which can fill this function.

A definition like:

```
GROUP monoclausalConstr EXTENDS clausalConstruction
   CHILD adjunct CHILDTYPE adverbialPhrase ENDCHILD
ENDGROUP
```

defines one kind of child for `monoclausalConstr` ("mono-clausal construction"): the adjunct, which can only be adverbial phrases in this example. When editing a group's attributes in the SyntaxEditor, a list proposes all the functions the group can have relative to its parent.

It is important to note that inheritance can be used here as well. It is possible to define verbal predicative patterns as follows:

```
GROUP verbalPred EXTENDS monoclausalConstr
   CHILD subject CHILDTYPE nounPhrase ENDCHILD
   CHILD predicate CHILDTYPE verbalPhrase ENDCHILD
ENDGROUP
```

In this case, the verbal predicative pattern is defined as a kind of mono-clausal construction and can have three kinds of children: adjuncts, inherited from mono-clausal construction, and a subject as well as a predicate, which are specific to the verbal predicative pattern.

In the present state of the SDL, an element has only one parent, which means that the syntactic analysis is practically limited to trees. Now, as was mentioned in the introduction, there are many relationships between phrases that are not of a strictly hierarchical nature. One can think of anaphoric relations as a typical example. (In this domain, there is also an interest among the members of the Ramses project in the interaction between narration and discourse in the texts: we want to be able to link related instances of indirect speech, for example). For all those reasons, we need a second type of link between groups that enables us to annotate graph structures, and we have termed this link `relation`. Relations have the same kind of definition as children do; this time, however, there is no embedding constraint at all. At the moment, there is no support in the SyntaxEditor for creating actual relations, but they will probably be materialised by arrows between the two related groups.

4. ERGONOMICS OF THE USER INTERFACE

Considering that a large part of the treebank will be created manually in a first step, we needed a powerful annotation tool: this user interface, called "SyntaxEditor", had to follow some ergonomic principles in order to minimize the need for training and speed up the annotation work. In this section, we will briefly describe the steps for manually annotating with syntactic structures an (admittedly quite simple) independent clause in a text; thus we illustrate further (see Fig. 3-5; 7) the capabilities of the SyntaxEditor.

The text to be annotated is first imported into the SyntaxEditor as a sequence of tokens. Fig. 8 shows clearly that the other levels of annotation are easily accessible in the SyntaxEditor, simply by clicking on the token one wishes to have information about; here, the verb *ʿḳ* "to enter" has a spelling and a defined inflexion (subjunctive).

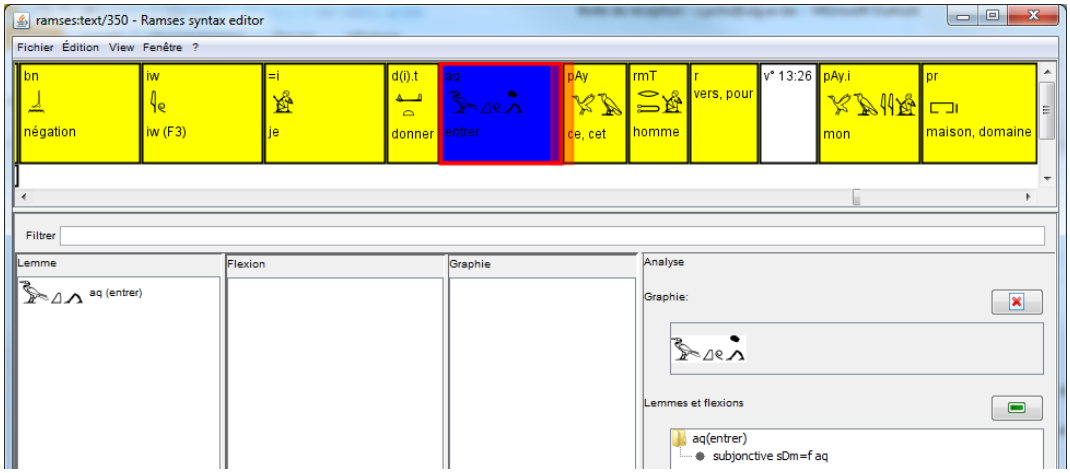

Figure 8. Other levels of annotation in the SyntaxEditor: spelling, lemma and inflexion

