Norwegian resources – things we have, and would like to develop, related to *Valency*

Lars Hellan, Workshop on Machine Learning, Trondheim 22-23 Sept 2016

Applications related to valency

1. Indicating a verb of a language, you can call on examples of various frames in which the verb can occur.
2. With a verb and a given frame in mind, you can indicate both, and get examples of how the verb is used in this frame (meaning, concurrent items, style, ...)
3. Indicating a valency frame abstractly, you can get an overview of which verbs occur with that frame.
4. Across languages, point 3 can also be applied.
5. Replicas of points 1 and 2 across languages must refer to verb *meanings* (or cognates), but are then feasible as well.
6. Abstractly, for a given verb, you can get all the frames in which it occurs in the language, abstractly represented or with a standard example not necessarily involving that verb.
7. Across languages, the same is again feasible, but then with reference to a verb *meaning*.

Points 1, 2 and 5 are suited for *services* – for learning, writing assistance, translation.
Points 2, 3, 6 and 7 are more for research. For Norwegian we already have 3 and 6 realized, and our concern is to build corpora reflecting the information, and helping expanding it. First, examples of online queries concerning 3 and 6, and some research perspectives:

3. From http://regdili.hf.ntnu.no:8081/multilanguage_valence_demo/multivalence

ane_io-epon EXPL+NP+Sdecl

angå_io-epon EXPL+NP+Sdecl

bekymre_io-epon EXPL+NP+Sdecl

forekomme_io-epon EXPL+NP+Sdecl

foresveve_io-epon EXPL+NP+Sdecl

plage_io-epon EXPL+NP+Sdecl

plage_trExpnDECL EXPL+NP+Sdecl

synes_io-epon EXPL+NP+Sdecl

undre_trExpnDECL EXPL+NP+Sdecl

vedkomme_io-epon EXPL+NP+Sdecl

6. From http://regdili.hf.ntnu.no:8081/multilanguage_valence_demo/multivalence

spise-ppshift NP+PP[refl]+NP

spise_atelobl-av NP+PP

spise_iv NP

spise_tr NP+NP

spise_tr-refl-obl NP+NPrefl+PP

spise_tr-secpred-arg1 NP+NP+XPpred

spise_tr-secpred-arg1-refl NP+NPrefl+XPpred

Some research perspectives

A verb can often be seen as taking more than one frame, and we can refer to such a set of frames taken by a given verb as a *frame pod*. From existing estimates, we may say that about half of the verbs in a language take only one frame, the others more, mostly two or three, but the number can be up to 20 for certain types of verbs, given fairly conservative criteria for what can be recognized as parts of a frame. In a given pod, we thus have one and the same verb V occurring with more than one frame, so that the pod abstractly speaking is a set of pairs <V, Frame1>, <V, Frame2>, <V, Frame3>, etc.; we may refer to each such pair as a *val*(ency)-*instance* of the verb, and the verb V by itself as the *host* relative to that pod. By rough estimate, from a totality of verbs of a language, if one assigns the possible pods to all of them, the total number of val-instances will be about 30-40% higher than the number of verbs-per-se.

One can also make the abstraction of speaking of the possible *verb frames* of a language by themselves, then using notions like 'intransitive', 'transitive', 'ditransitive', etc.; these notions constitute a familiar dimension of classification, although the number of such frames, given the same criteria as alluded to above for what can be recognized as part of a frame, may easily amount to between 250 and 300 for well-investigated languages, with up to 10,000 val-instances recoded (and thus around 7000 verbs-per-se, or 'pod hosts'). These frames are not the same from language to language (although there is of course overlap, and more so the closer the languages), and so we may refer to the set of verb frames obtaining for a language as the *valency profile* of that language.[1] Out of such a profile and a verb inventory, it is then straightforward to make overviews of which frames take which verbs and how many verbs, which frames co-occur within the same frame pods, and similar figures. With a set of languages investigated in these ways, one can extend the methodology to also *compare* the languages in all the dimensions already calibrated.

However, in an attempt to *understand* such distributions, and in turn *explain* why they unfold as they do within a language and across languages, one also needs measures of what distinct verbs can have in common apart from possible shared membership in frame-pods, both inside and across languages. Such a measure has to refer to *meaning*. Well known is for instance the assumption that shared pod membership between verbs is correlated with meaning similarity between the verbs, phrased in such terms as 'verb of motion', 'verb of perception' and the like - this is a hypothesis proposed (in slightly different terms) in the work of Levin 1993, where the term 'valency class' is used for approximately the same as what we here call 'shared membership in a frame-pod', however with the implication that interesting shared semantic properties underlie shared membership – this pertains to the semantics of the individual verbs involved, as well as the potentials of meaning that can be sustained by the frame types in question. Levin's work has been pursued for English during *VerbNet*,[2] at a large scale, with 6340 verbs divided into 273 verb classes, and at smaller scales for many languages.

Thus, in the domain of so-called 'valency classes' (conceived as classes of verbs sharing the same valency frame potential, along with semantic similarities), it could be that languages with very different general syntactic patterns have *corresponding* valency class inventories, in the sense that when a given set of verbs in one language belong to the same class, then the set of semantically corresponding verbs in another language also belong to the same class in *that* language; the array of languages here available is a good selection for such studies. Moreover, at both in- and across languages basis, questions concerning 'meaning' of valency frames could be investigated: although commonalities between, say, all transitive verb frames would obviously not obtain at the level of commonly discussed event types, there might still be factors, beyond simply two-participant-hood, that could be detected once a large enough number of verbs are taken into consideration. There being more than 300 valency types available in the database, this is a rich resource for potential research along these lines.[3] [4]

---

[1] For one approach to establishing valency profiles, see Dakubu and Hellan (forthcoming), and https://typecraft.org/tc2wiki/Category:Valence_by_language

[2] Cf. Korhonen and Briscoe 2006; http://verbs.colorado.edu/~mpalmer/projects/verbnet.html), with links also to *Prop-Bank* and *FrameNe*t (Fillmore 2007).

[3] Apart from the Ga lexicon, these lexicons all lack English glossing; if this were uniformly available, one could have a pragmatically more easy way of conducting some of the research mentioned, even though from a theoretical viewpoint, it is a somewhat lame substitute for a linguistic semantics.

[4] Even so, as argued, e.g., in Faulhaber 2011, there are many exceptions to hypotheses so far made in this domain, and optimism for such investigations should only be maintained with caution..

Current research practice offers two strategies involving reference to meaning of verbs. One is deployed, e.g., in the *The Leipzig Valency Classes* project ('LVP'; Malchukov and Comrie 2015), where the arrays of frames for 80 verb meanings are compared across 30 languages: here one uses English verbs as 'names' of verb meanings (e.g., the 'kill verb' maintained as a constant entity across related as well as unrelated languages). The other strategy is one of 'lexical decomposition', used e.g. in VerbNet, where putatively systematic semantic discriminating factors are represented by abstract predicates, such as 'CAUSE', 'BECOME', 'GO', etc., thus maintaining English 'verbs' only in a restricted meta-language vocabulary, as opposed to the non-decomposed meaning labels used in LVP. Being thus some notches removed from the vocabulary of an actual language, the decomposition approach has a better chance at pinpointing semantic factors on a cross-linguistic basis.

