frame type, there is no need for the feature OPT on syntactic arguments. The PART|SAT value is *plus*, which means that the verb is not a particle verb.

$$\begin{bmatrix} \textit{arg1-12\_np\_le} & \\ \text{VAL} & \begin{bmatrix} \text{C-FRAME} & \textit{arg1-12} \\ \text{C-ARG2|HEAD} & \textit{nominal} \\ \text{PART|SAT} & + \end{bmatrix} \end{bmatrix}$$

Figure 23: The arg1-12_np_le type

The lexical type for unaccusative verbs that can be causativized, like *burn*, and for variable behavior verbs, like *arrive*, is given in Figure 24. The C-FRAME value is specified to be *arg12-2*, which accounts for the alternation between unaccusative and transitive. The HEAD value of C-ARG2 is specified to be *nominal*, which constrains the ARG2 to be an NP.

$$\begin{bmatrix} \textit{arg12-2\_np\_le} & \\ \text{VAL} & \begin{bmatrix} \text{C-FRAME} & \textit{arg12-2} \\ \text{C-ARG2|HEAD} & \textit{nominal} \end{bmatrix} \end{bmatrix}$$

Figure 24: The arg12-2_np_le type

The lexical type for verbs like paint, which can be intransitive, transitive or transitive resultative, is given in Figure 25. The C-FRAME value is specified as *arg1-12-124*, which means that it can enter an unergative frame, a transitive frame, and a transitive frame with a delimiter. The HEAD value of C-ARG2 is specified to be *nominal*, and the HEAD value of C-ARG4 is specified to be *adj*. This ensures that the internal argument is an NP, and that the delimiter is an adjective.

$$\begin{bmatrix} \textit{arg1-12-124\_np\_ap\_le} & \\ \text{VAL} & \begin{bmatrix} \text{C-FRAME} & \textit{arg1-12-124} \\ \text{C-ARG2|HEAD} & \textit{nominal} \\ \text{C-ARG4|HEAD} & \textit{adj} \end{bmatrix} \end{bmatrix}$$

Figure 25: The arg1-12-124_np_ap_le type

Given the means I have described for restricting the syntactic environment of verbs in Norsyg, the C-FRAME values, the HEAD values of the C-ARG2 and C-ARG4 arguments, the KEY value of the C-ARG4 argument, and the PRED value of the particles, one is free to give very specific constraints, only allowing one particular argument frame, or one can let the constraints be less specific, so that the verb can enter more frames.

4.7 Adaptation of Norsk Ordbank

Norsyg uses Norsk Ordbank,[16] which is an open source full-form lexicon for Norwegian with 1,179,549 entries (148,141 different lemmas), to fill out its lexicon. The verbs in Norsk Ordbank are annotated with the argument frame information from the NorKompLeks

[16] http://www.edd.uio.no/prosjekt/ordbanken/

project.[17] A program *convlex* converts the lexicon into a format compatible with the Norsyg grammar (143,263 uninflected lexical entries, of which 8,647 are verbs). It gathers the argument frame information about each verb and creates the corresponding type if this type does not exist already. This is often necessary if a verb can enter many argument frames. The lexical types for verbs have five kinds of information. First, they specify what kind of constructions the verb can enter. If the verb can enter the arg1-construction, the arg12-construction, and the arg124-construction, it is assigned the C-FRAME value arg1-12-124. Second, they specify the HEAD value of the C-ARG2 argument (if applicable). If the C-ARG2 is either an NP or a subordinate clause, the new verb lexical entry type inherits from the type *arg2_cp-np*. Third, the C-ARG3 value is specified to be a reflexive (if applicable). Forth, the new verb lexical types specify the C-ARG4 value (if applicable). If the C-ARG4 value is a PP, the type inherits from the type arg4_pp. Fifth, the new verb lexical entry type specifies whether the verb is a particle verb. If it is a particle verb, it inherits from the type *part-verb*, and if not, it inherits from *non-part-verb*. Other information, like the PRED values of selected particles and prepositions, is specified on each individual lexical entry. Based on the argument frame information specified on verbs in Norsk Ordbank, the lexicon conversion program builds 126 new types for verb lexical entries in addition to the 100 lexical entry types for verbs that already exist. An example of an automatically created verb lexical type is given in (8).

(8)
```
arg12-124-2_part_np_pp_le := arg2_np & arg4_pp &
                             part-verb &
[ SYNSEM.LOCAL.CAT.VAL.C-FRAME arg12-124-2 ].
```

The type in (8) is the type for the verbs *etse* (‘corrode’), *helle* (‘pour’/‘slope’), *hive* (‘throw’), *kippe* (‘flip up’), and *knalle* (‘crack’). What these verbs have in common is that they can enter the arg12-construction, the arg124-construction, and the arg2-construction, hence the C-FRAME value arg12-124-2. The verbs are particle verbs, so the type inherits from *part-verb*. The verbs require an NP as value of C-ARG2 and a PP as value of C-ARG4 (if applicable), so the type inherits from *arg2_np* and *arg4_pp*.