In a second step, the annotator selects the elements to be grouped together or the elements to be chunked in the clause; both bottom-up and top-down approaches are possible. The annotator can define the construction types either directly or afterwards. The SyntaxEditor allows for much flexibility in this respect: as shown in Fig. 9, the main constructions of the sentence have been identified, but only two of them actually received a type; the other ones have simply been tagged as generic groups.

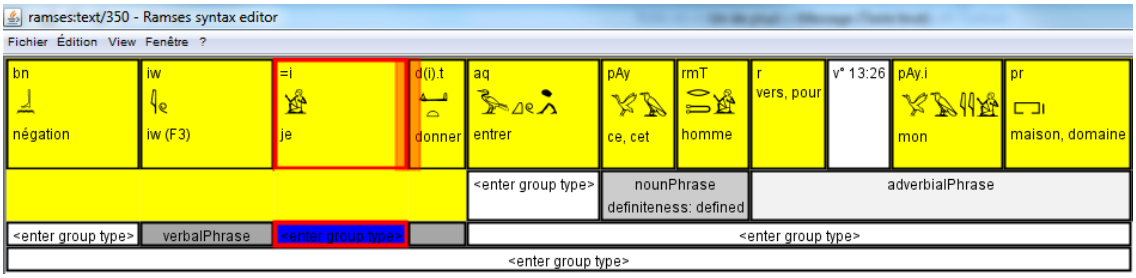

Figure 9. Creation of groups and identification of group types

Once all the top-level constructions have been annotated with types and related attributes, the syntactic analysis appears as in Fig. 10. Two points are worth mentioning here: (a) if the value of the attribute of any construction is the default one, it is not actually displayed in the box summarizing the data;[22] (b) in order to avoid duplicating any kind of information, the verbal phrase *iw* … *di.t* "(I) will cause" is analysed as a single discontinuous group — the information about the status of the conjugation base *iw* and the infinitive *di.t* are already annotated at the level of the TextEditor.

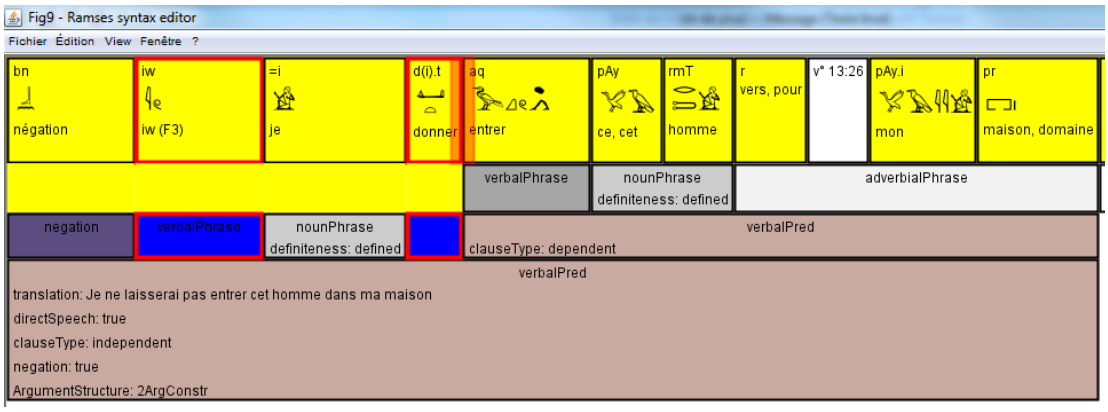

Figure 10. A clause with the top-level construction identified

22. In the example of Fig. 10, there is no data regarding the interrogative status of the main clause construction, for its default value is "false".