A more radical approach along the latter line, started by Dorothee a few years ago, illustrated by Locomotion verbs which involve classifying factors like those summarized in Table 1 (all having *iteration* as ACTIVITY-PATTERN, and 'Agentive Mover' as ROLE):

German	English	Norw	KIND	INCREM-DIM	ITER-PATT	VELOCITY	PARTor-ORGAN
laufen	*run/go*	*løpe/gå*	legged-anim	path-horisontal	zip-lock	unmarked	all-legs
gehen	*walk*	*gå*	legged-anim	path-horisontal	zip-lock	unmarked	all-legs
rennen	*run*	*løpe*	legged-anim	path-horisontal	zip-lock	fast	all-legs
schlendern	*stroll*	*slentre*	Human	path- horisontal	zip-lock	slow	both-legs
spazieren	*stroll*	*spasere*	Human	path- horisontal	zip-lock	slow	both-legs
klettern	*climb*	*klatre*	legged-anim	path- vertical	zip-lock	unmarked	all-limbs
hinken		*hinke*	2- legged-anim	path- horisontal	succession	unmarked	one-leg
	creep	*krype*	Anim	path- horisontal	succession	unmarked	under-side
fliegen	*fly*	*fly*		path	succession	unmarked	wings/arms

Table 1 Locomotion verbs of iteration

Now to points 1 and 2, repeated, and the development of data resources:

1. Indicating a verb of a language, you can call on examples of various frames in which the verb can occur.
2. With a verb and a given frame in mind, you can indicate both, and get examples of how the verb is used in this frame (meaning, concurrent items, style, ...)

The stickword – introduced by Dorothee - is *incubation data* – data between 'raw' data and 'targeted' data (data annotated for some specific purpose); data qualified as relevant, but not yet propelled into use.

As 'raw' data related to valency for Norwegian:
Large corpora in Norway include the Nationalbiblioteket's 20 billion words database, which has metadata (http://www.nb.no/sp_tjenester/beta/ngram_1/) , and NoWac, which lacks metadata, but is popular among linguists for its easy access (http://www.tekstlab.uio.no/nowac/) .
Then there is the LCC Norwegian corpus ...

As 'targeted' data related to valency for Norwegian:
Norwegian text data which can serve for *valency* investigation (and perhaps sentence semantics investigation). Such data should contain *POS* and standard *morphological Gloss* information, and **valency annotation on each verb reflecting the type of frame it has in the given sentence**.
TC has many annotation instances along these lines, showing one for illustration:

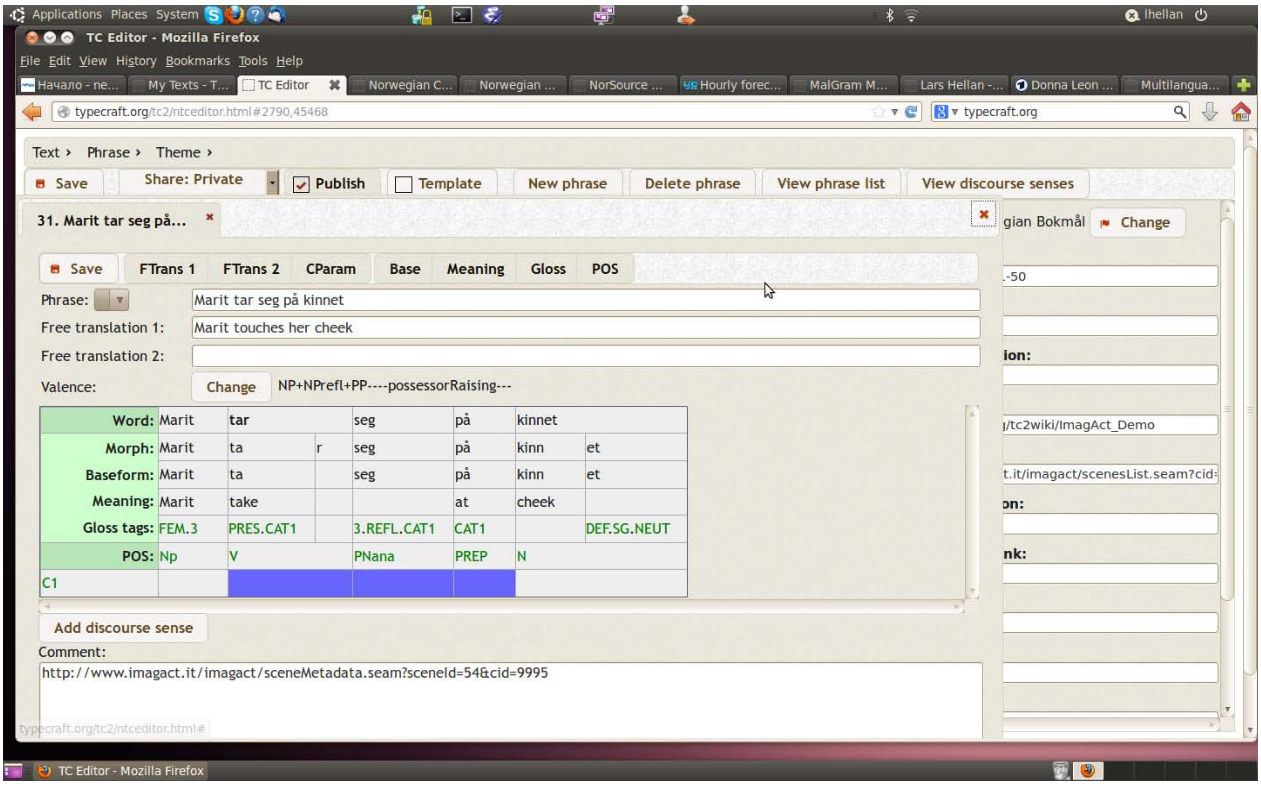

Figure 1 Annotation in TypeCraft Editor

Figure 2 Still-picture of ImagAct scene

Examples of how valency annotation can look:

- TC and MultiVal 'global' Syntactic Argument Structure tags (*see Figure 1, and Appendix 1 below*)
- MultiVal 'global' Functional Structure tags (*see Appendix 2 below; https://typecraft.org/tc2wiki/Multilingual_Verb_Valence_Lexicon*)
- TC 'on-verb' tags (see POS Tags list in TC)
- Construction Labeling tags in TC 'Description' (and the background overall system)
  https://typecraft.org/tc2wiki/Valence_Profile_Norwegian
- ValPaL 'typological' tags http://valpal.info/

The incubation data
They should be instances of recognized (or candidate) valency frames, but not yet annotated. Instances of each intended frame can have as 'group' metadata information about the intended frame (to ease the annotation when it takes place).

