Increasing grammar coverage through fine-grained lexical distinctions

Petter Haugereid

Abstract. In this paper, I show how the development of Norsyg, an HPSG-inspired constructionalist grammar of Norwegian, benefits from the highly specific and precise implementation of NorGram, an LFG grammar for Norwegian. I focus on one aspect, NorGram's fine-grained lexical categories. The aim of the paper is twofold: (i) to give a glimpse of the process of developing a computational grammar, and (ii) to illustrate how a constructionalist grammar benefits from the insights of NorGram, even though the grammatical models differ significantly.

1 Introduction

There are different approaches to the automatic analysis of sentences. A common approach is to train a shallow statistical parser based on large amounts of syntactically annotated texts (a treebank). The advantage of these systems is that they utilize already existing resources (annotated texts), they have large coverage, and they are very fast. The main problems are that once they have reached a certain level, it is hard to make improvements, and there is never a guarantee that the analysis provided is the correct one, or even a possible one.

A different approach to automatic analysis of sentences is to develop deep, rule-based computational grammars. These systems take much more time to develop, and in the beginning, the coverage is very limited. If there is a missing lexical item or a missing rule for a certain linguistic construction, the grammar does not provide a parse. Scaling up these systems may take several years. However, given the fact that the systems are rule based, the grammar developer is in control of what analyses are possible, and corrections that address particular linguistic phenomena may be made. So if the aim of the system is high precision, building a deep grammar is a better option than building a shallow parser in the long term.

One possible reason for the relatively high interest in linguistically founded computational grammars in Norway is that treebanks required for building statistical parsers

The very model of a modern linguist. Edited by Victoria Rosén and Koenraad De Smedt. BeLLS Vol. 8 (2017), DOI 10.15845/bells.v8i1.1334.

have not been available for Norwegian until recently.[1] Another reason is the interest in grammar formalisms like LFG and HPSG, which are both associated with environments for grammar implementations. In this paper I will describe two quite different computational grammars, NorGram and Norsyg, and show how insights in NorGram can be used to develop the coverage of Norsyg.

2 NorGram and Norsyg

NorGram (Dyvik 2000) is the result of a long-term, incremental effort to develop a theoretically motivated, large coverage grammar for Norwegian.[2] It is written within the framework of Lexical Functional Grammar (LFG) (Bresnan 2001), under the ParGram umbrella (Parallel Grammar Project), which is an association of groups working on computational LFG grammars for various languages (Butt et al. 2002). LFG is a lexicalist framework where linguistic objects are represented with mainly two structures: c-structure (constituent structure) and f-structure (functional structure). The c-structure shows the hierarchical organization of constituents in a clause at the same time as it shows how the parser has worked, combining constituents by means of phrase structure rules. The f-structure represents linguistic information about each constituent and shows the functional relationship between the constituents. In this paper we will mainly consider c-structures. The tree in Figure 1 shows the c-structure of the main clause in example (1) analyzed with NorGram.[3]

(1)	*Dessverre*	*ville*	*ikke*	*disse*	*studentene*	*lære*	*syntaks.*
	unfortunately	wanted	not	these	the students	learn	syntax

‘Unfortunately, these students didn’t want to study syntax.’

Norsyg is a typed feature structure grammar, and is implemented with the LKB system (Copestake 2002) as a part of the DELPH-IN effort (http://www.delph-in.net/). It is based on the Grammar Matrix (Bender et al. 2002), which is a starter kit for HPSG grammar development. Norsyg has kept most of the HPSG feature geometry, but the intuition behind the analyses is radically different from regular lexicalist HPSG grammars. It is a constructionalist grammar, and the backbone of the grammar consists of about 15,000 constructions. Argument-frame constructions constitute the main part of

[1] NorGramBank (Dyvik et al. 2016), a large-scale syntactically annotated treebank of Norwegian, has recently become available. NorGramBank has been developed through the projects TrePil (Rosén, De Smedt, Dyvik, et al. 2005) and INESS (Rosén, De Smedt, Meurer, et al. 2012). This treebank is based on parses obtained with NorGram, which will be further discussed in the present paper.