The third step consists of the annotation of the functions of the constructions. As already explained (see §3.4), this step can only be achieved when the status of the higher level construction have been defined, for the only available functions in the SyntaxEditor for a given element are the ones that are defined to be acceptable for this type of element in a higher construction within the annotation scheme. In Fig. 11, we see that an unconverted dependent clause can only be annotated as a complement clause or as an adjunctive (consecutive) clause when it is an element of an independent main clause.

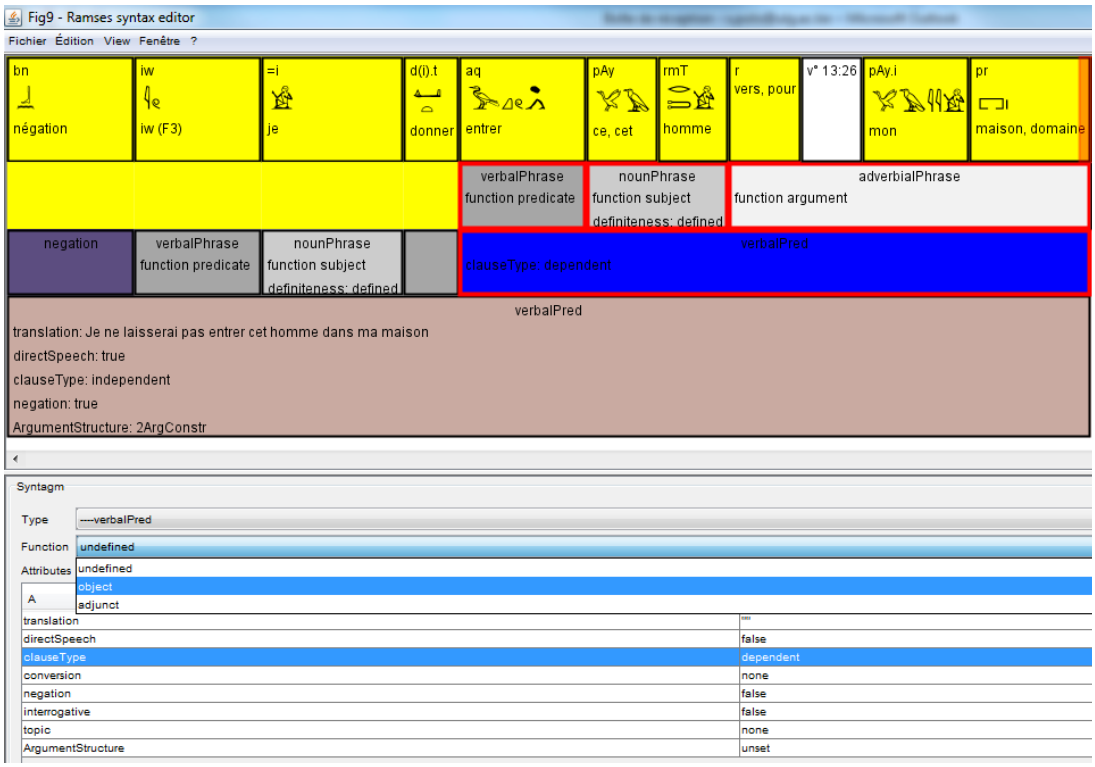

Figure 11. Annotation of syntactic roles

Finally, as illustrated in Fig. 12, the constructions can be flattened using a toggle function in order to limit the information to the types and functions, without displaying the data that concerns the attributes.