Envisioned strategies to find and classify incubation valency data, addressing problems of:
- counting the number of NPs
- distinguishing argument NPs from adjunct NPs
- identifying 'particles'
- distinguishing adjunct PPs from argument PPs
- identifying 'light verb' clusters (LVCs)

All the problems, except the first, involve enumeration of:
- heads of non-argument NPs (like time units, dates, length units, …)
- particles (adverbs: essentially the following: …)
- strings consisting of verbs and their preferred 'selected' preposition(s)
  Cf. https://typecraft.org/tc2wiki/Verb_-_Preposition_expressions_in_Norwegian , and a snippet illustration in *Appendix 3.*
- light-verb supporting nouns (situational nouns)
- for each lightverb-supporting noun, the LVC strings V+N(+Prep) that the noun can occur in.
  (see remarks and paper on https://typecraft.org/tc2wiki/Bare_Nominalizations_in_Norwegian)

Background resources:
Lexical resources: Norsource, NorKompLeks, MultiVal, Norsk Ordbank …
(for some, see *Appendix 4*)

APPENDIX 1 *TC and MultiVal 'global' Syntactic Argument Structure (SAS) tags*

EXPL
EXPL+APpred+S
EXPL+APpred+adpos
EXPL+INF
EXPL+NP
EXPL+NP+INF
EXPL+NP+INF[equiOBJ]
EXPL+NP+NP
EXPL+NP+NP+INF
EXPL+NP+Sdecl
EXPL+NP+SquestWH
EXPL+NP+SquestYN
EXPL+NP+adpos
EXPL+NPpred+Sdecl
EXPL+NPpred+SquestWH
EXPL+NPpred+SquestYN
EXPL+NPrefl+NP
EXPL+NPrefl+NP+adpos
EXPL+NPrefl+S
EXPL+NPrefl+Sdecl
EXPL+NPrefl+SquestWH
EXPL+NPrefl+SquestYN
EXPL+NPrefl+adpos
EXPL+NPrefl+adpos+NP
EXPL+PP+Sdecl
EXPL+PP+SquestWH
EXPL+PP+SquestYN
EXPL+PP[Sdecl]
EXPL+PP[SquestWH]+SquestWH
EXPL+PP[SquestWH]+SquestYN
EXPL+PP[SquestYN]+SquestWH
EXPL+PP[SquestYN]+SquestYN
EXPL+PPpred+Sdecl
EXPL+PPpred+SquestYN
EXPL+PRTP[S]
EXPL+PRTPpred+S
EXPL+Sdecl
EXPL+Squest
EXPL+adpos
EXPL+adpos+INF
EXPL+adpos+NP
EXPL+adpos+PP[S]
EXPL+adpos+Sdecl
INF
INF+NP
INF+NP+NP
INF+NPpred
INF+PP
NP
NP+ADVP+PP
NP+ADVPpred
NP+ADVPpred+NP
NP+APpred
NP+APpred+adpos
NP+EXPL+INF
NP+EXPL+S
NP+EXPL+Sdecl
NP+INF
NP+INF:equiSBJ
NP+INF:raisingSBJ
NP+NP
NP+NP+APpred
NP+NP+INF:equiOBJ
NP+NP+INF:equiSBJ
NP+NP+INF:raisingOBJ
NP+NP+INF:raisingSBJ
NP+NP+NP

NP+NP+NP+PP
NP+NP+NPpred
NP+NP+PP
NP+NP+PP+PP
NP+NP+PP[INF:equiOBJ]
NP+NP+PP[INF]
NP+NP+PP[Sdecl]
NP+NP+PP[Squest]
NP+NP+PRTPpred:for
NP+NP+PRTPpred:som
NP+NP+PRTPpred:til
NP+NP+Sdecl
NP+NP+SquestWH
NP+NP+SquestYN
NP+NP+VPpass
NP+NP+XPpred
NP+NP+adpos
NP+NPpred
NP+NPrefl
NP+NPrefl+ADVP
NP+NPrefl+ADVPpred
NP+NPrefl+APpred
NP+NPrefl+INF:equiOBJ
NP+NPrefl+INF:raisingOBJ
NP+NPrefl+INF:raisingSBJ
NP+NPrefl+NP
NP+NPrefl+NPpred
NP+NPrefl+PP
NP+NPrefl+PP[INF:equiOBJ]
NP+NPrefl+PP[Sdecl]
NP+NPrefl+PP[Squest]
NP+NPrefl+PRTP:for[INF:raisingOBJ]
NP+NPrefl+PRTPpred:som
NP+NPrefl+PRTPpred:til
NP+NPrefl+S
NP+NPrefl+SquestWH
NP+NPrefl+SquestYN
NP+NPrefl+XP:raisingSBJ
NP+NPrefl+XPpred
NP+NPrefl+adpos
NP+PP
NP+PP+NP
NP+PP+PP
NP+PP:til[INF:raisingSBJ]
NP+PP[INF:equiSBJ]
NP+PP[Sdecl:som]
NP+PP[Sdecl]
NP+PP[Squest]
NP+PP[refl]
NP+PP[refl]+NP
NP+PPpred
NP+PRTPpred
NP+PRTPpred:som
NP+Sdecl
NP+Squest
NP+VP
NP+VPpass
NP+VPperf
NP+XPpred
NP+adpos
NP+adpos+NP
NP+adpos+PP:som[S]
NP+adpos+PP[INF:equiSBJ]
NP+adpos+PRTP
NP+adpos+S
S+PP[Sdecl]
Sdecl
Sdecl+APpred
Sdecl+NP

Sdecl+NP+NP
Sdecl+NP+PP
Sdecl+NP+PP[Sdecl]
Sdecl+NP+PP[Squest]
Sdecl+NP+Sdecl
Sdecl+NP+Squest
Sdecl+NPpred
Sdecl+PP
Sdecl+PPpred
Sdecl+Sdecl
Sdecl+Squest
Squest
Squest+APpred
Squest+NP
Squest+NP+NP
Squest+NP+Squest
Squest+PP
Squest+PP[Squest]
Squest+Squest
SquestWH+NPpred
SquestYN+NPpred
(158)

APPENDIX 2 *MultiVal 'global' Functional Structure tags*

intransitive
intransImpersonal
intransImpersonalWithParticle
intransImpersonalWithOblique
intransPresentational
intransPresentationalWithPath
intransPresentationalWithLocative
intransWithImplicitObject
intransWithPath
intransWithLocative
intransWithAdv
intransWithParticle
intransWithParticleAndOblique
intransWithParticleAndSecondaryPredicate
intransWithSecondaryPredicateAndParticle
intransWithSententialComplement
intransWithOblique
intransWith2Oblique
intransWithObliqueAndAdv
intransWithObliqueWithRaisingcomplement
intransWithSecondaryPredicate
intransLightWithSecondaryPredicate
intransWithExtraposedClause
intransWithParticleAndExtraposedClause
intransWithObliqueWithRaisingcomplement
intransWithParticleAndObliqueWithRaisingcomplement
intransWithObliqueAndExtraposedClause
intransWithObliqueWithExtralinkedClause
intransWithParticleAndObliqueWithExtralinkedClause
intransModauxWithSecondaryPredicate
intransPerfauxWithSecondaryPredicate
intransPassauxWithSecondaryPredicate
intransWithSecondaryPredicateAndExtraposedClause
intransWithDem ; Ga
intransByPassive

transitive
transImpersonal
transImpersonalWithParticle
transImpersonalWithOblique
transPresentational
transPresentationalWithPath
transPresentationalWithLocative
transWithImplicitObject