[2] Helge Dyvik has been a pioneer of computational grammar and parsing in Norway. He started already in the 1980’s with the D-PATR formalism, a development environment for unification-based grammars. This work was carried over in the PONS project (Dyvik 1989), a machine translation project with semantic transfer. Since 1999, he has been the main developer of NorGram, which was used as a parsing grammar not only for NorGramBank but in the translation projects LOGON (Oepen et al. 2007) and HandOn.

[3] The tree is slightly simplified for expository reasons.

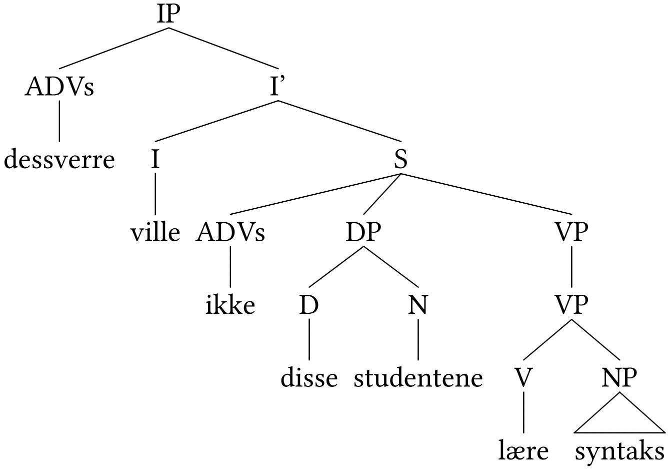

Figure 1: LFG c-structure of a Norwegian main clause with NorGram, cf. example (1)

these; each argument frame of each verb is assumed to be a construction. For example, the transitive and ditransitive frames of the verb *lære* 'learn' are assumed to be unique constructions.

The high number of constructions, in addition to well known phenomena such as flexibility with regard to positioning of adjuncts, the active–passive voice alternation, and different kinds of clause structures, make the assumption of flat phrase structure rules impossible. Instead, the grammar is given a "fragmented" design, where constructions are assumed to be built up of *subconstructions.* A subconstruction may be a binary phrase structure rule with a word as its second daughter (see the rules in Figure 2), or it may be a lexical item, like a verb, a function word or an idiomatic word. Each subconstruction contributes a simple type, which by itself may carry very little meaning. During parsing, however, the types provided by the subconstructions are unified, and if the parse succeeds, the unification of the subconstruction types yields one of the 15,000 construction types licensed by the grammar.[4] This subconstructional design gives the grammar the flexibility needed to accommodate a wide variety of syntactic phenomena while limiting the number of phrase structure rules to 110. The tree in Figure 2 shows the parse tree of the main clause in (1) analyzed with Norsyg.

The trees in Figure 1 and Figure 2 illustrate a principal difference between the two grammars. NorGram on the one hand is based on standard X-bar theory. It consists of phrase structure rules such as VP → V NP. The grammar relies heavily on the formula-

[4] Needless to say, the type hierarchy where the possible combinations of subconstruction types are defined, is rather big, but once it is compiled, its size does not affect the efficiency of the parser very much. A small test indicates an increase in parsing time of about 15% when the lexicon is scaled up.