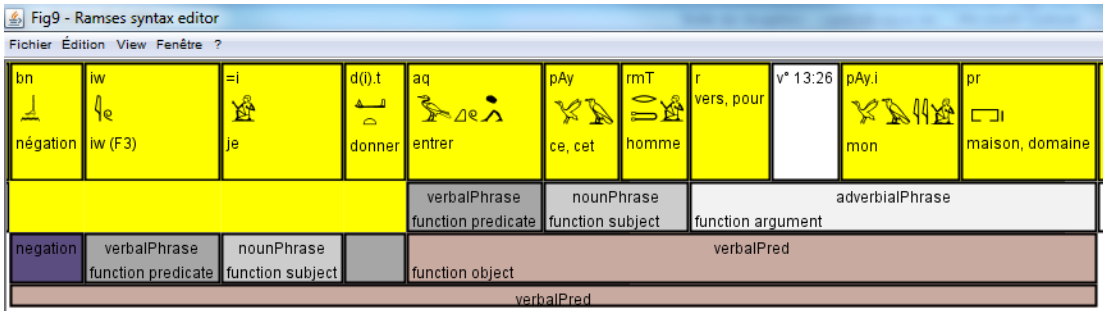

Figure 12. Annotation of syntactic roles

5. FUTURE DEVELOPMENTS

There are many points still on the to-do list for the SyntaxEditor. Some of them are rather mundane, but very important, such as the implementation of a versioning system, in order to track changes in the texts.

Other are more exciting, mostly those that include a natural language processing component. There are indeed several areas where the system should be cleverer. First, it would be important to be

able to define, in a reasonably simple way, validation rules, to detect problems in the analysis. Another important area of work is the developments of tools for updating "old" analysis: when a new annotation scheme becomes available, we will certainly want to transfer the previous work done with the old annotation scheme.[23] In some cases, it will be easy a task (e.g. when a category is common to both schemes), but some changes will be trickier, and others will require some manual editing (an important feature of the current software is that the old annotations are kept when changing the annotation scheme for a text).

Finally two more ambitious tasks are the implementation of a search engine and the development of a parser for Late Egyptian texts.

5.1. *Search engine*

Such a syntactically annotated corpus would evidently be useless without efficient search facilities. There are a number of existing tools which are suitable for searches in structured text databases. To name but a few: emdros, TIGERSearch, and the Intex family (Intex, Unitex and Nooj). Given the current state of Ramses, we will likely base our search engine on TIGERSearch,[24] which can be easily interfaced with Ramses as both are written in Java. It satisfies two conflicting requirements: it is easy to learn (close to grammar formalism) and its expressiveness is constrained in order to guarantee efficient query processing. It is of course too early to describe the engine, but it is possible to give an idea of the possibilities of existing systems. For instance, given a correctly structured database, TIGERSearch would allow queries like:

```
(#n1:[cat="nounPhrase" & defined="undefined"]
.*
#n2:[cat="relativeClause"])
& (#n2 >~antecedent #n1)
```

which would find all relative clauses whose antecedent is not defined. Note that such a query cannot be done on a corpus annotated only for part-of-speech and lemma. Syntactic annotation is definitely needed. TIGERSearch also provides a graphical interface to avoid the direct use of the query language, and probably makes simple queries easier to write.

5.2. *Interactive parser*

It has already been shown[25] that noun phrase constructions (at least the ones that do not include relative or participial clauses) can be parsed with a very high degree of accuracy using automata, which explains why we decided to focus first on the higher rank constructions (and not on the phrasal constructions).

Now, the most ambitious work in front of us in this domain is certainly the partial automation of the analysis, which is part of the PhD thesis on which Benjamin Martin Leon (University of Liège) is working. Much like the parser which has been developed in the framework of the TIGER Treebank,[26] we include in our future plans an "interactive parser": "[i]nteractive annotation is an efficient combination of automatic parsing and human annotation. Instead of having an automatic parser as preprocessor and a human annotator as postprocessor, the two steps are interwoven in our approach." Moreover, given the length of attestation of Late Egyptian linguistic data (more than five hundred years), the wealth of registers and the strong influence of the writing support on the spellings, the

23. In this respect, see the comparative procedure in Mazziotta (2010b: §2.3).

24. For the query language, see König & Lezius 2002a & 2002b.

25. See Benjamin Martin Leon's unpublished MA thesis: "Projet Ramsès: réalisation d'une bibliothèque de traitement à états finis" (Liège; 2008-2009).

26. See Brants *et al.* 2002: §3.1.

parser will have to take into account mark-up data on the corpus if one wishes to reach an acceptable degree of accuracy.