transWithNonreferentialObject
transWithNonreferentialObjectAndSecondaryPredicate
transWithPath
transWithLocative
transWithAdv
transWithParticle
transWithSententialComplement
transWithOblique
transWith2Oblique
transWithSecondarySubjectpredicate
transWithSecondaryObjectpredicate
transLightWithSecondarySubjectpredicate
transLightWithSecondaryObjectpredicate
transLight ; Ga
transLightWithOblique ; Ga
transWithSecondaryObjectpredicateAsComplexPassive
transWithExtraposedClause
transWithSubjlinkedExtraposedClause
transWithObjlinkedExtraposedClause
ditransitive
ditransWithNonreferentialObject
ditransWithOblique
ditransWithSubjlinkedExtraposedClause
ditransWithSententialComplement ; Ga
ditransLightSecobj ; Ga
doubleObject
doubleObjectWithOblique
copulaWithPredicativeAdjective
copulaWithPredicativeNoun
copulaWithPredicativePP
copulaWithPredicativeParticlephrase
copulaWithPredicativeAdjectiveAndLocative
copulaWithPredicativeAdverbAndLocative
copulaWithIdentityNoun
copulaWithIdentityInfinitive
copulaWithIdentityDeclClause
copulaWithIdentityYesNoClause
copulaWithIdentityWhClause
copulaWithIdentityInterrClause
copulaWithPredicativeAdjectiveAndExtraposedClause
copulaWithPredicativeNounAndExtraposedClause
copulaWithPredicativePPAndExtraposedClause
copulaWithPredicativeParticleAndExtraposedClause
copulaImpersonalWithPredicativeAdverbAndLocative
(88)

APPENDIX 3 Examples of Verb-Preposition strings (with *av* ‘of’)

ause av
avhenge av
avhenge av å
avhenge av at
avhenge av hv
benytte seg av
benytte seg av at
benytte seg av hv
bestå av
betjene seg av
betjene seg av å
betjene seg av at
briste av
bugne av
ernære seg av å
flire av
flire av at
fnise av
fnise av å
fnise av at

fnyse av
fnyse av å
fnyse av at
fnyse av hv
forgå av
framgå av
framgå av at
framgå av hv
framkomme av
fremgå av
fremgå av at
fremgå av hv
fremkomme av
fyse av
gjespe av
glemme av
glippe av
glømme av
grøsse av
grøsse av at
grøsse av hv
gyse av
gyse av at
hånflire av
hånle av
kimse av
kimse av at
kimse av hv
kjøpe av
kjøre av
komme av
korte av
koste av
le av
le av å
le av at
le av hv
lide av
livberge seg av
livnære seg av
logge seg av
lære av
lære av å
lære av at
lære av hv
ose av
plukke av
skryte av
skryte av å
skryte av at
skryte av hv
skvette av
sleike av
slikke av
smekte av
sovne av
spise av
strutte av
stønne av
ta seg av
vansmekte av
vri seg av

APPENDIX 4 Paper Hellan and Bruland, from LTC Poznan 2015:

A cluster of applications around a Deep Grammar

Lars Hellan Tore Bruland

NTNU
7491 Trondheim, Norway
lars.hellan@ntnu.no t-brul@online.no

Abstract

We describe a cluster of applications for Norwegian residing in lexical repositories, computational grammars, online valency dictionaries, a grammar tutoring device, and more, some of it generalized to other languages as well. The developments have taken place in a largely cumulative manner, so that earlier stages retain relevance to later stages, and with a shared knowledge basis of lexical and grammatical information. The design is replicable, and the paper describes both the applications and aspects of the persistent linguistic information.

Keywords: computational grammar, LKB, lexica, multilingual valency lexicon, grammar tutor, Norwegian

1. Introduction

We describe a sequence of developments of 'deep' applications for Norwegian, starting with lexical repositories. These were integrated into a computational grammar, and from this grammar online valency dictionaries, a grammar tutoring device, and a pos-tagger have been derived, some of these generalized to other languages. The developments have taken place in a largely cumulative manner, so that earlier stages retain relevance to later stages, and with a shared knowledge basis of lexical and grammatical information. The design is replicable, and is an example of how well motivated linguistic analysis and well understood computational techniques and platforms can be combined to constitute sustainable, information-rich repositories.

Chronologically first in the development were two lexical repositories, *TROLL* in the late 80ies and *NorKompLex* in the late 90ies, the latter partly extending the former. They were followed by a computational grammar built on the LKB platform (cf. Copestake 2002) using HPSG (cf. Pollard and Sag 1994), called *NorSource*, started in 2001 and still being developed, with information from the lexical repositories as its main 'start capital'. NorSource in turn has the following offsprings: an on-line language learning tool called the *Norwegian Grammar Sparrer* running on NorSource (from 2011 on); a large multi-lingual online valency lexicon, *MultiVal*, in its construction development based crucially on NorSource (from 2013 on), and as part of a robustness enhancement of the grammar, a POS-tagger constructed from the information in NorSource (2014).

From our perspective, NorSource may be seen as the architectural center point of these applications, with a typed feature structure (TFS) build-up which accommodates all the information in the lexical repositories, and with a computational TFS-based processing system which allows this information to be operative both in the general parser and in the three further applications.

In section 2 we describe the development of the resources, with a focus on how the creation of each one has been facilitated by components of the previous ones. Section 3 comments on aspects of the information encoding crucial to the way in which information has been able to migrate between the resources.

2. The applications

2.1 TROLL ('Trondheim Linguistic Lexicon')[5]

TROLL has about 10 000 lemmas of nouns, verbs and adjectives (acquired by the project). Its main focus is verbs, hosting around 2000 verb entries, with detailed information about valency, including both individual frames of the lemmas and information about the array of valency alternation frames relative to each frame (thus slightly predating the design of Levin 1993, later followed in many resources). In the definition of each basic frame, each argument is defined by a semantic role (out of an inventory of nine commonly recognized roles), a syntactic category and a grammatical function, constituting a triple <role, category, function> for each argument ('rkf-triple'). For each template, alternation possibilities are stated as *derivational* rules (D-rules), which can apply in ordered sequences, a bit like transformations in a transformational grammar, being defined over the information represented in the argument triples. D-rules in TROLL apply not only within the category of verbs ('intra-lex D-rules'), but also across categories, as when adjectives are derived from verbs, and nouns in turn from adjectives. With the 'basic' frames and the possible results of applications of sequences of D-rules, nearly 200 frames are defined as such by the system, and the total number of frame in-

[5] 1987-91, conducted at NTNU (then: University of Trondheim). Sponsored by NFR (The Norwegian Research Council), and NTNU. The main publication is Hellan et al. 1989, which, together with files and the lexicon database, can be downloaded at http://www.nb.no/sprakbanken/show?serial=sbr-40&lang=nb. Participants: Lars Hellan (project leader), Lars G. Johnsen, Margaret Magnus, Anneliese Pitz, Tor Åfarli, Hanneke van Hoof, Elisabeth Wennevold Silva, in a first phase, and Jon Atle Gulla, Ingebjørg Tonne, Anja Seibert and Sjur Moshagen in a subsequent phase.

stantiations across the 1000 ‘basic’ verb entries lies around 4000. The following is an example of a TROLL verb entry, representing the verb *bruke* ‘use’, where ‘101707’ is the entry identifier; ‘para’ and ‘cat’ introduce information about POS and inflectional properties (‘v2’ a ‘paradigm’ code):

(1)

```
101707 :=
 [para : v2,cat : [c : "V",infl : "inf"],
  real : [ascii : "bruke"],  templ :  (6, 27)].
```

‘templ(ates)’ introduces the verb’s argument structure properties, which here reside in two frames, one defined by the index ‘6’, the other by the index 27, these being independently defined macro labels: ‘6’ is the frame consisting of the rkf combination ’<ag, np, subj> + <th, np, gov>’ (that is, a typical transitive frame), and ‘27’ is the combination ’ <ag, np, subj> + <th, å-inf, gov>’ (thus a transitive frame with infinitival clause with the infinitive marker *å* as object). For each template there is a defined set of derivational rules; this single entry here thus encapsulates two ‘base’ templates relative to derivational rules, and the assembled ‘aggregate’ of what these rules can produce.

