Unique lexical entries in a subconstructional grammar

Petter Haugereid
Bergen University College
petter.haugereid@hib.no

Abstract

Function words like prepositions, adverbs, particles, and complementizers may be assigned more than one category due to the different functions they can have. In this paper I present an approach that assumes unique lexical entries for words that are assigned more than one category. I will focus on prepositions and how they may function as heads of modifying PPs, selected prepositions, or as particles.

1 Introduction

The Norwegian LFG grammar NorGram (Dyvik, 2000) has a long list of lexical entries where one form is assigned more than one category. Table 1 shows for each pair of a selected set of categories, the number of word forms that are assigned both categories. There are 43 adjectives (A) that also can be degree adverbs (ADVdeg). One of them, *merkelig*, is illustrated in (1) as an adjective (1a)) and as a degree adverb ((1b)).

(1) a. Det var en merkelig følelse.
it was a strange feeling
It was a strange feeling.

b. Rommet blir merkelig stille.
room-DEF becomes oddly quiet
The room becomes oddly quiet.

As the table shows, many prepositions also can be adverbs (66), particles (PRT) (38) and selected prepositions (Psel) (53). One of them, *unna* ('away'), is exemplified in (2) where it is an adverb ((2a)), a preposition ((2b)), a particle ((2c)), and a selected preposition ((2d)).

(2) a. Han kjørte unna.
he drove away
He drove out of the way.

b. De gikk unna flammene.
they walked away flames-DEF
They walked away from the flames.

c. Han smatt unna.
he escaped away
He escaped.

d. Han sluntret unna pliktene sine.
he idled away duties his
He shirked his duties.

The most obvious way to treat these words in the lexicon, is to create separate lexical items for each category assigned to it. This is not entirely satisfying, given the the intuition that most of them share a meaning. The aim of this paper is to show that these forms can be assigned unique lexical items that will be compatible with the functions that are required from them.

2 Multiple lexical items

There are several reasons for assuming several lexical entries for one form, specially within a framework like HPSG where there are no derivations and no information gets lost. In particular, this holds for semantic relations. Once a semantic relation is entered on the RELS list by a lexical item, a lexical rule or a syntactic rule, the compositional nature of HPSG requires that this relation also is a part of the semantic representation of the phrase that the lexical item, lexical rule or rule is a part of. So if the noun *tabs*

	A	ADV	ADVdeg	ADVs	Cadv	P	PRT	Psel
Psel	0	38	1	0	4	53	31	-
PRT	5	39	2	3	3	38	-	
P	5	66	1	3	9	-		
Cadv	4	8	4	7	-			
ADVs	6	15	31	-				
ADVdeg	43	15	-					
ADV	13	-						
A	-							

Table 1: Pairing of categories and the number of words assigned to both categories in NorGram.

introduces a relation _*tab_n_rel* and the preposition *on* introduces a relation _*on_p_rel*, these relations have to appear in the resulting semantic representation. This is a little problematic in the case of idioms like *He kept tabs on the competition.* The composition of semantic relations requires the *tabs_n_rel* and the *on_p_rel* to be a part of the resulting representation, even though the idiomatic meaning is to *observe.*

Sag *et al.* (2003, 347–355) solves this problem by assuming a special lexical entry for the idiomatic version of *keep* that has three items on the SUBCAT list; (i) the NP subject, (ii) an idiomatic noun *tabs*, and (iii) a constituent marked by the preposition *on.* (See (3).) The relation of the idiomatic version of *keep* is *observe*, and the idiomatic noun *tabs* and the selected preposition *on* are both assumed to be semantically empty. This gives the intended *oberve*-relation between the OBSERVER (*he*) and the OBSERVED (*the competition*).

(3)
$$\left[\begin{array}{l} \textit{ptv-lxm} \\ \text{STEM} \left\langle \text{keep} \right\rangle \\ \text{ARG-ST} \left\langle \text{NP}_i\ , \left[\text{FORM tabs}\right], \left[\begin{array}{l}\text{FORM on}\\ \text{INDEX } j\end{array}\right] \right\rangle \\ \text{SEM} \left[\begin{array}{l} \text{INDEX } s \\ \text{RESTR} \left\langle \left[\begin{array}{ll} \text{RELN} & \textbf{observe} \\ \text{SIT} & s \\ \text{OBSERVER} & i \\ \text{OBSERVED} & j \end{array}\right] \right\rangle \end{array}\right] \end{array}\right]$$