6. CONCLUSION

The SyntaxEditor inaugurates the next important step in the Ramses project. The decision — taken, to be honest, more out of necessity than on theoretical grounds — to start with a "simple" lemmatised corpus now allows us to start working on this level of annotation with a large database already tagged for lemma, inflexions and spelling. We envisage it as a solid foundation on which to build this next stage. In particular, having all these data means that we can test syntactic hypotheses more easily, and try statistical methods if needed.

From a linguistic viewpoint, we take seriously the assumption of Construction Grammar that *constructions* are the basic units of syntactic representation; consequently, we consider as a real possibility that the syntactic annotation will lead to generalizations concerning elements across constructions that are not congruent with the pre-existing (e.g. part-of-speech) categorization (as annotated in the TextEditor). This means that syntactic annotation will undoubtedly have a feed-back effect on the previous analyses, thereby avoiding the methodologically untenable position (see e.g. Hunston 2002: 93) of a priori defining a category such as part-of-speech.

BIBLIOGRAPHY

BAMMAN, David & Gregory CRANE. 2011. The Ancient Greek and Latin Dependency Treebanks, in: Caroline SPORLEDER, Antal VAN DEN BOSCH & Kalliopi ZERVANOU (eds.), *Language Technology for Cultural Heritage*, Springer, Foundations of Human Language Processing and Technology. http://nlp.perseus.tufts.edu/syntax/treebank/publications.html

BILGER, Mireille (ed.). 2000. *Corpus : Méthodologie et application linguistique*, Paris, Honoré Champion (= Bibliothèque de l'INaLF, Les français parlés – Textes et études 3).

BLACHE, Philippe. 2000. À quoi sert l'annotation syntaxique de corpus, in: BILGER (ed.), p. 82-94.

BRANTS, Sabine, Stefanie DIPPER, Silvia HANSEN, Wolfgang LEZIUS, George SMITH. 2002. The TIGER Treebank, in: *Proceedings of the Workshop on Treebanks and Linguistic Theories*, Sozopol, p. 24-41.

CROFT, William. 2001. *Radical Construction Grammar. Syntactic Theory in Typological Perspective*, Oxford, Oxford University Press.

DIPPER, Stefanie & Michael GÖTZE. 2005. Accessing heterogeneous linguistic data — Generic XML-based representation and flexible visualization, in: *Proceedings of the 2nd Language & Technology Conference*, Poznan, p. 206-210.

GOLDBERG, Adele E. 1995. *Constructions. A Construction Grammar Approach to Argument Structure*, Chicago, The university of Chicago Press.

—. 2006. *Constructions at Work. The Nature of Generalization in Language*, Oxford, Oxford University Press.

HAJIC, Jan. 1999. Building a syntactically annotated corpus: The Prague Dependency Treebank, in: Eva HAJIČOVÁ (ed.), *Issues of Valency and Meaning. Studies in Honour of Jarmila Panevova*, Prague, Charles University Press, p. 106-132.

HAUG, Dag *et al.* 2009. Computational and linguistic issues in designing a syntactically annotated parallel corpus of Indo-European languages", in: *Traitement Automatique des Langues* 50/2, p. 17-45.

HUNSTON, Susan. 2002. *Corpora in Applied Linguistics*, Cambridge, Cambridge University Press.

KÖNIG, Esther & Wolfgang LEZIUS. 2002a. *The TIGER language — A Description Language for Syntax Graphs. Part 1: User's Guidelines*, University of Stuttgart. (http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/papers/tigerLanguage.ps.gz)

—. 2002b. *The TIGER language — A Description Language for Syntax Graphs. Part 2: Formal Definition*, University of Stuttgart. http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/papers/tigerLangForm.ps.gz

LES = Alan H. GARDINER. 1932. *Late Egyptian Stories*, Brussels (= Bibliotheca Aegyptiaca 1).

MARCUS, Mitchell P., Beatrice SANTORINI & Mary Ann MARCINKIEWICZ. 1993. Building a large annotated corpus of English: The Penn Treebank, in: *Computational Linguistics* 19/2, p. 313-330.

