Long distance dependencies and basic clause structure in Norwegian

Petter Haugereid
Norwegian University of Science and Technology
Dragvoll, N-7491 Trondheim
petterha@hf.ntnu.no.

Abstract

This paper presents an HPSG approach to long distance dependencies in Norwegian, where the extraction site is assumed to dominate the extracted item. The approach involves a radically new analysis of basic main clause and subordinate clause structures. The analysis is extended to Irish, where complementizers register the extraction path.

1 Introduction

In frameworks like GPSG (Gazdar et al., 1985, 137-168), HPSG (Pollard and Sag, 1994, 157-207 and 376-388) and (Sag et al., 2003, 427-452), LFG (Bresnan, 2001, 180-208) and CCG (Steedman, 2000), the extracted item of a long distance dependency is assumed to be filled in at the top of the tree. The way an element is extracted, varies from framework to framework.

In the HPSG literature there are two main approaches to long distance dependencies, pointed out by Levine (2003). In one approach, the *trace* approach (Pollard and Sag, 1994, 157-207), it is assumed that an empty category is taken as an argument and introduces an element on the SLASH list that is transferred up the tree until it is filled in by the head filler rule. In a more recent approach, the lexical approach (Bouma et al., 2001), it is assumed that a lexical head gathers the slashes from its arguments and adjuncts by means of relational constraints. Then the SLASH value is transferred up the tree until the head filler rule fills it in.

In the approach I am suggesting in this paper, the extraction site is assumed to dominate the filler rule. I am going to show how this single assumption gives a natural account of the registering of the extraction path in languages like Irish.

2 The new approach

Long distance dependencies are accounted for by means of three operations. This is illustrated in Figure 1. First, there is a filler rule at the bottom of the tree (the VP/NP node[1]). Second, the SLASH list is copied up the tree. This is done by unifying the SLASH value of the first daughter with that of the mother. Third, the filled in element is extracted. This is done by means of a unary extraction rule (node VP2).

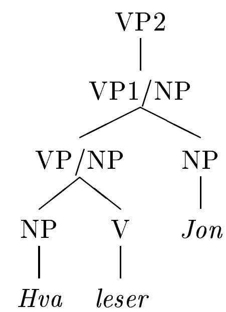

Figure 1: Analysis of *Hva leser Jon* ('What does John read')

2.1 The filler rule

The filler rule is given in Figure 2.[2] Here it is illustrated how the element on the SLASH list of the mother is unified with the LOCAL value of the first daughter (see tag 2). The SLASH list of the head daughter is empty.

2.2 The extraction rule

The extraction rule for argument 2 is given in Figure 3. Here, the LOCAL value of ARG2 (tag 2) is unified with the SLASH element of the daughter. The SLASH list of the mother is empty. The fact that argument 2 is extracted by this rule is marked by switching the LINK value from *arg2+* in the daughter to *arg2−* in the mother. The grammar

[1]The label VP is used for phrases with head value *verb*. The number on a node indicate that the rule is a valence rule, and also what kind of argument that was realized. The numbers correspond in general to the roles found in the initial stratum in Relational Grammar (Blake, 1990). Instead of assuming valence features such as SPR and COMPS, the valence features ARG1, ARG2, ARG3 and ARG4 are employed (Søgaard and Haugereid, 2005, 196–197).

[2]The feature SYNSEM is left out in order to make the figures as small as possible.

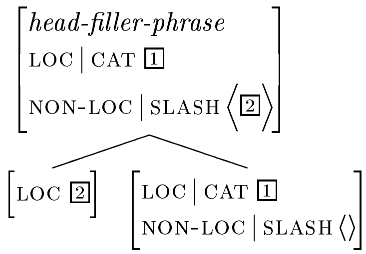

Figure 2: Head-filler rule

has one extraction rule for each kind of argument (five in all), and one for adjuncts.

