

How aspects of descriptive and formal linguistics can inform LT for all languages

Lars Hellan

NTNU

Nidareid 5, 7017 Trondheim

lars.hellan@ntnu.no

Abstract

What exists for many ‘digitally less resourced’ languages (LRL) are grammars and dictionaries. Common to formats like CSV (underlying many dictionary tools) and grammar encoding formalisms like attribute-value matrices and feature unification, is that through 30-40 years of development and sustenance of life-long projects, sustainable tools for ‘whole language’-size resource creation are by now well established. Interesting possibilities resides in exploring ways in which the content of such resources can be channeled onto further types of structures and processing tools. The paper exemplifies this point through procedures for digital lexicon creation and annotation schemata reflecting advanced formal analysis.

Keywords: Less Resourced Languages, Valence dictionaries, Toolbox, Typed feature grammar, Corpus annotation schemata

Résumé

Hva mange 'digitalt lite utviklede språk' har er grammatikker og ordbøker. Felles for formater som CSV (bruket i mange ordboks-redsaker) og grammatikkformalismen som AVM of faktor-unifikasjon, er at det nå foreligger mange bærekraftige 'hel-språk'-ressurser. Det ligger interessante muligheter i å utforske muligheter for hvordan innholdet i slike ressurser kan bli kanalisert inn i videre typer prosesseringsredsaker. Artikkelen eksemplifiserer dette gjennom prosedyrer for fremstilling av digitale leksika og annotasjonsskjema som avspeiler avansert formell analyse.

1. Introduction

An aspect of language technology (LT) somewhat less highlighted in recent years is its contributions to grammar description and lexicography, in respects as basic as those of formal structuring of content and enhancing consistency. Albeit the creation of grammars and dictionaries are tasks belonging within linguistics/lexicography, when these activities are conducted using computationally tractable formats, the consistency thereby attained can be readily further exploited in the creation of processing tools and applications.

Traditionally, the creation of grammars and dictionaries is very much ‘rule based’ – although the induction of such resources from corpora and through machine learning are exciting issues in current NLP, hardly any language has had its basic grammar and lexicon resources produced by such means. For most ‘digitally less resourced’ languages (LRL), procedures in this domain are not even in question given the scarcity of digital resources. What nevertheless exists for many such languages are grammars and dictionaries – in many cases just in printed versions (or even just handwritten), but since the 1990ies also to a growing extent with dictionaries encoded in tools like Shoebox and Toolbox. Formal grammars of LRLs are much rarer, but common to formats like CSV (underlying dictionary tools as mentioned) and grammar encoding formalisms like attribute-value matrices and feature unification, is that through 30-40 years of development and sustenance of (person-) life-long projects, sustainable tools for ‘whole language’-size resource creation are by now well established. From the perspectives of ‘LT for all’ related to

indigenous languages, most of which will count as LRL, interesting possibilities resides in (i) furthering the use of formats like those mentioned to ever more languages; (ii) exploring ways in which the content of such resources can be channeled onto further types of structures and processing tools.

It is well known that Toolbox dictionaries can be converted into lexicon modules of feature structure grammars; we describe how a further step in this line of development underlies the creation of a large scale valence dictionary of the West African language Ga (Kwa), viz. Dakubu 2010, 2011.

The representation of grammatical structures of types prevalent in many LRLs, although less common in European languages, such as complex verb extensions in Bantu and Serial Verb Constructions (SVCs) in Kwa, is by now well established using feature structure formalisms. We exemplify formalisms for sorting of lexical items and for corpus annotation which reflect feature structure analysis, thereby opening for further connections between formal analysis in the domain of lexicon and grammar on the one hand, and notational features applicable in dictionaries and corpora.

In both cases what is exemplified are ways in which research in, and resource development for, LRLs can be enhanced using well established formalisms within logic and computational linguistics, based on independently established linguistic work. These may not yet yield digital applications ready for a user marked, but with attention thus directed to types of resources that already exist or are within reach for a multitude of LRLs, one may reduce the distance between linguistic resources and LT.