MAZZIOTTA, Nicolas. 2010a. Logiciel NotaBene pour l'annotation linguistique. Annotations et conceptualisations multiples, in: *Recherches qualitatives. Hors série "Les actes"* 9, p. 83-94.

—. 2010b. Building the *Syntactic Reference Corpus of Medieval French* Using *NotaBene RDF Annotation Tool*, in: *Proceedings of the Fourth Linguistic Annotation Workshop (LAW IV)*, Uppsala, Association for Computational Linguistics, p. 142-146.

MCGILLIVRAY, Barbara, Marco PASSAROTTI & Ruffolo PAOLO. 2009. The *Index Thomisticus* Treebank project: Annotation, parsing and valency lexicon, in: *Traitement automatique des Langues* 50/2, p. 103-127.

MEL'ČUK, Igor. 1988. *Dependency Syntax: Theory and Practice*, University of New York Press, Albany.

PETERSEN, Ulrik. 2004. Emdros — A Text Database Engine for Analyzed or Annotated Text, in: *ACL, COLING 2004 Geneva, 20th International Conference on Computational Linguistics*, Volume 2, p. 1190-1193. [The application is available at http://emdros.org/]

POLGUÈRE, Alain & Igor MEL'ČUK (eds.). 2009. *Dependency in linguistic description*, Amsterdam-Philadelphia, John Benjamins.

POLIS, Stéphane, Anne-Claude HONNAY & Jean WINAND. Current volume. Building an annotated corpus of Late Egyptian. The Ramses Project: Review and perspectives.

RECASENS, Marta, M. Antònia MARTÍ & Mariona TAULÉ. 2007. Where anaphora and coreference meet. Annotation in the CESS-ECE corpus, in: *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP-2007)*, Borovets.

ROBINSON, Jane. 1970. Dependency structure and transformational rules, in: *Language* 46, p. 259-285.

STEDE, Manfred. 2004. The Potsdam Commentary Corpus, in: *Proceeding DiscAnnotation '04. Proceedings of the 2004 ACL Workshop on Discourse Annotation*, p. 96-102.

TESNIÈRE, Lucien. 1965. *Éléments de syntaxe structurale*, Paris, Klincksieck.

WINAND, Jean. 1997. La négation *bn … iwn3* en néo-égyptien, in: *Lingua Aegyptia* 5, 223-236.

Abstracts

Peter Dils & Frank Feder, The *Thesaurus Linguae Aegyptiae*. Review and Perspectives

The *Thesaurus Linguae Aegyptiae* (*TLA*) represents today the largest available database of Egyptian texts and, moreover, it is worldwide accessible on the Internet with free access. It combines a text corpus of Egyptian texts from nearly all periods of Egyptian history with an electronic lexicon. Both are linked to each other and are regularly updated. The TLA provides also access to the digitalized material on which the edition of the *Wörterbuch der aegyptischen Sprache* was based (slip archive). The text corpus and the lexicon can be searched in a number of ways and for different purposes; tools for statistical analysis are provided as well. As the *TLA* is a dynamically developing database system the text corpus and the lexicon will further be expanded, especially by adding the still lacking Coptic material of the Egyptian language, and by improving the research tools gradually.

Stéphane Polis, Anne-Claude Honnay & Jean Winand, Building an Annotated Corpus of Late Egyptian. The Ramses Project: Review and Perspectives

This paper reviews the experience of the Ramses Project in constructing a richly annotated corpus of Late Egyptian that consists of 300 000 words in 2011 (and is expected to grow up to more than 1 million words in coming years). During the first five years of the project, this corpus has been encoded in hieroglyphic script, translated in French or English and received annotations for part-of-speech information, lemmatization, and morphological analysis. The methodology and working tools that have been developed in order to build this corpus are here discussed and future developments are presented.