The problem with this approach is that in addition to an idiomatic and non-idiomatic version of the verb *keep*, it also presupposes an empty preposition (in addition to the standard preposition with an _*on_p_rel*) and an idiomatic noun *tabs* in addition to the regular word *tabs* with the relation _*tab_n_rel.*

3 Incremental parsing with left-branching structures

Instead of assuming that lexical entries are specific to the extent that multiple lexical entries are needed for the same form (where the basic meaning is the same), I suggest an approach where lexical items are allowed to be underspecified with regard to what function they fill. This approach depends on three factors; (i) underspecification, (ii) multiple inheritance, and (iii) category specific phrase structure rules that access the words in question. While the the first two factors are common practice in HPSG, the third factor is an innovation. It can be achieved by means of incremantal parsing with left-branching structures.

In my approach I assume that parse trees are distinct from constituent trees, and that the parse trees are completely left-branching (Haugereid and Morey, 2012). The strategy is that of a shift reduce parser, namely to use a stack to store information about constituents that are not completed. This gives us parse trees without center-embeddings, and allows for incremental processing of sentences.

There are mainly three types of rules: (i) *embedding rules*, that initiate a constituent, (ii) *attaching rules*, that add words to an already initiated constituent, and (iii) *popping rules*, that mark the completion of a constituent.

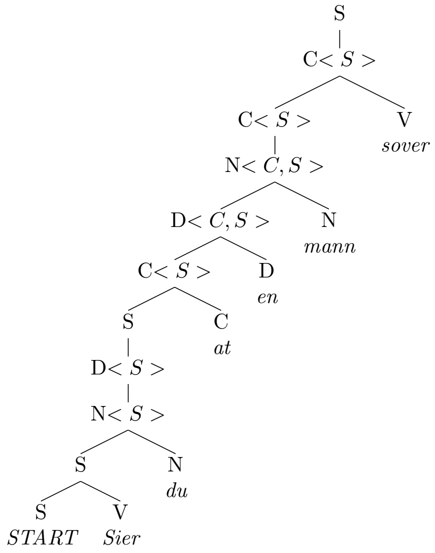

Figure 1: Parse tree

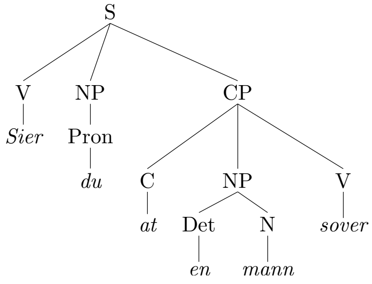

Figure 2: Constituent tree

The syntactic structure is built incrementally, word by word, as shown in Figure 1. The analysis starts with a START sign in the bottom left. The START sign is combined with the first word of the sentence with a binary rule, in this case the rule for attaching the verb *Sier* (an attaching rule). The structure that now consists of the start sign and the first word (represented by the node S) is then combined with the next word *du* with a rule that initiates nominal constituents (an embedding rule) (N<S>). The features of the *S* are then put on a stack. The next rule is a unary rule that adds a quantifier relation (D<S>), and the following rule is a rule that pops the features of the start symbol from the stack, and the category goes back to S. Similar embedding, attaching and popping rules apply for the rest of the clause. The constituent tree is formed simply by adding a left bracket when there is an embedding rule and a right bracket when there is a popping rule. The constituent tree corresponding to the parse tree in Figure 1 is shown in Figure 2.

This left-branching design opens for subconstructions that attach single words, and not full constituents, and it gives us the possibility to tailor subconstructions for every category of words, and the words attached by the subconstructions are allowed to be more or less specific.

4 Analysis of prepositions as unique lexical entries

In this section I will focus on prepositions and show how a preposition can be attributed one lexical entry that accounts for all its functions. It is allowed by a combination of the constructionalist approach sketched in Section 3, underspecification, and the exploitation of types. The analysis is implemented in an HPSG-like grammar of Norwegian within the LKB system (Copestake, 2001).

A preposition like *on* can be both a particle (*I logged on*) and a selected preposition (*He relied on the kindness of strangers*/ *We kept tabs on our checking account*). In addition, it can also be a regular preposition as in *He sleeps on the floor.*

My approach to prepositions is inspired by the treatment of particles and selected prepositions in the English Resource Grammar (ERG) (Flickinger, 2000), where the lexical entry for *on* as a particle and selected preposition is shown in (4).