2. Creating a valence dictionary from a Toolbox lexicon

The digital lexicon (Dakubu 2010) is an amended version of a Ga Toolbox lexicon project holding data for a general-purpose dictionary. (Dakubu 2011) is a free-standing linguistic monograph. The former consists of 80,000 lines of code with 7080 entries, of which 5014 are for nouns and 935 for verbs. Here valence codes are written into the lexical entry following the general field style of Toolbox, where for the item *ba*, for instance, fields named \pdl-\pdv represent inflectional information of the lexeme , and the fields \xe, \xg, \xv together constitute a standard linguistic glossing with \xv as a word-and-morph break-up, \xg as morphological and English gloss, and \xe as a free English translation. With valence information added, a verb with more than one valence frame has one entry specified per frame; thus the verb *ba* ‘come’, for instance, is represented by 18 different entries in this edition of the Toolbox file. In this way, 547 verb lexemes from the original file are represented through altogether 2006 entries. The valence specification follows principles and formalization laid out in (Hellan and Dakubu 2010),¹ the *Construction Labeling (CL)* system. In this formalization one of these frames can be represented on the form given in (1), to be read as ‘a verb-headed intransitive syntactic frame where the subject carries an agent role and the situation expressed belongs to the type ‘MOTIONDIRECTED’.

(1) v-intr-suAg-MOTIONDIRECTED

The semantic specification here consists of two parts, *semantic role* as exemplified by ‘suAg’ and *situation type* as exemplified by ‘MOTIONDIRECTED’, the latter out of a total inventory of about 130 situation types.²

The classification using all the parameters recognizes about 100 construction types, which for mono-verbal constructions could also be seen as valence types. This addition to the Toolbox file thus constitutes a valence lexicon, with illustrating sentences. A small corpus further illustrates these construction types.³

With a set of 2000 entries classified by strings like (1), the valence notation allows one to investigate the frequency of frames used relative to these frames, correspondences between syntactic and semantic structure, the clustering of certain valence types for sets of verbs, and more.

The specifications of the Toolbox valence lexicon are also used for classification in a Ga lexicon with the 2000 verb entries at the online (4-language) valence lexicon *MultiVal*.⁴ The lexicon is also used in a computational grammar of Ga based on the HPSG framework. These are examples of how information, once digitally encoded, can be recast in other formats and used for other purposes.

This briefly illustrates how a general-purpose lexicon can be expanded to a valence lexicon, in turn used in a so-

called ‘deep processing’ grammar and in an online multilingual valence resource. All steps are technically straightforward, only the task of specifying valences is time-consuming, and can only be done by a linguist interested in creating such specifications – which is after all the normal way of creating linguistic resources.

3. Representing grammatical analysis in corpus annotation

In this section we illustrate a schema for integrating grammatical analysis into corpus annotation. The schema provides construction-level annotation tags which in one-line strings provide much of the information that could otherwise be expressed in multi-tier syntactic and semantic annotation. The strings are subject to semi-automatic consistency control, and can also be applied in valence specification in lexicons, grammatical parsing, and more. The tag system is referred to as Construction Labeling (CL), mentioned in the previous section, earlier presented in Hellan and Dakubu 2010 and Dakubu and Hellan 2017, but with the added capacity of serving as types in a Typed Feature Structure (TFS) system, enabling the consistency control and the parsing functionality.

To illustrate the complexity of information that can be accommodated, we use examples from Bantu instantiating verbal derivation and what may be called ‘skewed’ semantics.

The construction tags can be combined with standard word-by-word&morph-by-morph IGT annotation, as in TypeCraft (cf. Beermann and Mihaylov 2014), adding just a single line as annotation to the verb, as schematically illustrated with an example of verb derivation from Citumbuka (Bantu).

(2)

Mary wa-ka-mu-phik-isk-a	John	nchunga
Mary ISM-Pst-IOM-cook-Caus-FV	John	beans
N	V	N

vCaus-dbobCs

‘Mary made John cook beans’

vCaus means that the head is a verb and has a causative morpheme, and *dbobCs* means that the construction is a double object construction ‘derived’ through causativization and with the corresponding semantics non-isomorphic to the syntactic structure, a constellation we refer to as ‘skewed’ semantics.