The TROLL system is physically a set of text files, concise enough to feed computational algorithms.

2.2 NorKompLex（Norwegian Computational Lexicon) [6]

NorKompLex (abbreviated *NKL*) has the aim of providing large scale dictionaries of both Bokmål and Nynorsk, covering in addition to the parameters of TROLL, phonetic specification for all word forms in a full-form lexicon.

In this project, about 100 of the 150 verb types accounted for in TROLL were selected, with argument structure specifications done as in TROLL, using essentially the ‘rkf’ format, but with macros expressed by labels such as ‘intrans1’, ‘intrans2’, ‘intrans3’, …. Basic and derived types were however not distinguished, and with a restriction of only one frame encoded per entry, many lexemes would appear in multiple entries. With a large import of words from *Bokmålsordboka*, the verb part of the NKL dictionary thus counts around 10 000 entries; adding adjectives and nouns, NKL hosts around 80 000 entries. Like TROLL, NKL physically consists of text files.

2.3 NorSource (‘Norwegian HPSG Resource Grammar')

As a so-called *Deep Computational Grammar*, NorSource sustains a *generic* parser (not restricted with regard to style of text or domain of use) representing wide lexical coverage, encoding linguistically well motivated morpho-syntactic and semantic analyses of nearly all aspects of the grammar, and applying this knowledge in the parsing process such that every parse reflects this knowledge. [7]

As mentioned, NorSource has as its formal and theoretical framework *Head-Driven Phrase Structure Grammar* (*HPSG*) (Pollard and Sag 1994, Sag et al. 2003), on which the computational project initiative *LinGO* at CSLI, Stanford, was started, using the *LKB platform* (Copestake 2002), which is a general platform with the format of typed feature-structures (TFS), and has integrated in it a format of semantic representation called *Minimal Recursion Semantics* (‘MRS’; cf. Copestake et al. 2005). Before year 2000 there were three grammars in this framework, viz. the *English Resource Grammar* ('ERG'), the Japanese grammar 'Jacy', and the German grammar 'GG'. Essential to the development of further grammars of this type was the *HPSG Grammar Matrix* (‘the *Matrix*’; see Bender et al. 2002, 2010), which was mainly based on ERG, and had its first phase of deployment during the EU-project *DeepThought* (2002-4). NorSource was the first grammar based on this platform, and the since then growing family of grammars (by now 10-12 well developed grammars) is now hosted by the DELPH-IN consortium.[8] The general system has since the inception undergone significant steps in increasing efficiency while maintaining the ‘depth’ towards this tradition of linguistic analysis is committed.[9]

At its start in 2001, NorSource adopted much of the general architecture of ERG, and has kept in step with DELPH-IN technical developments such as use of treebanking and adoption of the

---

[6] 1996-2001. Conducted at NTNU. Sponsored by: NFR, Telenor, and NTNU. Main publication: Nordgård 1998. Project responsible: Torbjørn Nordgård. NKL has been a resource both in commercial and academic applications. It has served in the build-up of the lexicons of *NorSource* (as mentioned, see also below) and the LFG grammar *NorGram* (http://clarino.uib.no/iness/xle-web), and in the construction of *Norsk Ordbank* http://www.edd.uio.no/perl/ search/search.cgi?appid=72&tabid=1106 .

[7] NorSource was started in 2001 and is still being maintained and developed, conducted at NTNU. It has been sponsored by EU, NFR, NTNU. For publications: see *References*. Online access, for description: http://typecraft.org/tc2wiki/Norwegian_HPSG_grammar_NorSource . Webdemo: http://regdili.hf.ntnu.no:8081/linguisticAce/parse
The NorSource code files are downloadable from: http://www.nb.no/sprakbanken/show?serial=sbr-32&lang=en
The system LKB as such can be downloaded from http://moin.delph-in.net/LkbTop.

[8] http://moin.delph-in.net/

[9] Early steps in developments toward speed and efficiency of DELPH-IN tools are described in Oepen and Carroll 2000; publications summarizing the progress up to now do not exist, but indications are presented at http://moin.delph-in.net/.

efficient processing platforms PET and ACE,[10] while LKB is for development mainly. NorSource has about 85,000 lexical entries, 250 syntactic rules, and 40 lexical rules (for derivation and inflection). In addition to preprocessing facilities following these platforms, NorSource has also developed facilities for unknown text, where especially names, terms and compound words constitute challenges. The system uses a webtagger[11] and a compound word splitter to identify and analyse the new items, based on information among the 85,000 already existing items and their inflectional properties. The newly identified words are put into an extra lexicon rather than the standard lexicon, in order to use less resources. Although from the outset focused on ‘deep’ properties of grammar, NorSource has thereby attained a reasonable degree of robustness for larger texts,[12] without compromising on ‘depth’ of analysis.

Content-wise, NorSource has gone through the following stages: Phase 1, *Grounding* phase (2001-04), Phase 2, *Semantic Expansion* (2005-07), Phase 3, *Cross-Linguistic Coding* (2008-10), and Phase 4, *‘Offsprings’ development* (2010-).[13] Phase 1 resided in the building of a basic core grammar around the Matrix skeleton, which includes the MRS system. This stage included the accommodation of the lexicons from TROLL and NKL, using the 'flat' format of verb derivations for concerns of efficiency of parsing, and also here with one frame per entry. In the tdl code used in LKB grammars, the counterpart of the entry in (1) is (2), reflecting the option with template ‘6’:

(2)
```
bruke_tv-reg := v-tr-suAg_obTh &
  [STEM <"bruke">,
  INFLECTION nonfstr,
  SYNSEM.LKEYS.KEYREL.PRED "_bruke_v_rel"].
```

Here ‘nonfstr’ stands for ’non-final stress’, as a key to inflectional rules. The feature ‘PRED’ introduces an identifier of the meaning of *bruke* (following a general strategy in HPSG and LFG which succeeds in keeping the meaning apart from the meaning of other items, but hardly characterizes the meaning beyond that).

Tying up with the earlier discussion, the item *v-tr-suAg_obTh* is the lexical type of this usage of *bruke*. It corresponds to the macro-index ‘6’ in (1), but here using a more transparent code reflecting the content of the type. This content is shown in the partial AVM in Figure 1 below, which reflects the general feature system in the Matrix grammars. The attribute RELS, on the path SYNSEM.LOCAL.CONT.RELS, is a *difference list*[14] which accumulates the information encoded under LKEYS.KEYREL for every word in the structure; this list turns up as the MRS representation of the structure, as will be illustrated further below in Figure 2.

What one can take the AVM in Figure 1 to illustrate is that the valency and semantic role information from the TROLL stage is directly carried on into the grammar formalism, and integrated in a formalism which lets this information play into a full-scale parser of the language. Note in particular the attribute ‘QVAL’, which supplements the standard valency feature ‘VAL’ used in HPSG with a representation of grammatical functions like what is used in LFG, and in the rkf specifications of TROLL.