(4)
$$\begin{bmatrix} \text{ORTH} \left\langle \text{”on”} \right\rangle \\ \text{CAT} \begin{bmatrix} \text{HEAD} \begin{bmatrix} \textit{prep} \\ \text{MOD} \langle\rangle \end{bmatrix} \\ \text{VAL|COMPS} \left\langle \begin{bmatrix} \textit{synsem} \\ \text{CAT|HEAD } \textit{nom} \\ \text{CONT|HOOK } \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{CONT} \begin{bmatrix} \text{HOOK } \boxed{1} \\ \text{RELS} \left\langle !! \right\rangle \end{bmatrix} \\ \text{KEYREL} \begin{bmatrix} \textit{basic\_ arg12\_ relation} \\ \text{PRED } \textit{\_ on\_ p\_ sel\_ rel} \end{bmatrix} \end{bmatrix}$$

The ERG lexical entry for selected prepositions/particles has an empty RELS list, which means it is semantically empty. Still, it has specified a KEYREL with a PRED value (*_on_p_sel_rel*) that will be required by the verb that selects it. But this relation does not end up on the RELS list.

My approach is similar in that I assume a lexical entry with an empty RELS list. (See the lexical entry for *på* ('on') in (5).) It also has a relation as value of KEYREL, but the PRED value is an underspecified type, *_på_prd*, which allows it to function as a normal preposition, as a selected preposition, and as a particle.

(5)
$$\begin{bmatrix} \textit{prep-word} & \\ \text{ORTH} & \left\langle \text{"på"} \right\rangle \\ \text{CAT} & \left[\text{HEAD } \textit{prep}\right] \\ \text{CONT} & \left[\text{RELS } \left\langle !! \right\rangle\right] \\ \text{KEYREL} & \left[\text{PRED } \textit{\_på\_prd}\right] \end{bmatrix}$$

I can do this, firstly, because the PRED value is underspecified, which means that it is compatible with different relations as *_på_p_rel* (regular preposition relation) and all predicates that include *på* as a part of a complex predicate, like *_fokusere*på_14_rel* ('focus on') and *_logge-på_1_rel* ('log on'). Secondly, I use phrasal subconstructions, which makes it possible to decompose argument frames and predicates and let each sign of the grammar, be it a lexical item, an inflectional rule, or a syntactic rule, only contribute that piece of information that positively can be attributed to it, even if it is underspecified information. When the signs are put together, the pieces of information contributed by each sign about the argument frame and the predicate are unified, and the predicate is determined. The simplified type hierarchy in Figure 3 shows how the type *_på_prd* is compatible with the predicates *_logge-på_1_rel*, *_fokusere*på_14_rel*, and *_på_p_rel*.[1]

It is the function *på* has in the clause that determines which predicate it will end up with. If it functions as a particle of *logge* ('log'), *_på_prd* will be unified with the PRED value of *logge* (*logge_v*), and the resulting relation will be *_logge-på_1_rel*. If it functions as a selected preposition of *fokusere* ('focus'), *_på_prd* will be unified with *_fokusere_v*, yielding the predicate *_fokusere*på_14_rel*. And if it functions as a modifier, *_på_prd* will be unified with the type *mod+*, which gives the predicate *_på_p_rel*.

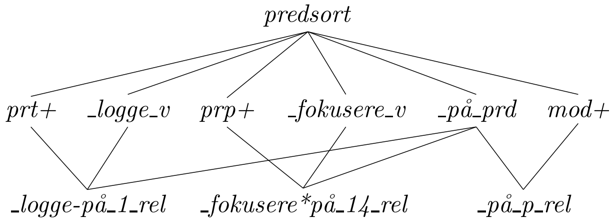

Figure 3: Type hierarchy of pred values of *på* ('on')

The subconstruction rule that attaches particles is given in Figure 4. It unifies the KEYREL value of the structure built so far (the first daughter) with that of the particle, and also the mother. It marks the PART value of the first daughter as *prt+*, and this value is unified with that of KEYREL|PRED. This ensures that *på* is interpreted as a particle.

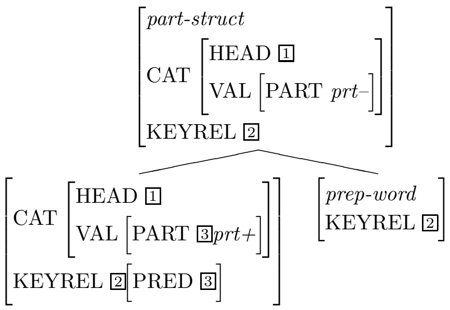

Figure 4: Rule for attaching particles

Similar to this rule attaching particles, the grammar also has a rule *marker-struct* that attaches selected prepositions.