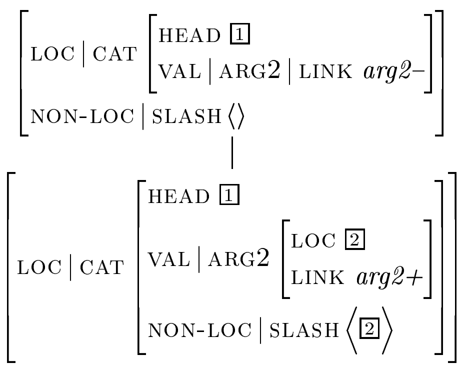

Figure 3: Arg2 extraction rule

It is assumed in this approach that also subjects are extracted. This is illustrated in Figure 4, where the rule VP1 extracts the subject *han* ('he'). The head filler rule (VP/NP) is the daughter of the extraction rule.

2.3 Floating quantifiers

One reason for extracting the subject is that this makes it possible to account for so-called "floating quantifiers" (Sportiche, 1988) in Norwegian. The assumption is that floating quantifiers appear where the extraction has taken place. The floating quantifier of a subject in a Norwegian main clause appears after the finite verb, as the data in (1) indicate.[3] In (1a) *alle* ('all') appears between the finite verb and the direct object (the position for the subject in a yes-no question). In (1b) *alle* is positioned before the finite verb, and the sentence is ungrammatical. In (1c) *alle* is positioned after the direct object, and the sentence is ungrammatical.

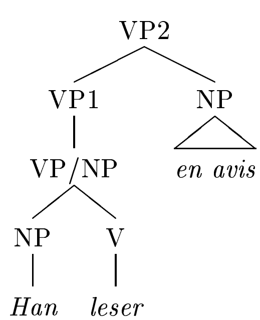

Figure 4: Analysis of *Han leser en avis* ('He reads a newspaper')

(1) a. Barna leser alle en bok.
children-DEF read all a book
'The children all read a book.'

b. * Barna alle leser en bok.
children-DEF all read a book

c. * Barna leser en bok alle.
children-DEF read a book all

The data in (2) show that the floating quantifier of the object, *begge* ('both'), appears in the "canonical" position after the subject (see 2a). If *begge* is attached to the extracted object ((2b)) or comes in the position after the finite verb ((2c)), the sentence is ungrammatical.

(2) a. Bøkene leser jeg begge på to timer.
books-DEF read I both on two hours
'The books I read both in two hours.'

b. * Bøkene begge leser jeg på to timer.
books-DEF both read I on two hours

c. * Bøkene leser begge jeg på to timer.
books-DEF read both I on two hours

Given the analysis where the subject is assumed to be extracted, it is possible to assume a rule that extracts the subject and at the same time realizes the "floating quantifier," as illustrated in Figure 5. Here the rule labeled VP enters an element in the SLASH list of the daughter, and realizes the "floating quantifier" *alle*.

2.4 The *merge* rule

Verbs that are *not* heading the projection are accounted for by means of a rule called *merge-rule*.

[3]Faarlund et al. (1997, 920–922) has more examples of floating quantifiers of subjects in Norwegian. Exceptional cases occur in connection to light (or weak) pronouns, like in (i) (so-called "Object Shift"). The floating quantifier here occurs in the position after the object ((ia)), instead of before the object ((ib)).

(i) a. Barna leser den alle.
children-DEF read it-LIGHT all
'The children all read it.'

b. * Barna leser alle den.
children-DEF read all it-LIGHT

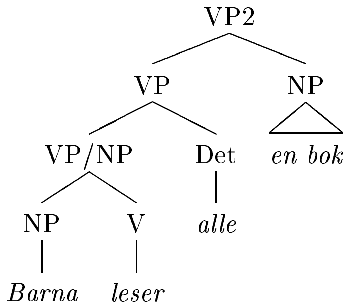

Figure 5: Analysis of *Barna leser alle en bok* ('The children all read a book')

This rule will attach the main verb to the projection in cases where an auxiliary or a complementizer heads the projection. This is illustrated in Figure 6, where the auxiliary *har* ('has') heads the projection, and the main verb *lest* ('read') merges with the auxiliary projection in the node AUXP.