We can make specifications of arguments of a derived verb in terms of their derivational histories, e.g., extending the formula *vCaus-dbobCs* to

(3) *vCaus-dbobCs-suC-obCsu-ob2Cob*

where the added items read as follows, similar to a formalism used in Relational Grammar:

<i>suC</i>	-	<i>subject created by Causativization</i>
<i>obCsu</i>	-	<i>object derived (‘demoted’) from subject by Causativization</i>
<i>ob2Cob</i>	-	<i>second object derived from object by Causativization</i>

¹ Also see (Dakubu and Hellan 2017).

² See (Dakubu 2011) and (Hellan and Dakubu 2010).

³ See https://typecraft.org/tc2wiki/Ga_Valence_Profile. The data are searchable, so that a search for, e.g., the constructional factor *obPostp* (‘object is a postposition’) yields an array of urls for the sentences instantiating the factor.

⁴ Cf. (Hellan and Beermann. 2014).

Expanding from what was said in section 2, each CL tag is a string consisting of, first, a label specifying POS of head of the construction and salient morphological marking (like *vCaus* in (2)), second, a label designating the overall structure of a construction (encoding notions like intransitive, transitive, ditransitive/double object, etc. (such as *dbobCs* in (2)), third a string of labels classifying features of the arguments - first syntactic features and then semantic features -, and finally a string of labels for TAM features and situational content. (4) further illustrates the format, applicable to a sentence like *John ate the cake*:

(4) *v-tr-suAg_obAffinccrem-COMPLETED*

Whenever a putative CL string is composed, the labels of the string have to match – for instance, if one label is *intr*, then there cannot be an argument label prefixed by *ob*, since *intr* is not defined such a label. A processing mechanism enforcing such consistency is provided using a Typed Feature Structure (TFS) system, in which the CL tag labels are defined as *types*.

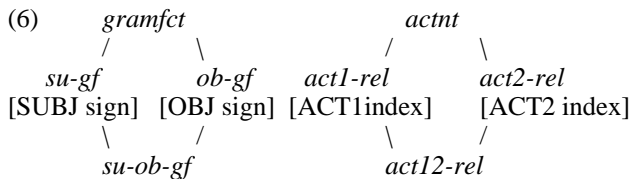
Information in such a system is generally exposed through Attribute Value Matrices (AVMs), where each AVM belongs to a type, and attributes are introduced (declared) according to the following conventions:

(5) [A] A given type introduces the same attribute(s) no matter in which environment it is used.

[B] A given attribute is declared by one type only (but occurs with all of its subtypes).

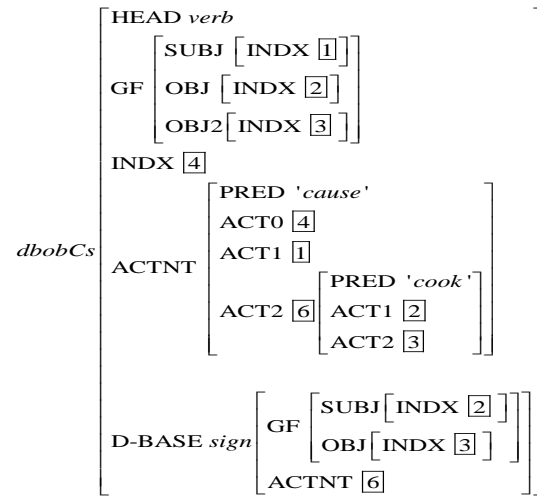
In a TFS representing a grammar, there are many type hierarchies, representing POS, tenses, semantic roles, etc.; some of these hierarchies do without attributes, while the following ones do.

Types for grammatical functions (values of ‘GF’) and actants (values of ‘ACTNT’) include those indicated below, the *gramfct* subtypes declaring GF attributes (‘SUBJ’ and ‘OBJ’) and the *actnt* subtypes declaring semantic participant attributes (‘ACT1’ and ‘ACT2’):