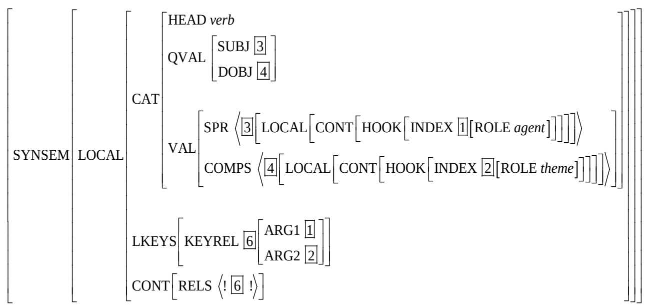

Figure 1. AVM reflecting the type *v-tr-suAg_obTh*

[10] Cf. http://moin.delph-in.net/

[11] http://regdili.hf.ntnu.no:8081/webtagger/tagger

[12] These are perhaps the main factors relative to which NorSource is significantly behind ERG, the leading grammar in the DELPH-IN consortium. In terms of size of grammar and depth of analysis, they probably come out even, with strengths in different domains. Since depth of analysis is a factor notoriously difficult to measure and compare across grammars, no ‘hard’ comparative figures can be presented here.

[13] Participants through these phases: Lars Hellan (project leader); directly engaged in development: Petter Haugereid (phase 1), Dorothee Beermann (phase 2), Ben Waldron (phase 3), Tore Bruland, Elias Aamot, Mads H. Sandøy (phase 4); as advisors: Dan Flickinger (throughout) and Woodley Packard (phase 4), as well as Ann Copestake and Francis Bond; further support (especially in phase 1) by Kaja Borthen, Jostein Ven, Siri Simonsen, Stephan Oepen, and Lars G. Johnsen.

[14] Cf. Copestake 2002.

An essential part of the parse result in an LKB grammar is the MRS representation, illustrated for a sentence with *bruke* as verb as defined in (2) and Figure 1 together:

$$
\begin{bmatrix}
\text{LTOP: h1} \\
\text{INDEX e2}\begin{bmatrix}\text{E}\begin{bmatrix}\text{TENSE: PRES}\end{bmatrix}\end{bmatrix} \\
\text{RELS:}\left\langle
\begin{array}{l}
\begin{bmatrix}\text{\_def\_q\_rel} \\ \text{LBL h5} \\ \text{ARG0 x4}\begin{bmatrix}\text{ROLE agent}\end{bmatrix} \\ \text{RSTR h6} \\ \text{BODY h7}\end{bmatrix},
\begin{bmatrix}\text{\_gutt\_n\_rel} \\ \text{LBL h3} \\ \text{ARG0 x4}\end{bmatrix},
\begin{bmatrix}\text{\_bruke\_v\_rel} \\ \text{LBL h8} \\ \text{ARG0 e2} \\ \text{ARG1 x4} \\ \text{ARG2 x9}\end{bmatrix}, \\
\begin{bmatrix}\text{\_def\_q\_rel} \\ \text{LBL h11} \\ \text{ARG0 x9}\begin{bmatrix}\text{ROLE theme}\end{bmatrix} \\ \text{RSTR h12} \\ \text{BODY h13}\end{bmatrix},
\begin{bmatrix}\text{\_pumpe\_n\_rel} \\ \text{LBL h10} \\ \text{ARG0 x9}\end{bmatrix}
\end{array}
\right\rangle \\
\text{HCONS:}\langle \text{h6 QEQ h3, h12 QEQ h10}\rangle
\end{bmatrix}
$$

Figure 2   MRS representation of *Gutten bruker pumpen ‘the boy uses the pump*)

It will be observed how the part (the ‘EP’ – ‘elementary predication’) displaying *bruke_v_rel* corresponds exactly to the information in (2) and Figure 1, and that the alignment of the variables with other items in the sentence is induced by the further equivalences expressed in Figure 1.

Representations like these are used as basis for *generation*, which produces the same sentence and possible alternatives which the grammar would assign the same MRS; an instance of the use of this facility is mentioned in 2.4 below. MRS is also used for MT purposes (with transfer operations over the word items in the MRS) and is in principle a candidate for serving into automatic reasoning.[15]

2.4 Norwegian Grammar Tutor[16]

This is an online L2-instruction tool along the lines of Bender et al. 2004, and Suppes et al. 2014, falling within the overall initiatives described in Heift and Schultze 2007: specific types of grammatical mistakes are accommodated by ‘mal-rules’ in an extended ‘mal’-version of the grammar, and parses involving such mal-phenomena are reported to the user as tutoring instructions. For each ‘mal’-parse, an MRS is constructed reflecting a corresponding sentence where the error in question does not occur, and from the MRS this sentence is generated as an example of a correct sentence with respect to the phenomenon in question. Figure 3 shows a screenshot where 10 ungrammatical sentences have been submitted to the system, and the responses for all of them are shown below, 8 receiving a diagnosis, 2 of them just deemed as ungrammatical (responses can also be called in Polish).

To reduce the possible confusion of many errors per sentence and to guarantee speedy processing, input sentences can be at most 10 words. The system is freely available online, and although all processed sentences are kept in a log, whereby evaluation of performance is possible, no track is currently kept of the users, whereby evaluation of impact on learning for any individual is currently not possible. The daily number of parses or generations varies in average between 100 and 200. By its ‘auto-generation’ of corrections and examples for freely chosen inputs, this is a promising design within ICALL.

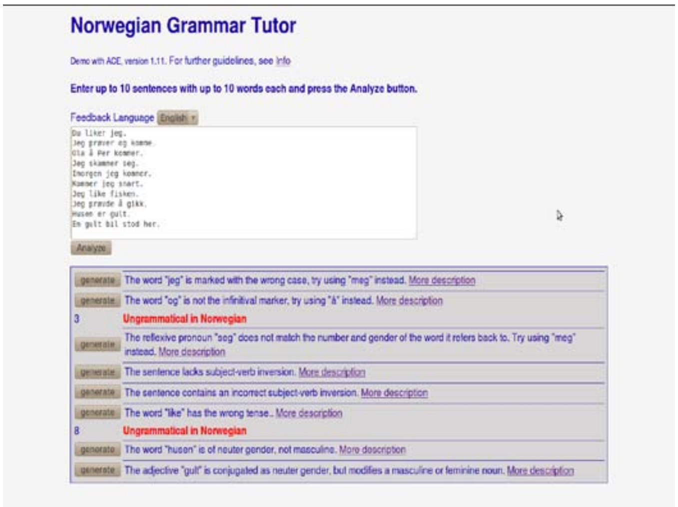

Figure 3   Screenshot showing error messages for 10 ungrammatical sentences

[15] Cf. Bruland 2013.

[16] Conducted at NTNU, sponsored by NTNU, started in 2011. Main publication: Hellan et al. 2013. Online access: A Norwegian Grammar Sparrer, for description, and for direct access: http://regdili.hf.ntnu.no:8081/studentAce/parse

2.5 A Multilingual Valence repository - *MultiVal* [17]

A monolingual online valency lexicon for Norwegian was constructed from NorSoure, taking into account the lexicon files and the lexical types of the grammar, such that in the online lexicon, each entry is partly based on the information in the relevant entry in the grammar lexicon, partly on the lexical type given there. The information encoded in the type is unfolded through a conversion script, exemplified in (3) below with one out of the nearly 300 rewrite rules. The leftmost item in this rule is a lexical type, which reflects both grammatical and semantic properties. This rule rewrites the type symbol 'v-ditr' ('ditransitive headed by verb'), into the syntactic argument structure (SAS) counterpart 'NP+NP+NP', the functional specification 'ditransitive', and the semantic specification of a three-place relation.