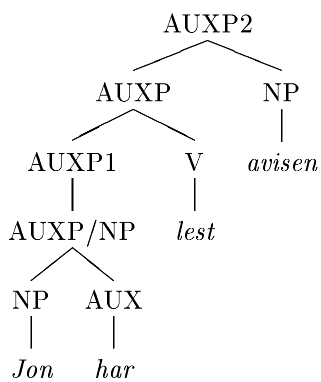

Figure 6: Analysis of *Jon har lest avisen* ('Jon has read the newspaper')

The *merge rule* is given in Figure 7. The illustration shows that the first daughter is the head daughter (see tag [1]). The valence values of the two daughters are unified with that of the mother (tag [2]).[4] The MERGE value of the first daughter is unified with the second daughter (see tag [4]), and the MERGE value of the second daughter is unified with the MERGE value of the mother (tag [3]).

2.5 Subordinate clauses

Subordinate clauses are accounted for by means of a particular construction, the binary complementizer rule. This construction takes as its first daughter a projection that can take a subordinate clause as its second argument. The second

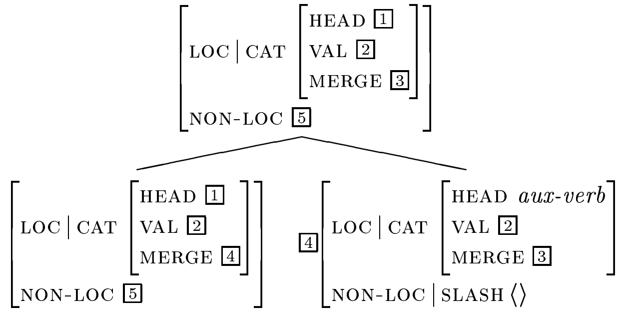

Figure 7: Merge rule

daughter of the construction is a complementizer. This is illustrated in Figure 8, where the node CP combines a verb projection and a complementizer. With such an approach to subordinate clauses, it is possible to maintain the assumption that the extraction site dominates the gap.

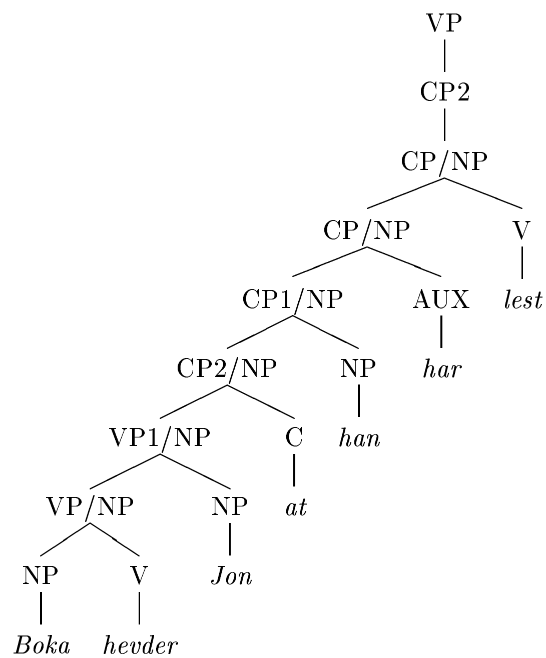

Figure 8: Analysis of *Boka hevder Jon at han har lest* ('The book, John claims that he has read')

The tree in Figure 8 shows that the complementizer (C) becomes the head of the structure, and that the following words attach to the C projection. At the top of the tree, the structure is turned back into a V projection. What happens in this analysis is that most of the structure of the VP sister of C is put in a stack in CP2/NP. This structure is then popped at the top of the tree (VP). Figure 9 and 10 show in detail how this works.