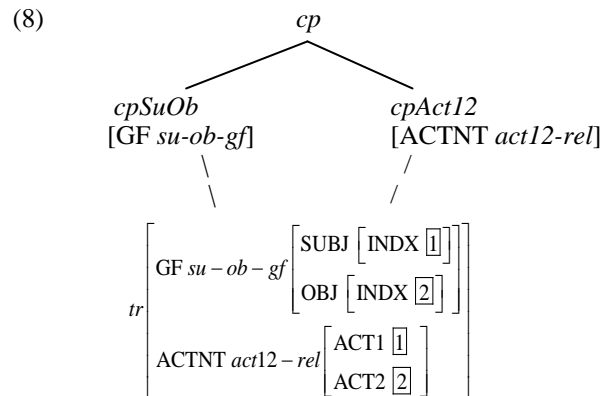


With such features as basis, one can represent, e.g., (2) as (7), which is an AVM representing a construction, which involves a specification of grammatical functions and actants acting together, identified through the attributes GF and ACTNT, neither of which are introduced in (6), but which are introduced at the level of constructions.

(7) AVM for double object construction with causative semantics and causative derivation:

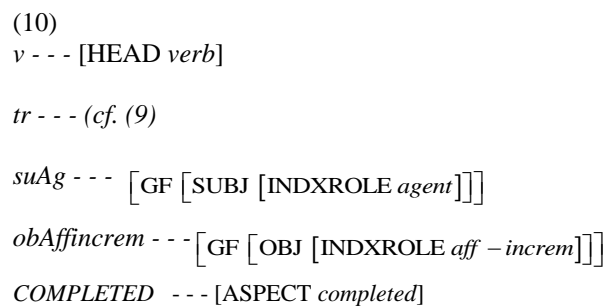


In their capacity as types, CL labels define AVMs at the formal level of constructions, thus as subtypes of *cp* with all the attributes defined for *cp*. Type definitions sustaining the type *tr* (‘transitive’) reflect this, as shown in the following. The definition of *tr* is achieved through a join of *cp* specifications:



tr is thus formally defined as a type of *sign*, or *construction*. Similar depths of specification are required for all CL labels.

When CL labels occur in a string, as in (4), they unify. To illustrate, the types to which the labels in (4) correspond are indicated in (9), and the unification result is (10):



$$(11) \left[\begin{array}{l} \text{HEAD verb} \\ \text{GF} \left[\begin{array}{l} \text{SUBJ} \left[\text{INDX} \left[\begin{array}{l} \text{1} \\ \text{[ROLE agent]} \end{array} \right] \right] \\ \text{OBJ} \left[\text{INDX} \left[\begin{array}{l} \text{2} \\ \text{[ROLE aff-increm]} \end{array} \right] \right] \end{array} \right] \\ \text{ASPECT completed} \\ \text{ACTNT} \left[\begin{array}{l} \text{ACT1} \left[\begin{array}{l} \text{1} \end{array} \right] \\ \text{ACT2} \left[\begin{array}{l} \text{2} \end{array} \right] \end{array} \right] \end{array} \right]$$

Returning to the more complex label in (3), the AVMs for the ‘derivational histories’ will be as in (12), the unification of which with a structure for dbobCs in isolation will be the structure in (8);

$$(12) \text{ a. } \textit{suC} \left[\begin{array}{l} \text{GF} \left[\text{SUBJ} \textit{sign} \left[\text{INDX} \left[\begin{array}{l} \text{1} \end{array} \right] \right] \right] \\ \text{ACTNT} \left[\begin{array}{l} \text{PRED 'cause'} \\ \text{ACT1} \left[\begin{array}{l} \text{1} \end{array} \right] \end{array} \right] \end{array} \right]$$

$$\text{ b. } \textit{obCsu} \left[\begin{array}{l} \text{GF} \left[\text{OBJ} \left[\text{INDX} \left[\begin{array}{l} \text{2} \end{array} \right] \right] \right] \\ \text{ACTNT} \left[\begin{array}{l} \text{PRED 'cause'} \\ \text{ACT2} \left[\begin{array}{l} \text{6} \end{array} \right] \left[\text{ACT1} \left[\begin{array}{l} \text{2} \end{array} \right] \right] \end{array} \right] \\ \text{D-BASE} \textit{sign} \left[\text{GF} \left[\text{SUBJ} \left[\text{INDX} \left[\begin{array}{l} \text{2} \end{array} \right] \right] \right] \right] \\ \text{ACTNT} \left[\begin{array}{l} \text{6} \end{array} \right] \end{array} \right]$$