The subconstruction rule for attaching verbs (*vbl-struct*) is shown in Figure 5. It selects the verb via the VBL feature, and the VBL requirement of the verb is transferred to the mother. Like the subconstruction rules for particles and prepositions, this rule unifies the KEYREL value of the structure built so far (the first daughter)

[1] The predicate names also indicate the number of arguments as well as their function. This is discussed in Haugereid (2014).

with that of the attached word (the verb), and the mother.

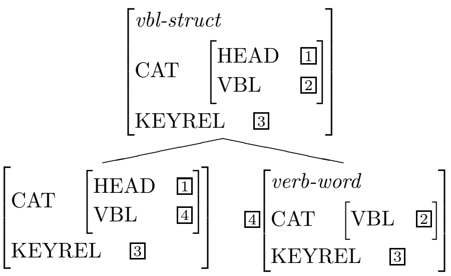

Figure 5: Rule for attaching verbs

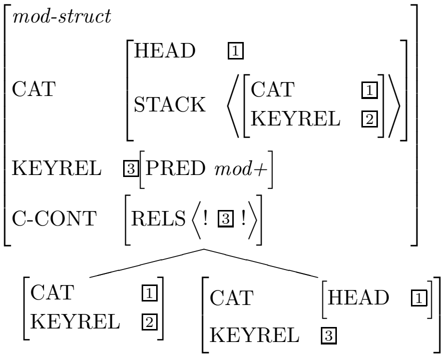

Figure 6: Embedding rule for attaching modifiers

The unification of KEYREL values in *part-struct* and *vbl-struct* ensures that when they apply in the same clause, the PRED values of the verb and the particle have to unify. Only the combinations of verb predicate and preposition/particle predicate that are defined in the type hierarchy are licenced by the grammar.

The modifier rule is given in Figure 6. It is an embedding rule, which means that the key features of the structure built so far (here, the CAT and the KEYREL of the first daughter) are put on a STACK in the mother, and the HEAD and the KEYREL features of the word initiating the modifying constituent are unified with those of the mother. The KEYREL of the modifier is entered onto the C-CONT|RELS list. In addition, its PRED value is unified with the *mod+* type, which means that if the word initiating the modifying constituent is the preposition *på*, its PRED value *_på_prd* will be unified with the type *mod+*, yielding the PRED value *_på_p_rel*, which appears in the semantic representation of the sentence.

Also other categories are treated in the same fashion. Nouns are not specified with a relation on the RELS list. Like the prepositions, their relation is specified as value of KEYREL, and the relation is entered on the RELS list when the words are added by their respective rules. This allows us to have special subconstructions for idiom nouns, like *tabs* in *keep tabs on*, that rather than treating the relation of the noun as a separate relation by entering it on the RELS list, unifies its predicate with the predicate of the verb (*keep*) and the preposition (*on*), resulting in a single idiom predicate.

The aim is to extend this analysis also to other categories, like adjectives that can be degree adverbs (see (1)), and complementizers that can be prepositions or adverbs. I want to develop a grammar that ultimately has unique lexical entries for all the words in the lexicon, regardless of whether they are content words or function words.

References

Copestake, A. (2001). *Implementing Typed Feature Structure Grammars*. CSLI Lecture Notes. Center for the Study of Language and Information, Stanford.

Dyvik, H. (2000). Nødvendige noder i norsk: Grunntrekk i en leksikalsk-funksjonell beskrivelse av norsk syntaks. In Ø. Andersen, K. Fløttum, and T. Kinn, editors, *Menneske, språk og felleskap*. Novus forlag.

Flickinger, D. P. (2000). On building a more efficient grammar by exploiting types. *Natural Language Engineering*, **6**(1), 15–28.

Haugereid, P. (2014). Vp idioms in norwegian: A subconstructional approach. In S. Müller, editor, *Proceedings of the 21st International Conference on Head-Driven Phrase Structure Grammar, University at Buffalo*, pages 83–102, Stanford, CA. CSLI Publications.

Haugereid, P. and Morey, M. (2012). A left-branching grammar design for incremental parsing. In S. Müller, editor, *Proceedings of* *the 19th International Conference on Head-Driven Phrase Structure Grammar, Chungnam National University Daejeon*, pages 181–194.

Sag, I. A., Wasow, T., and Bender, E. M. (2003). *Syntactic Theory: A Formal Introduction*. CSLI Publications, Stanford, 2 edition.