The "stacking" and "popping" mechanism allows for several embeddings into subordinate clauses. The function of the pop rule is to arrive at the main clause level again after entering a subordinate clause. The pop rule allows for the

[4] The illustration of how valence information is merged is simplified in this presentation. In a grammar implementation of this analysis, I am using two valence features in order to give a constructional account of passive. But as long as passive is not involved, the values of VAL will always be unified, so this is not relevant for presentation given here.

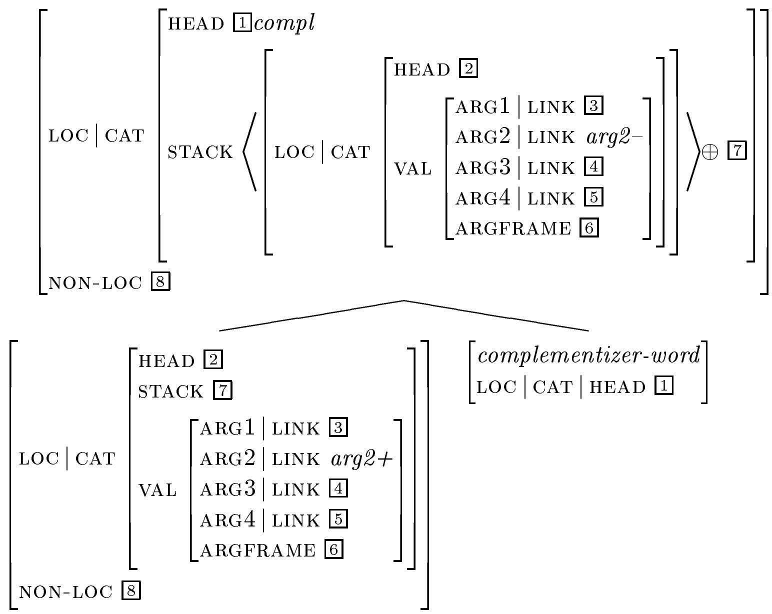

Figure 9: Binary complementizer rule

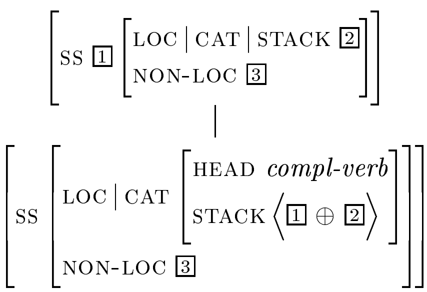

Figure 10: Pop rule

expected PP attachments, as the trees in Figure 11 and 12 show. In Figure 11, the PP attaches inside the subordinate clause, while in Figure 12, the PP attaches at main clause level.

3 Registering of extraction path

One argument for letting the extraction site dominate the filler rule is data from languages where the extraction path is registered by verbs or complementizers. The Irish data in (3) (originally from McCloskey (1979)) are used by Hukari and Levine (1995) and Sag (2005) among others to illustrate this fact. In Irish, the choice of complementizer reflects whether the complementizer intervenes between an extraction site and the filler or not. The complementizer *goN* is not on the extraction path, while the complementizer *aL is* on the extraction path. In (3a) there is no extraction taking place, so the complementizer *goN* is used. In (3b) there are two complementizers on the extraction path. Both of them *aL*. And in (3c) there are three complementizers, all of them *aL*, on the extraction path. (3d) is an example of an NP with

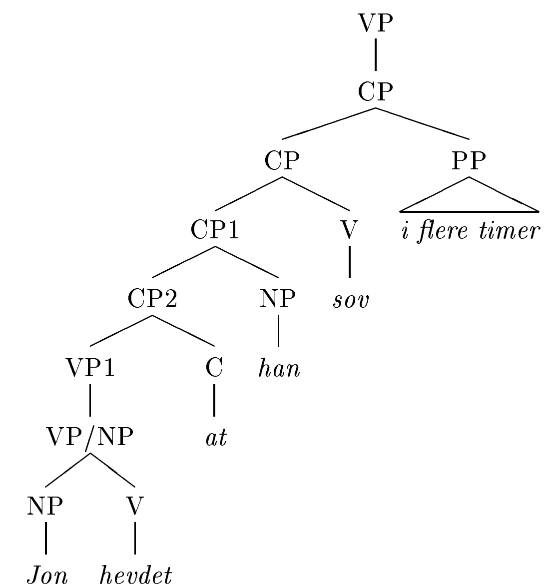

Figure 11: Analysis of *Jon hevdet at han sov i flere timer* ('John claimed that he had slept for several hours'). PP attachment to subordinate clause.