$$\text{ c. } \textit{ob2Cob} \left[\begin{array}{l} \text{GF} \left[\text{OBJ2} \left[\text{INDX} \left[\begin{array}{l} \text{3} \end{array} \right] \right] \right] \\ \text{ACTNT} \left[\begin{array}{l} \text{PRED 'cause'} \\ \text{ACT2} \left[\begin{array}{l} \text{6} \end{array} \right] \left[\text{ACT2} \left[\begin{array}{l} \text{3} \end{array} \right] \right] \end{array} \right] \\ \text{D-BASE} \textit{sign} \left[\text{GF} \left[\text{OBJ} \left[\text{INDX} \left[\begin{array}{l} \text{3} \end{array} \right] \right] \right] \right] \\ \text{ACTNT} \left[\begin{array}{l} \text{6} \end{array} \right] \end{array} \right]$$

Unification presupposing feature compatibility, a control of consistency is inbuilt in this formalism.

It is possible to run the TFS system as a parser where the IGT annotations serve as a ‘pre-processed’ input, and where the CL type assigned to the verb acts as a valence requirement. Consistency relative to the whole IGT can then be similarly imposed.

With such a parsing facility one can also generate for each sentence a detailed total structure, exposing for instance which parts of a sentence are subject and object, information one otherwise will expect to find in a treebank.

A design as now outlined addresses one aspect of what concerns correctness of annotation – that of consistency. Another aspect is of course factual correctness. In-between lurks the issue of ‘using a correct annotation set’. Within a given project building up a database, it can be essential that the same tags are used for the same phenomena. But in a general perspective, there are many

reasonable ways in which to name a phenomenon and assign tags to it.

In the present approach, one can freely add labels to the defined set, as long as they have concise definitions into the TFS system. Thus, if it is a matter of an alternative tag for an already ‘tagged’ phenomenon, one just equals the tags. In cases of a not yet accommodated phenomenon, more will be involved, ranging from filling a gap in an already established paradigm, to creating a new analysis in the TFS, the latter of course interesting but also more involved.

4. Conclusion

The CL annotation tagset consists of symbols which are on the one hand descriptive labels, and on the other hand labels for types reflecting multiple layers of analysis. The descriptive labels allow one to stay within the IGT overall format, while their type definitions allow for additional layers of analytic representations, and for the possibility of defining semi-automatic consistency-checking procedures. We have illustrated its relevance both in the development of computational lexical resources from linguistic lexicons, and in the design of a corpus annotation schema reflecting ‘deep’ analytic features.

The general point that we want to make through this illustration is that descriptive linguistic resources need not be far from constituting interesting digital resources, also for so-called less resourced languages, and that ‘deep’ and formal structures of languages can be readily reflected in annotation schemata applied to small or larger corpora, equally readily for less resourced languages with a decent descriptive literature, as for well resourced languages.

5. References

- Beermann, Dorothee and Mihaylov, Pavel. 2014. Collaborative databasing and Resource sharing for Linguists. *Languages Resources and Evaluation* 48. Dordrecht: Springer, 1-23.
- Dakubu, Mary Esther Kropp. 2010. ‘[Ga verb dictionary for digital processing](https://typecraft.org/tc2wiki/Ga_Valence_Profile)’.
- Dakubu, Mary Esther Kropp. 2011. Ga Verbs and their constructions. Monograph ms, Univ. of Ghana.
- Dakubu, M.E. Kropp and Lars Hellan. 2017. A labeling system for valency: linguistic coverage and applications. In Hellan, L., Malchukov, A., and Cennamo, M (eds) *Contrastive studies in Valency*. Amsterdam & Philadelphia: John Benjamins Publ. Co.
- Hellan, Lars and M.E. Kropp Dakubu. 2010. *Identifying verb constructions cross-linguistically*. In *Studies in the Languages of the Volta Basin* 6.3. Legon: Linguistics Department, University of Ghana.
- Hellan, Lars, and Dorothee Beermann. 2014. Inducing grammars from IGT. In Z. Vetulani and J. Mariani (eds.) *Human Language Technologies as a Challenge for Computer Science and Linguistics*. Springer.