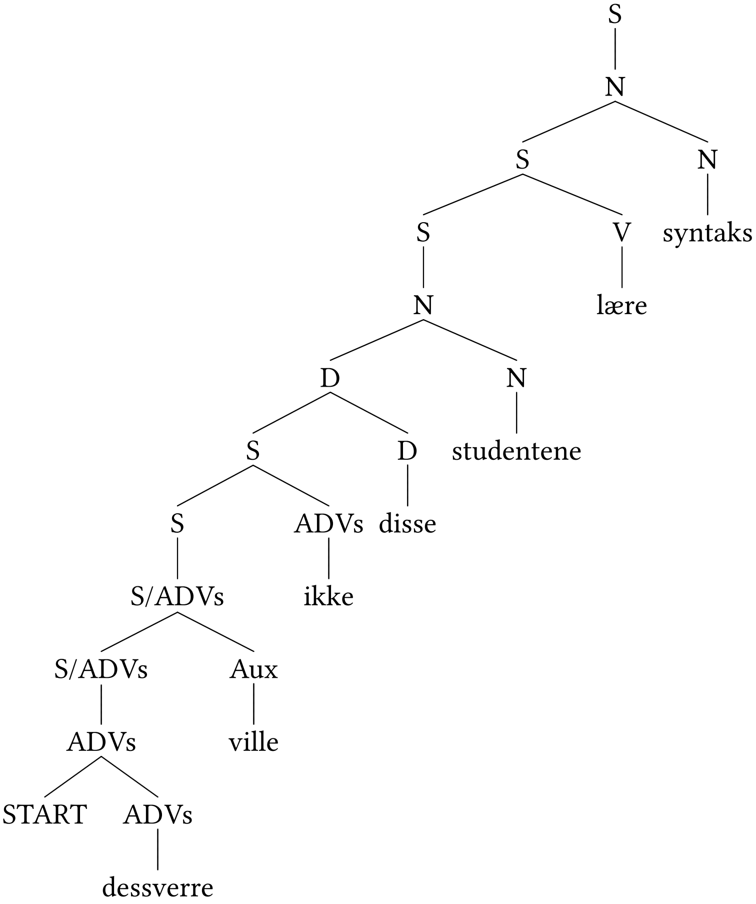

Figure 2: Parse tree of a Norwegian main clause with Norsyg, cf. example (1)

tion of this kind of phrase structure rules. The structures are relatively flat, a rule may have more than five daughters, and phrasal constituents may appear in a non-initial and non-final position, like the DP *disse studentene* in Figure 1.

Norsyg on the other hand consists of only binary and unary phrase structure rules where the second daughter (if there is one) is a word. As the words combine, a feature structure is built where linguistic information of the clause is represented. From the resulting feature structure, the constituent structure in Figure 3 is derived.[5] The parse also results in a semantic representation, an MRS (Copestake et al. 2005).

[5] By separating the parse tree (see Figure 2) from the constituent tree (see Figure 3), Norsyg allows for flat constituent structures at the same time as the phrase structure rules are unary or binary. The motivation behind this separation is explained in Haugereid and Morey (2012).

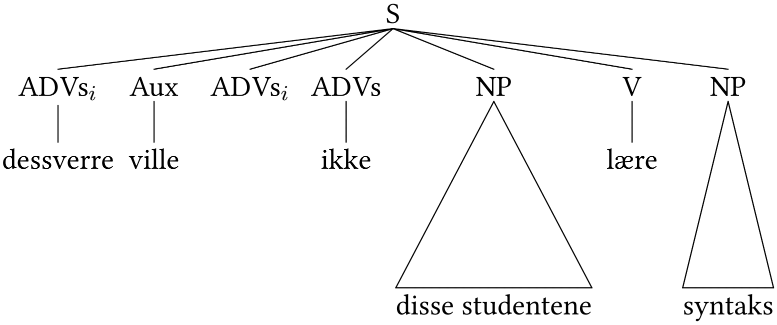

Figure 3: Constituent structure of a Norwegian main clause by Norsyg

Allthough Norsyg has been developed over many years, there are several phenomena that are covered by NorGram that are yet to be implemented in Norsyg. In the next section I will show how I use NorGramBank, the treebank syntactically annotated with the help of NorGram, to identify phenomena that will have an impact on the coverage of Norsyg.