Stéphane Polis & Serge Rosmorduc, Building a Construction-Based Treebank of Late Egyptian. The Syntactic Layer in Ramses

This paper reports on the construction-based Treebank currently under development in the framework of the Ramses Project, which aims at building a multifaceted annotated corpus of Late Egyptian texts. We describe the specifications that have been implemented and we introduce the syntactic formalism and the related representation format that are used for the syntactic annotation. Furthermore, the annotation scheme is discussed with particular attention paid to its evolutionary nature. Finally, we explain the methods as well as the annotating tool, called *SyntaxEditor*; we conclude by

addressing the question of forthcoming developments, especially the search engine and a context-sensitive parser.

Stéphanie GOHY, Benjamin MARTIN LEON & Stéphane POLIS, Automated Text Categorization in a Dead Language. The Detection of Genres in Late Egyptian

This paper is a first step in applying machine learning methods typical of Automated Text Categorization (ATC) for Automatic Genre Identification (AGI) in Late Egyptian, a language written in either hieroglyphic or hieratic scripts that is found in documents from Ancient Egypt dating from ca. 1350-700 BCE. The study is divided into three parts. After a general introduction on AGI (§1), we introduce the levels of annotation that are integrated in the Ramses corpus and can be used when performing AGI on Late Egyptian (§2). In the following section (§3) we offer a brief survey of the types of features that have been discussed in the literature on AGI, before proceeding with three case studies where we apply supervised machine learning methods — namely the naïve Bayes classifier (§4.1), the Support Vector Machine (§4.2), and the Segment and Combine approach (§4.3) — to a selection of texts in the corpus. Their respective performances are tested using lexical, part-of-speech and inflectional features.

Mark-Jan NEDERHOF, Flexible Use of Text Annotations and Distance Learning

In this paper, we discuss a framework that allows independently created annotations of texts to be combined and presented as one unified interlinear format. Applications for distance learning are also considered. As proof-of-concept, we present PhilologEg, a tool that can be used to study an Ancient Egyptian hieroglyphic text in combination with any number of translations and grammatical annotations. The tool is a fully integrated system that runs on all major platforms.

Roberto GOZZOLI, Hieroglyphic Text Processors, Manuel de Codage, Unicode, and Lexicography

This paper gives an overview of the different software available to scholars working in the field of Egyptian language, with a special focus on hieroglyphic typesetting, Unicode and lexicographical databases that systematically encodes hieroglyphs. Various problems with the *Manuel de Codage* are discussed, as well as the need for a more active interaction between computers and Egyptology. A proposal for Egyptological software is given at the end of the paper.

Mark-Jan NEDERHOF, The Manuel de Codage Encoding of Hieroglyphs Impedes Development of Corpora

In this paper, we discuss the encoding of hieroglyphic text and argue that the set of requirements for an encoding scheme depend on the intended application. Our main claim is that if this application is the development of text corpora with long lifespans and diversity of use, then encoding schemes within the tradition of the Manuel de Codage are unsuitable.

Vincent EUVERTE & Christian ROY, Hieroglyphic Text Corpus. Towards Standardization

Sharing the heritage of Ancient Egyptian written production means facing numerous technical challenges. The goal of this paper is to build a preliminary inventory of these challenges and to propose some possible solutions. After a quick overview of the topics that are possible candidate to an international standardization, the paper focuses on two aspects. (1) The 'Multilingual Egyptological Thesaurus' (MET), initiated in 1996 by Dirk van der Plas, has not changed since 2003. It could be updated and expanded with minimal effort under the coordination of an official body such as the Center for Documentation of Cultural and Natural Heritage (CULTNAT). (2) The 'Manuel de Codage' (MdC) has not benefited from developments in computer science since the third edition was

published under the *Informatique & Égyptologie* mandate in 1988. Over time, each hieroglyphic software program has developed its own specific syntax to satisfy emerging needs, making it difficult for users to share ancient Egyptian texts. For these two topics, we will suggest a plan for improvement based on the Rosette Project's experience, though the input of the Egyptologists' community at large is appreciated to refine various concepts and identify the best route forward.