The entry of the infinitival form of *helle* in Norsk Ordbank has the information given in (9), where the fields in angle brackets show what argument frames the verb can enter, <intrans2>, <adv6>, and <part1/ut>.

(9)
```
27112 helle helle verb inf <intrans2> <adv6> <part1/ut>
```

These argument frame specifications are translated into the type in (8) according to a table distributed with the Norsyg grammar (‘nkl2lkb.txt’). When appearing alone, <intrans2> translates into the type *arg2_np_le* (the type for intransitive unaccusative verbs), <adv6> translates into the type *arg124_np_pp_le* (the type for transitive verbs with PP complements), and <part1/ut> translates into the type *arg12_part_np_le* (the type for transitive particle verbs (the PRED value of the particle *ut* (‘out’) is specified on the lexical entry)). When these three argument frames appear on the same lexical entry, the type *arg12-124-2_part_np_pp_le* is created, as shown above. It accommodates all the frames just

[17] NorKompLeks (NKL) is a Norwegian computational lexicon developed at NTNU, Trondheim, Norway. It contains information about inflectional patterns and phonological representations as well as argument structure frames for verbs. There are 105 different codes for argument structure frames in NKL, and each verb is provided with a list of codes showing the possible argument structure frames.

mentioned. The lexical entry of *helle* in the Norsyg grammar is given in (10).

(10)
```
helle-v := arg12-124-2_part_np_pp_le &
[ STEM <"helle">,
  INFLECTION v1,
  SYNSEM.LKEYS.KEYREL.PRED "_helle_v_rel",
  SYNSEM.LKEYS.ALTKEYREL.PRED _ut_p_rel ].
```

5 Comparison of 'packed' vs. expanded lexicon

In order to check the impact of a lexicon with packed argument frame representations as described in the previous section, I used the *convlex* program to generate two versions of the lexicon. In the first version, all verbs were given packed representations, and in the other, each argument frame version of a verb was spelled out as a separate lexical entry.

This means that a verb that has the type *arg1-12_np_le* (see Figure 23) in the packed lexicon, in the expanded version is given two lexical entries, one of the type *arg1_le* and one of the type *arg12_np_le* (one for each of the argument structure codes assigned by the original Norsk Ordbank lexicon). 5,329 of the verbs from the Norsk Ordbank lexicon are listed with only one frame, and are therefore given only one lexical entry in the expanded lexicon, while 3,318 verbs are listed with more than one argument frame and are given the corresponding number of lexical entries. This gave me an expanded lexicon with 12,213 lexical entries for verbs, rather than the 8,647 lexical entries for verbs in the packed lexicon, an increase of 3,566.

The data used for the comparison are taken from a 37 million word Norwegian Wikipedia corpus (2,252,972 sentences). I selected 8,271 sentences containing 5-10 words where all the words were covered by the dictionary of the grammar. This set is referred to as *All items* in the next section. The grammar had a coverage of 5,105 sentences with the packed lexicon and 5,078 sentences with the expanded lexicon. Of the sentences that were parsed with both version of the lexicon, 3,446 sentences were given the same number of analyses with both versions. This set is referred to as *Equal coverage* in the next section. I also created a third set of sentences where I excluded the sentences containing the copula verb 'å være' 'to be' from the Equal coverage set, since they are likely to be overrepresented in the data. (There are very many Wikipedia sentence like *Lesotho er et land i Afrika* 'Leshoto is a country in Africa'.) This set is referred to as *No copula* in the next section and has 1,902 items.

6 Results

I let the grammar loaded with the two different lexicons (packed and expanded) parse the three sets of sentences described in the previous section (All items, Equal coverage, and No copula) and compared the results. Table 2 shows that the two versions of the grammar, as already noted, have similar coverage on the All items set (61.4% and 61.7%), and, as expected, the same coverage (100%) on Equal coverage, and No copula. The two versions produce slightly more analyses on average with the expanded lexicon (23.69) than the packed lexicon (20.58) for All items.[18] For the Equal coverage, and No copula sets, the two

[18] The items in All items are assigned more analyses on average by the grammars. It is hard to constrain the grammars to perform equally on more ambiguous, and this is probably the reason behind the diverting numbers.

versions (as expected) produce the same number of analyses (14.39 and 13.66, respectively). More importantly, the table also illustrates the difference in lexical ambiguity of the two grammars. The difference in lexical ambiguity of expanded vs. packed is 4.20 vs. 3.41 on All items, 3.73 vs. 3.20 on Equal coverage, and 3.81 vs. 3.22 on No copula. This means that on average more lexical items are entered into the parse chart with the expanded lexicon than with the packed lexicon.