3 Identifying phenomena not covered by Norsyg

In grammar development, identifying phenomena that are not covered by the grammar, is usually very easy. One can just take a random sentence that the grammar does not parse, and use the diagnostic tools of the parser to find out what goes wrong. This is from my own experience the most common way to improve coverage of a grammar, and is the approach one would take during treebanking. If a sentence does not parse, one attempts to make it parse. Often adding a lexical item is enough, but one may also encounter very challenging phenomena which require a complete overhaul of the grammar. However, the impact of the changes made, whether they are large or small, may be just a very small increase in coverage.

In this section, I describe a more systematic way of identifying phenomena that are not covered by Norsyg, and which will have an impact on the coverage of the grammar. And in this, I will utilize the linguistic insights behind NorGram and the 60 million word corpus NorGramBank which has been syntactically annotated with the help of NorGram.

Syntactic rules in LFG have information about the c-structure of its constituents as well as the f-structure. In the parsing process, first the c-structure backbone is constructed (as a packed parse forest), and then, in the next step, the (f-structure) equations attached to the c-structure rules are solved. In order to reduce the size of the c-structure parse forest, NorGram has been equipped with a relatively large set of discriminative preterminals or lexical categories. This results in fewer equations that have to be solved, and makes the parser more efficient.

In order to see which preterminals are used by the grammar, I downloaded a frequency list of preterminals from NorGramBank. It turns out that the treebank has 220 different preterminals. A few of them have a grammar internal function (different kinds of tags for enhancing processing), and others represent punctuation marks. However, most of the preterminals have a solid linguistic foundation. The most frequent lexical categories are given in Table 1.[6] We can see that the most frequent categories are nouns (12.82%) and pronouns (8.83%). The table also shows that the lexical categories are fine-grained. There are for example separate categories for finite main verbs (Vfin), finite auxiliaries (Vauxfin), and finite copula verbs (Vcopfin).

Nr.	Category	Freq.
1	N	12.82
2	PRON	8.83
3	Vfin	7.28
5	P	6.53
6	A	5.84
8	Vauxfin	3.14
9	Vinf	2.95
10	Vcopfin	2.16

Table 1: The eight most frequent lexical categories in NorGramBank, with frequencies in percentages

Further down the list, there are some lexical categories that represent phenomena that are not covered or treated in a systematic way by Norsyg. The categories are well documented in the NorGram online documentation, and I use this documentation and my knowledge of Norsyg to identify the missing categories. The most frequent are shown in Table 2: finite inquit verbs (Vinqfin), prepositions that take subordinate clauses or infinitival clauses as complements (Pvbobj), interjections (INTERJ), correlative coordinators (CONJcorr), and titles (TTL).

In the following, I will show how I have utilized the information about the categories that are unaccounted for in the development of Norsyg. Given the constructionalist design of Norsyg, some of the phenomena will not be analyzed in terms of separate lexical categories, like inquit verbs and prepositions that take subordinate clauses or infinitival clauses as complements. Rather, the phenomena will be accomodated by constructions. Prepositions that take subordinate clauses or infinitival clauses as complements, for example, will still have the lexical category preposition, but they will

[6] The fourth and seventh most frequent preterminals are PERIOD and COMMA. They are left out of the table.

Nr.	Category	Freq.
41	Vinqfin	0.40
42	Pvbobj	0.29
67	INTERJ	0.08
68	CONJcorr	0.07
72	TTL	0.06

Table 2: The five most frequent NorGram lexical categories unaccounted for in Norsyg

be made compatible with constructions that involve a subordinate clause or infinitival clause complement.

4 Developing Norsyg on the basis of NorGram lexical categories

As mentioned, NorGramBank is a corpus of approximately 60 million words. The larger part of the corpus has been stochastically disambiguated, and approximately 315,000 words of parsed text are manually disambiguated. From the corpus of manually disambiguated sentences, I have selected 14,770 sentences marked as "gold" by the annotators (127,644 words). These are sentences that are parsed by NorGram and that have been disambiguated by an annotator and marked as correct.