(3) v-ditr => SAS: "NP+NP+NP"
FCT: ditrans
SIT: ternaryRel

This information is available in the online interface, whereby exactly the amount of consolidated information available in the other members of the cluster is now available also in an online query interface.

NorSource being one of the grammars hosted in the Delph-In consortium[18] sharing the Matrix TFS system, the format of construction of online valency lexicons described can be applied to other DELPH-IN grammars as well, with some variation in the lexical type names used, but with the same types of information and categories of information for the parameters SAS, FCT and SIT for all the grammars. A next step in the development of the online valency lexicon has thereby been the extension to other languages, viz. the Spanish Resource Grammar SR, the Bulgarian grammar Burger, and the Ga grammar GaGram, to constitute a multilingual valency base with common categories across the languages. The resulting interface, and a search for verbs with the initial letter "s" and the functional valency category 'intransitive', is shown in the screenshot in Figure 4:

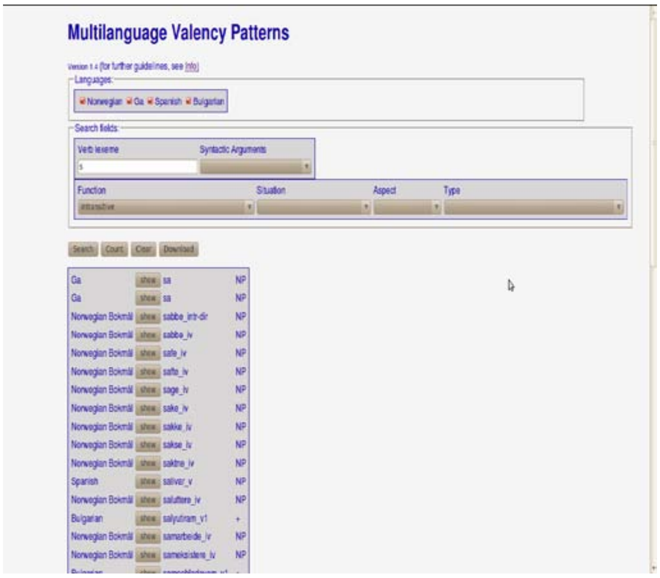

Figure 4 MultiVal screenshot of search for verbs with initial letter "s" and functional valency category 'intransitive'

At this point MultiVal is one of just three existing multilingual valency lexicons, the others[19] both with very different design, purpose and source of origin. Possible ways of aligning these should be explored, and for MultiVal, extension of the 'recruitment basis' beyond HPSG grammar resources is a natural aim, also going to non-grammar related monolingual valency resources,[20] and digital lexical resources like ToolBox lexicons.

3. Content: consistency and flexibility

In keeping with standard assumptions, factors such as the following should be addressed when analytic 'depth' is invoked relative to argument structure:

a. Syntactic argument structure, i.e., whether there is a subject, an object, a second/indirect object, etc., referred to as

[17] Developed at NTNU, 2013--. Database: Derby. Financed by NTNU. Main publication: Hellan et al. 2014. Online access: http://typecraft.org/tc2wiki/Multilingual_Verb_Valence_Lexicon , for description, and for direct access: http://regdili.hf.ntnu.no:8081/multilanguage_valence_demo/multivalence

[18] http://moin.delph-in.net/

[19] Viz., ContraGram: http://www.contragram.ugent.be/), and ValPaL: http://www.eva.mpg.de/lingua/valency/index.php.

[20] Examples include, apart from English resources such as FrameNet , VerbNet, PropBank, and Hanks' Corpus Pattern Analysis, for German, Evalbu (http://hypermedia2.ids-mannheim.de/evalbu/), Wortschatz (http://wortschatz.uni-leipzig.de/), and GermaNet (http://www.sfs.uni-tuebingen.de/lsd/verb_frames.shtml); for Czech Vallex (http://ucnk.ff.cuni.cz) and VerbaLex; for Polish, Walenty (http://clip.ipipan.waw.pl/Walenty).

grammatical functions, and the formal categories carrying them; the NorSource cluster here provides the standard inventory, from TROLL on.
b. Semantic argument structure, that is, how many participants are present in the situation depicted, and which roles they play (such as ‘agent’, ‘patient’, etc.); this we comment on below.
c. Linkage between syntactic and semantic argument structure, i.e., which grammatical functions express which roles, and possible roles not expressed; here also belong identity relations, part-whole relations, etc., between arguments; as exemplified in the view in Figure 1, linkage is clearly accommodated in NorSource.
d. Aspect and Aktionsart, that is, properties of a situation expressed by a sentence with the valence in question in terms of whether it is dynamic/stative, continuous/instantaneous, completed/ongoing, etc.; these factors are likewise accommodated.
e. Type of the situation expressed, in terms of some classificatory system; this aspect we address next.

Although all semantic information is syntactically linked, the NorSource architecture leaves a certain room open for exactly how detailed the semantic information will be. Still focusing on verbs, with as much as 10,000 entries, and migration of the information of these entries across all the applications present, it is clear that the format of encoding of information as such must be robust, and the codes must be simple. For instance, formats like csv or attribute-value are preferable, and the further that complex contents can be defined by simple expressions and throughout the systems be represented by these simple expressions, the better. The adoption of macros and types illustrated above instantiates this strategy; even if not exactly the same expressions or symbols are used in all the applications, translation between them – as when TROLL macros are translated into NorSource types, or NorSource types into MultiVal types – is straightforward as long as the contents on each side are the same.

This leads to the issue of semantic information. While grammatical categories are largely fairly closed and commonly agreed upon sets, inventories of roles and situation types are more, resp., very much more, open both in number and categories, and subject to much disagreement between frameworks as well as scholars (and even at the ‘intra-scholar’ level over time). What is needed here is therefore a ‘minimal’ specification level on which everybody can agree, and which easily lends itself to large scale specification, but relative to which it is possible to add descriptions of greater specificity when wished for, although such that the ‘minimal’ specification is then not obliterated, but preserved. Such a format of specification is included in the display in Figure 1, where on the feature path ‘LOCAL.CONT.HOOK.INDEX.ROLE’ of both arguments, a role label is provided. Here it is the type *index* which introduces the attribute ROLE, and *index* is also the value of the semantic attributes ARG1, ARG2, …[21] This means that ROLE always occurs inside of one of the ARG attributes, and can be seen as sub-specifications of what is encoded in the ARG attributes, as displayed in the AVM in (4):

(4)
$$\left[\textit{...KEYREL} \left[\begin{array}{l} \text{ARG1 index} \left[\text{ROLE agent} \right] \\ \text{ARG2 index} \left[\text{ROLE theme} \right] \end{array} \right] \right]$$

Leaving a role status underspecified can be done by using as value of ROLE simply a super-type of all the candidate role names, i.e., *role*.[22]