two complementizers, but where only one is on the extraction path. The complementizer on the extraction path is *aL* and the one occurring after the extraction site is *goN*. (3e) has three complementizers. Two on the extraction path (both *aL*), and one after the extraction site (*goN*).

The element that is extracted does not have to be a complement. It can also be an adjunct.

(3)	a. Dúirt	mé	**gurL**	shíl	mé	**goN**
	said	I	goN.PAST	thought	I	COMP

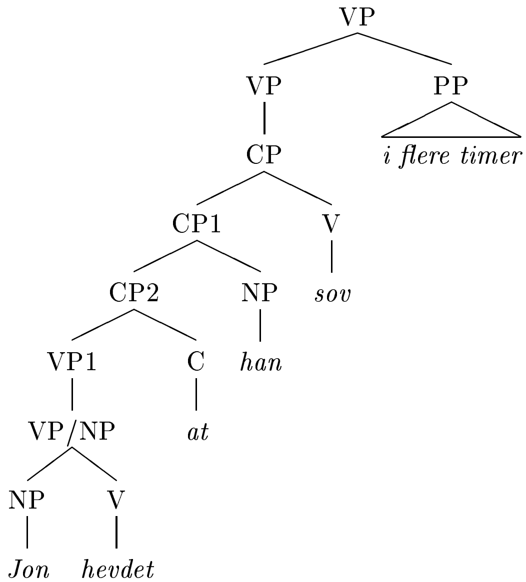

Figure 12: Analysis of *Jon hevdet at han sov i flere timer* ('John claimed that he had slept for several hours'). PP attachment to main clause.

mbeadh sé ann.
would-be he there

'I said that I thought that he would be there.'

b. an fear **aL** shíl mé **aL**
the man COMP thought I COMP
bheadh _ ann
would-be _ there

'the man that I thought would be there'

c. an fear **aL** dúirt mé **aL**
the man COMP said I COMP
shíl mé **aL** bheadh ann
thought I COMP would-be there

'the man that I said I thought would be there'

d. an fear **aL** shíl _ **goN**
[the man]$_j$ COMP thought _ COMP
mbeadh sé ann
would-be he$_j$ there

'[the man]$_j$ that thought he$_j$ would be there'

e. an fear **aL** dúirt sé **aL**
the man COMP said he COMP
shíl _ **goN** mbeadh sé ann
thought _ COMP would-be he there

'the man that he said thought he would be there'

In an approach where the extraction site dominates the filler, this kind of data can be accounted for, since the mother of the complementizer can tell whether the complementizer is on the extraction path or not. So the complementizer only has to "agree" with its mother. In all the other approaches, where the filler is on the top of the tree, there is no straightforward account of the extraction path facts. In the *trace* account of Pollard and Sag (1994) the non-empty SLASH value is only accessible above the extraction site.

Especially adjunct extraction is difficult, since adjuncts normally do not appear in the subcat frame of the verb. Bouma et al. (2001) suggest to collect all the dependents of a verb (arguments and adjuncts) on a DEPS list, and then, by means of relational constraints, letting the dependent that is extracted appear on the SLASH list of the verb. The SLASH then follows the head to the top of the tree, where it is filled in. Sag (2005) uses a lexical rule to achieve the same. In the approach suggested in this paper, no extra machinery is needed as long as one assumption holds, namely that the extraction site dominates the filler.

The tree in Figure 13 shows how the NP in (3e) can be analyzed. Note that the mothers of the two *aL*-complementizers have a non-empty SLASH list, while the mother of the *goN*-complementizer has an empty SLASH list.