In my work on adding missing analyses to Norsyg I started with the lexical categories on top of the list in Table 2, inquit verbs, and worked my way down. For each phenomenon I added to the grammar, I did a test run on the gold corpus to check the impact of the changes made. Before the development started, I checked Norsyg's coverage of the gold corpus. It parsed 7216 of the 14770 sentences (48.86%).

4.1 Inquit verbs (Vinq)

Inquit verbs are a group of verbs that typically indicate direct speech, as shown in (2), but this group also includes verbs with the same syntactic behavior, like *tro* 'believe' and *synes* 'think'.

(2) *Jeg sov, sa han.*
I slept said he
'I slept, he said.'

This is a phenomenon that had not been implemented in Norsyg, and in order to account for the construction, three rules were introduced. One rule is created for sentences where the inquit complement is a full sentence, as in (2). There is also a rule where the complement is some other constituent, such as an NP or PP, or an interjection. The third rule marks the position where the complement is extracted from. In addition, the 107 verbs marked as regular inquit verbs in Norgram were constrained in such a way that they were allowed as verbs in inquit constructions in Norsyg.

After adding the new rules and lexical constraints to the grammar, 100 sentences that earlier did not get a parse, now were parsed by Norsyg, an increase of 0.67%. Some examples are given in (3)–(5).

(3) *Nå er jeg trygg, sier hun og smiler mot sykepleieren.*
now am I safe says she and smiles towards the nurse
'Now I am safe, she says and smiles towards the nurse.'

(4) *De liker meg ikke, skriver hun.*
you like me not writes she
'You don't like me, she writes.'

(5) *En syk øvelse, tenker hun sint.*
a sick exercise thinks she angry
'A sick exercise, she thinks angrily.'

An abbreviated Norsyg analysis of (4) is provided in Figure 4. It shows the application of two of the added rules. (The mother nodes of the rules are framed.) The top rule in the tree is the rule that marks the position the complement is extracted from. It is a unary rule that takes as input a structure which has a sentence on the SLASH list.[7] The framed S/S rule further down the tree is a rule that takes as input a main clause and a comma, and enters selected features of the main clause onto the SLASH list.

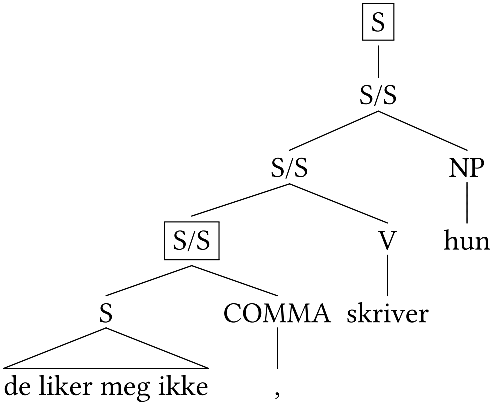

Figure 4: Norsyg analysis of sentence with inquit verb and main clause complement

[7] In HPSG, long distance dependencies are handled by a feature SLASH. Contrary to regular HPSG grammars, the extraction site in Norsyg dominates the topic, rather than the other way round. This is enforced by the incremental bottom-up parsing process.

4.2 Pvbobj: prepositions that take subordinate clauses or infinitival clauses as complements

In the second batch, I added analyses for Pvbobj, the category for prepositions that take subordinate clauses or infinitival clauses as complements to form PPs that function as adverbials. There are 23 such prepositions, among them *for* 'for', *i tillegg til* 'in addition to' and *uten* 'without'. The inclusion of a new analysis for these prepositions in Norsyg involved changing the constraints of these prepositions so that they were allowed in constructions where the head is a preposition and the complement is a subordinate clause or an infinitival clause.

After the analyses were added, the grammar produced analyses for 49 more sentences, among them, the sentence in (6).