The ARG attributes are partly *role labels*, partly *enumeration markers*. As enumeration markers, they list the participants present in the situation expressed (including implicit ones), starting with ARG1, using ARG2 only if there is an ARG1, and using ARG3 only if there is an ARG2.[23] This is analogous to the conventional listing of arguments of an operator in logical notation, where in expressions like ‘P(x,y)’ one introduces a comma only if there is more than one argument; and distinct from the conventions in PropBank[24], where each ARGn has a specific role interpretation. As role markers, when there is more than one argument, they express something close to ‘proto’ roles, so that when there is an ARG1 and an ARG2, ARG1 is the role associated with emanation of force, and ARG2 is the ‘impacted’ part relative to the force; an ARG3 will then express a slightly less directly involved participant than the ARG2, such as the recipient or benefactive in a ditransitive sentence; in these contrasts, the ARGs have the same intuitive basis as Dowty’s (1991) proto-roles. When there is only one actant, it will be marked as ARG1, regardless of its role.[25]

The independence of ARGs from role specifications is reflected in the lexical type labels in that when ROLE is not specified, the last parts of the labels are simply omitted; for instance, while the type *v-tr-suAg_obTh* in Figure 1 has roles specified, the type for a transitive verb with no roles specified being simply *v-tr*. In this way, flexibility is matched between the TFS formalism and the simple code. In this code, by itself defined within a system called *Construction Labeling (CL)*[26] -

[21] In the TFS version here used (Copestake 2002), the conventions for types and attributes are summarized as follows: *[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).
[22] Roles have been only sparingly developed in NorSource, most systematically in an account of locative and directional PPs and adverbs, see, e.g., Hellan and Beermann 2005.
[23] In this design and its accompanying semantic conventions, the system is close to what is outlined in Tesnière 1959, pp. 105-115. (We are grateful to an anonymous reviewer for pointing out Tesnière’s crucial role in these articulations.)
[24] http://verbs.colorado.edu/~mpalmer/projects/verbnet.html
[25] Again, this is analogous to conventional logical notation. That the ARGs are not grammatical functions is illustrated by the circumstance that in a passive sentence, the (passive) subject will correspond to the ARG2 or ARG3 participant, not to the ARG1.
[26] Cf. Hellan (2008), Hellan and Dakubu (2010).

situation type is likewise something one can add in the string, and which can be specified under a specific attribute in the AVM.

The general interchangeability between the CL code and the grammar formalism is being exploited in one further initiative: among conceivable ways of improving the NorSource system, one approach relates to the complexity the formal code, another to what one may call the cross-linguistic validity of the constructs defined. These concerns are jointly addressed in the ongoing construction of a system called *TypeGram*, details of which are described at http://typecraft.org/tc2wiki/TypeGram.[27] [28] Crucial to the present themes is that when carrying out the relevant restructurings, we want to transfer all essential insights of the established resources into the new design. The CL code serves as a bridge in such a transfer, and is thus a key element in a forthcoming extension of the cluster.

4. Conclusion

No application in the cluster was purportedly designed with a view to supporting the other applications (except that the lexicon applications perhaps might have a parser as a possible employment), thus each one was created in its own right. None of them were computationally innovative, but rather based on solid techniques and platforms. The linguistic content was also solid and as 'deep' as any computational application can have it, but not theoretically innovative per se. In all parts, the applications can be easily understood by linguists and computational linguists, a circumstance which has allowed for a certain change of maintainers over time, and which makes the further sustainability and development of the resources a realistic prospect. It also opens for replicability for other languages, no matter whether the starting point would be lexical resources, valency lexica, or parsers. Replicability has indeed been partly demonstrated within the Delph-in resources, but there is no reason why it should be limited thus, as long as the resources in question are consistently and transparently constructed. Through the factors we have described, and doubtlessly others as well, one could thereby start to define a methodology of creation of resource clusters, of which the one we have now described would be an instantiation.

References

Bender, E. M., Drellishak, S., Fokkens, A., Poulson, L. and Saleem, S. (2010). Grammar Customization. In *Research on Language & Computation*, Volume 8, Number 1, 23-72.

Bruland, T. (2011). Creating TypeGram data from TypeCraft. Presentation at *India 2011*, NTNU.

Bruland, T. (2013). Building World Event Representations From Linguistic Representations. PhD dissertation, NTNU.

Copestake, A. (2002). *Implementing Typed Feature Structure Grammars*. CSLI Publications.

Copestake, A., D. Flickinger, I. Sag and C. Pollard. (2005). Minimal Recursion Semantics: an Introduction. *Journal of Research on Language and Computation*. 281-332.

Heift, T., and M. Schulze. (2007). *Errors and Intelligence in Computer-Assisted Language Learning: Parsers and Pedagogues*. Routledge, New York.

Hellan, L. (2008) From Grammar-Independent Construction Enumeration to Lexical Types in Computational Grammars. COLING (http://www.aclweb.org/anthology-new/W/W08/#1700).

Hellan, L., Johnsen, L. and Pitz, A. TROLL. Ms, University of Trondheim. 1989

Hellan, L. and D. Beermann. 2005. Classification of Prepositional Senses for Deep Grammar Applications. In: Kordoni, V. and A. Villavicencio (eds.) *Proceedings of the 2nd ACL-Sigsem Workshop on The Linguistic Dimensions of Prepositions and their Use in Computational Linguistics Formalisms and Applications*, Colchester, UK.

Hellan, L. and Dakubu, M.E.K. (2010). *Identifying Verb Constructions Cross-linguistically*. SLAVOB series 6.3, Univ. of Ghana.

Hellan, L., Bruland, T., Aamot, E., Sandøy, M.H. (2013): A Grammar Sparrer for Norwegian. *Proceedings of NoDaLiDa* 2013.

Hellan, L., D. Beermann, T. Bruland, M.E.K. Dakubu, and M. Marimon (2014) MultiVal: Towards a multilingual valence lexicon. LREC 2014.

Hellan, L. and Beermann, D. (2014) Inducing grammars from IGT. In Z. Vetulani and J. Mariani (eds.) *Human Language Technologies as a Challenge for Computer Science and Linguistics*. Springer.

Nordgård, T. (1998) "Norwegian Computational Lexicon (NorKompLeks)". *Proceedings of NoDaLiDa* 1998.

Oepen, S. and Carroll, J. (2000). Performance profiling for parser engineering. *Natural Language Engineering, 6. Issue on Efficient Processing with HPSG*, 81 – 97.

Pollard, C. and Sag, I. (1994). *Head-Driven Phrase Structure Grammar.* Chicago University Press.

Sag, I., Wasow, T. and Bender, E. *Syntactic Theory*. A formal introduction. Stanford: CSLI Publications.

[27] This site also provides a download site for the system, running on a very simple version of LKB.

[28] A further application of TypeGram is in *grammar induction*, combined with the morphological annotation system *TypeCraft* (http://typecraft.org/tc2wiki/Main_Page) ; see Hellan and Beermann (2014), Bruland (2011).

Suppes, P, T. Liang, E.E. Macken and D. Flickinger (2014) “Positive technological and negative pre-test-score effects in a four-year assessment of low socioeconomic status K-8 student learning in computer-based Math and Language Arts courses ", Computers & Education, 71, pp. 23-32.
Tesnière, L. (1959). *Éleménts de syntaxe structurale*. Paris: Klincksieck.