		Expanded lexicon			**'Packed' lexicon**		
Data	items	lexical	analyses	coverage	lexical	analyses	coverage
All items	8,271	4.2	23.69	61.4%	3.41	20.58	61.7%
Equal coverage	3,443	3.73	14.39	100%	3.2	14.39	100%
No copula	1,902	3.81	13.66	100%	3.22	13.66	100%

Table 2: Comparison of competence

Table 3 shows how the use of packed argument frames affects the performance of the parser, compared to the use of the expanded lexicon. When applied to the Equal coverage set, the number of tasks[19] is reduced by 10.7%, and the use of space is reduced with 13.3%. Similarly, the number of tasks is reduced by 12.8%, and the use of space is reduced with 13.2%, when applied to No copula. The numbers showing reductions for All items are less reliable, since the set includes sentences where the two versions of the grammar produce different numbers of analyses.

		Expanded lexicon		'Packed' lexicon		Reduction	
Data	items	tasks	space	tasks	space	tasks	space
All items	8,271	3,265	208,324	2,480	139,430	24.0%	33.1%
Equal coverage	3,443	2,022	102,964	1,805	69,540	10.7%	13.3%
No copula	1,902	1,341	59,080	1,170	51,294	12.8%	13.2%

Table 3: Comparison of performance

One possible objection to this test would be that a grammar without the packing of argument structure information could be implemented in a different way, which would make parsing more efficient. However, this comparison is only done for testing the impact of the packing of argument structure information in a grammar that is implemented similar to Norsyg.

7 Conclusion

In this paper I have presented a constructionalist grammar design which allows for a packed representation of argument frame information. The design is implemented in Norsyg, which is an open source grammar of Norwegian. I have compared two versions of the Norsyg lexicon. One, where valence alternations are accounted for by means of multiple lexical entries, and one, where valence alternations are accounted for by means of packed type constraints. It has been demonstrated that the packed version of the lexicon introduces fewer lexical items in the parse chart. The use of the packed version of the lexicon also leads to a reduction of tasks performed and space used by the parser.

[19] Tasks stands for the average number of parser tasks executed, i.e. calls to the unifier.

Acknowledgements

I would like to thank the audience at PACLIC 11, Singapore, for their valuable feedback and for the comments of three anonymous reviewers. I also would like to thank Francis Bond and Mathieu Morey for useful comments and suggestions.

8. References

Bender, E. M., Flickinger, D., and Oepen, S. 2002. The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In J. Carroll, N. Oostdijk, and R. Sutcliffe, editors, *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14, Taipei, Taiwan.

Borer, H. 2005a. *Structuring Sense. An Exo-Skeletal Triology. Volume I. In Name Only*. Oxford University Press.

Borer, H. 2005b. *Structuring Sense. An Exo-Skeletal Triology. Volume II. The Normal Course of Events*. Oxford University Press.

Bresnan, J. 2001. *Lexical-Functional Syntax*. Blackwell Publishers.

Chomsky, N. 1981. *Lectures on Government and Binding. The Pisa Lectures.* Dordrecht, Holland and Cinnaminson, USA: Foris Publications.

Copestake, A. 2002. *Implementing Typed Feature Structure Grammars*. CSLI publications.

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

Gazdar, G. 1981. Unbounded dependencies and coordinate structure. *Linguistic Inquiry*, **12**, 155–184.

Haugereid, P. 2009. *Phrasal subconstructions: A constructionalist grammar design, exemplified with Norwegian and English*. Ph.D. thesis, Norwegian University of Science and Technology.

Haugereid, P. 2012. The adverb argument intersection field in a left-branching grammar of Norwegian. To be presented at the HPSG-2012 Conference, Daejeon, South Korea.

Haugereid, P. and Morey M. 2012. A left-branching grammar design for incremental parsing. To be presented at the HPSG-2012 Conference, Daejeon, South Korea.

Oepen, S. and Carroll, J. 2000. Ambiguity packing in constraint-based parsing. Practical results. In *Proceedings of the 1st Conference of the North American Chapter of the ACL*, pages 162 – 169, Seattle, WA.

Oepen, S. and Flickinger, D. 1998. Towards systematic grammar profiling. test suite technology ten years after. *Journal of Computer Speech and Language: Special Issue on Evaluation*, **12 (4)**, 441–437.

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

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

Siegel, M. and Bender, E.M. 2002. Efficient deep processing of Japanese. In *COLING:02*, pages 1–8, Taipei, Taiwan.

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

Åfarli, T. A. 2007. Do verbs have argument structure? In E. Reuland, T. Bhattacharya, and G. Spathas, editors, *Argument Structure*, pages 1–16. Amsterdam: John Benjamins.