(6) *Amygdala starter analyser for å se mulige farer.*
Amygdala initiates analyses for to see possible dangers
'Amygdala initiates analyses in order to see possible dangers.'

An abbreviated analysis of (6) is presented in Figure 5. The preposition with the new constraints is framed.

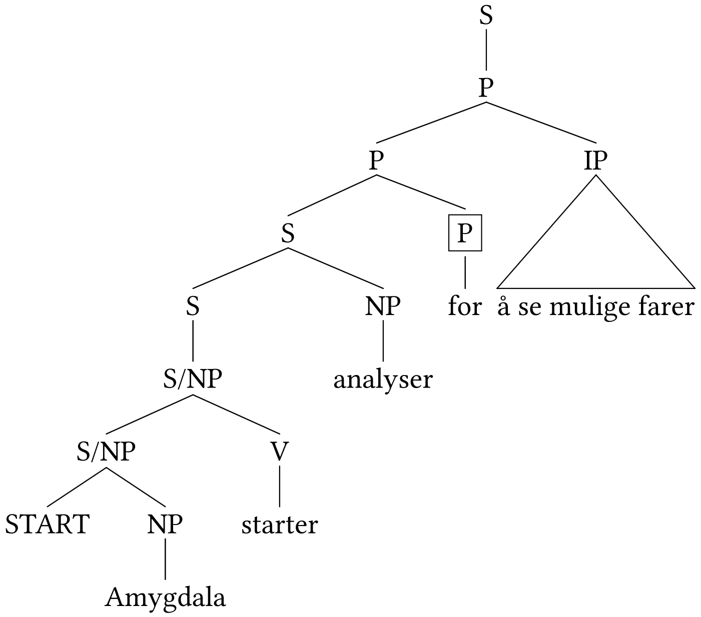

Figure 5: Norsyg analysis of sentence with a PP adjunct that has an infinitival clause complement

4.3 INTERJ: Interjections

In the third batch, I added analyses for interjections. In Norsyg, they are given the status of roots, which means that they are allowed to form sentences on their own, function as arguments of inquit verbs, or be coordinated with other roots (including sentences).

After the analyses were added, the grammar produced analyses for 31 more sentences, among them, the sentences in (7) and (8). The sentence in (8) also benefits from the recently added inquit analysis.

(7) – *Å, har du ikke hørt det?*
oh have you not heard it
'– Oh, haven't you heard?'

(8) – *Jada, mamma, fleipet han.*
of course mum he joked
'– Of course, mum, he joked.'

An abbreviated analysis of (7) is provided in Figure 6. It shows the new category of interjections (INTERJ) and the new rule for adding interjections (S). The rule that adds interjections, takes a START symbol as its first daughter and an interjection as its second daughter and forms a structure with root status. It is then coordinated with the following yes–no question.[8]

4.4 CONJcorr: correlative coordinators

In the fourth batch, the words *både* 'both', *verken* 'neither', and *såvel* 'both' were given an analysis. These are words that initiate a coordination and select the coordinator between the conjuncts. There had been an analysis for these words at an earlier stage, but it had become obsolete. After recreating the analysis, 11 more sentences got a parse, among them (9) and (10).

(9) *Den knuste både negl og bein.*
it broke both nail and bone
'It broke both nail and bone.'

(10) *Både han og jeg har fått tørre føtter.*
both he and I have got dry feet
'Both he and I have got dry feet.'

[8] Given the left-branching design of the grammar, the parse tree of the coordinated structures looks rather counter-intuitive, but it is chosen in order to maintain the overall incremental design. It also makes possible a novel account of gapping constructions (Haugereid 2017).