Christian MADER, Bernhard HASLHOFER & Niko POPITSCH, The MEKETREpository. A Collaborative Web Database for Middle Kingdom Scene Descriptions

Whilst representations, iconography and the development of scenes in private and royal tombs from the Old Kingdom have been studied extensively in the past, comparable research of Middle Kingdom (MK) representations and scene details is still underrepresented. The MEKETRE research project aims at closing this gap by systematic research of MK representations. In the course of this project, an online digital repository (the MEKETREpository) is being built that enables researchers to describe and annotate MK two-dimensional art at various levels of detail using images, free text, and controlled vocabularies. It also enables the collaborative development of semantic vocabularies for the description of these data. The MEKETREpository will publish the resulting data and vocabularies as Linked Data on the Web by utilizing Semantic Web technologies to enable their integration into other Linked Data sets such as DBpedia, Freebase or LIBRIS. The collected data is described using standardized and specialized vocabularies allowing for easy integration into existing databases and search engines. For the long-term preservation of the data, the MEKETREpository will make use of the University of Vienna's digital asset management system PHAIDRA. At its final stage the MEKETREpository will supply a platform that exposes collaboratively created, continuously evolving, and publicly available information about the MK on the Web.

Nathalie PRÉVÔT, The Digital Puzzle of the *talatat* from Karnak. A Tool for the Three-Dimensional Reconstruction of Theban Buildings from the Reign of Amenhotep IV

The revival of studies on the Atonist temples of Karnak (program of the French National Research Agency ATON-3D – ANR-08-BLAN-0202-01) required the implementation of an Information System dedicated to the Theban *talatat* that would also be accessible to the scientific community. This IS is associated with software which helps to reassemble the fragmented reliefs (a digital interactive puzzle), constituting a real tool for researchers and providing the knowledge needed to produce and validate hypotheses about the structures and dimensions of the buildings. The database is then enriched with images of the temple's extrapolated decoration, which involves 3D modelling of these extrapolations. *Talatat* indexing was based on the Multilingual Egyptian Thesaurus conventions regarding "passport" data, including iconographic description using descriptive operators called *unicos*. In the spirit of the international movement in favour of open access to scientific data, the *talatat* metadata and images are accessible online to researchers working on the proto-Amarna or Amarna periods. The *talatat* metadata is published using RDFa data model mapping for embedding RDF triples within the XHTML of our web pages, which can be extracted by compliant user agents. This corpus is stored in a secured warehouse with strong human and digital infrastructure for preservation of the images and of their metadata.

Carlos GRACIA ZAMACONA, A Database for the Coffin Texts

This article describes a database for the Coffin Texts. It was first conceived as a semantic study of verbs of motion, and for this reason many of its files are linguistically focused. Nevertheless, it may be useful for other kinds of studies, because the software employed allows integration of new files as well as modification of old ones. This is the ultimate aim of such a database: a tool appropriate for all kinds

of research on this corpus. Specific features of this corpus are discussed first, followed by the database conception and structure, and finally its use, results and developments.

Azza EZZAT, The Digital Library of Inscriptions and Calligraphies

The Digital Library of Inscriptions aims at recording all inscriptions on ancient Egyptian buildings and monuments throughout the ages. These inscriptions are digitally displayed for the user, including a brief description and pictures of the inscriptions. The languages included in the Digital Library are Ancient Egyptian, Arabic, Turkish, Persian and Greek languages. Moreover, there are inscriptions bearing Thamodic, Musnad, and Nabatean scripts.

Yannis GOURDON, The *AGÉA* Database Project.
Anthroponymes et Généalogies de l'Égypte Ancienne

Since the 30s, our understanding of the ancient Egyptian personal names has been dependent on Ranke's *Personennamen*. But, because the data and its philological and sociological analysis are based on the knowledge available in the first half of the 20th century, the *PN* requires a complete revision that takes into account recent developments on the subject. Launched in 2008 at the IFAO, the *AGÉA* database project aims, eventually, to create a systematic directory of personal names for every period of the Pharaonic history, completing and modernizing Ranke's work. As a tool facilitating more efficient analysis and a better interpretation of data, *AGÉA* will focus, in its first development, on the Old Kingdom.