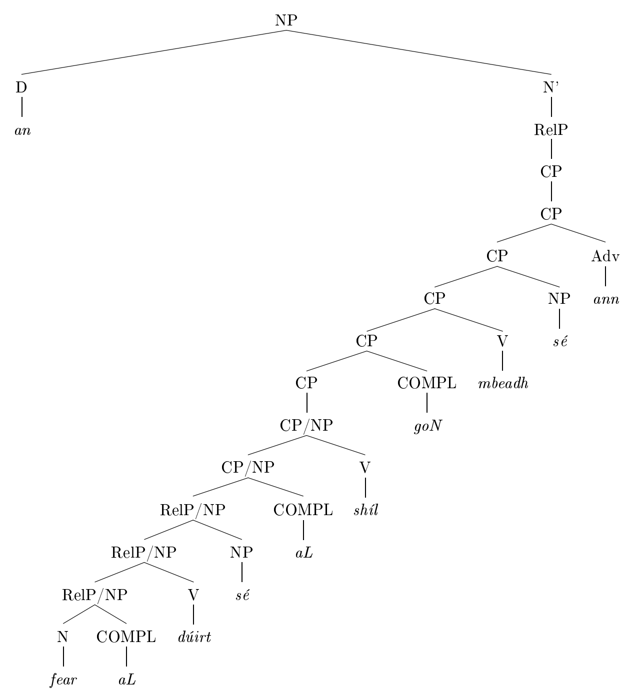

Figure 13: Analysis of (3e)

References

Barry Blake. 1990. *Relational Grammar*. London and New York: Routledge.

Gosse Bouma, Rob Malouf, and Ivan A. Sag. 2001. Satisfying constraints on extraction and adjunction. *Natural Language and Linguistic Theory*, 1(19):1–65.

Joan Bresnan. 2001. *Lexical-Functional Syntax*. Blackwell Publishers.

Jan Terje Faarlund, Svein Lie, and Kjell Ivar Vannebo. 1997. *Norsk Referansegrammatikk*. Oslo: Universitetsforlaget.

Gerald Gazdar, Ewan Klein, Geoffrey Pullum, and Ivan Sag. 1985. *Generalized Phrase Structure Grammar*. Harvard University Press, Cambridge, MA.

Thomas E. Hukari and Robert Levine. 1995. Adjunct extraction. *Journal of Linguistics*, 31(2):195–226.

Robert D. Levine. 2003. Adjunct valents: cumulative scoping adverbial constructions and impossible descriptions. In Jongbok Kim and Stephen Wechsler, editors, *The Proceedings of the 9th International Conference on Head-Driven Phrase Structure Grammar*, pages 209–232, Stanford. CSLI Publications.

James McCloskey. 1979. *Transformational Syntax and Model-Theoretic Semantics*. Dordrecht: Reidel.

Carl J. Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago.

Ivan A. Sag, Thomas Wasow, and Emily M. Bender. 2003. *Syntactic Theory: A Formal Introduction*. CSLI Publications, Stanford, 2 edition.

Ivan A. Sag. 2005. Adverb extraction and coordination: a reply to levine. In Stefan Müller, editor, *The Proceedings of the 12th International Conference on Head-Driven Phrase Structure Grammar, Department of Informatics, University of Lisbon*, pages 322–342, Stanford. CSLI Publications.

Anders Søgaard and Petter Haugereid. 2005. Functionality in grammar design. In Stefan Werner, editor, *Proceedings of the 15th Nordic Conference of Computational Linguistics*, pages 193–202, Joensuu: University of Joensuu. Electronic Publications in Linguistics and Language Technology.

Dominique Sportiche. 1988. A Theory of Floating Quantifiers and Its Corollaries for Constituent Structure. *Linguistic Inquiry*, 19(3).

Mark Steedman. 2000. *The Syntactic Process*. Cambridge, MA and London: The MIT Press.