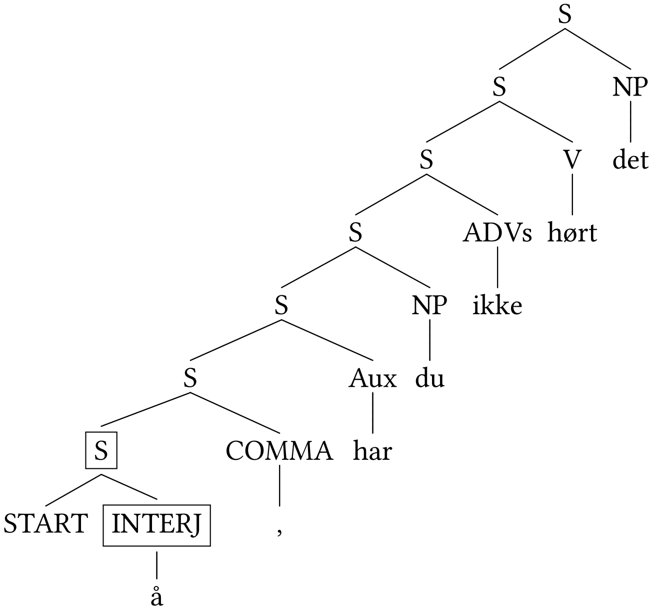

Figure 6: Norsyg analysis of sentence with interjection coordinated with yes–no clause

4.5 TTL: title

In the fifth batch, 5633 titles from Norgram were added, *dr.* 'dr.', *språkprofessor* 'linguistics professor', *fagekspert* 'professional', *norrønfilolog* 'Old Norse philologist', *hedersmann* 'man of honour', *ektemann* 'husband', *bestefar* 'grandfather', *onkel* 'uncle', and *pappa* 'daddy', among others.[9] They are analyzed as pre-modifiers of proper nouns in Norsyg. After adding the analysis of the titles, 24 new sentences were parsed by the grammar, among them the sentences in (11) and (12).

(11) *Mormor snudde seg og så mot onkel Ernst.*
Grandma turned and looked so towards uncle Ernst
'Grandma turned and looked in the direction of uncle Ernst.'

(12) – *Jomfru Bendeke, mener du?*
virgin Bendeke, mean you
'– Miss Bendeke, you mean?'

[9] As it happens, all these titles can be attributed to Helge Dyvik.

5 Results

The effect of the grammar development work is summarized in Table 5. It shows an increase of coverage on the gold corpus of 210 sentences, or 1.41%. It also shows that the categories at the top of the list resulted in the most significant gains of coverage.

phenomenon	coverage	%
Before	7216	48.86
Inquit verbs	7316	49.53
Pvbobj	7365	49.86
INTERJ	7396	50.07
CONJcorr	7407	50.14
TTL (title)	7426	50.27

The effect is also illustrated in the chart in Figure 5. It shows that the number of new sentences that receive an analysis by the grammar increases as the frequency of the added lexical category goes up. When the frequency is 0.06 % (titles), the number of added sentences is 24, and when the frequency is 0.4 % (inquit verbs), the number of new analyses is 100. This is of course an expected result, and it confirms the obvious, namely that there is more to gain from adding analyses for more frequent lexical categories.

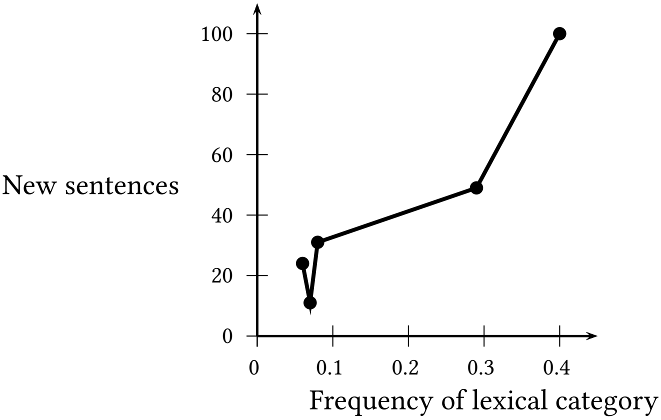

Figure 7: Number of new sentences with regard to frequency of added lexical category

A manual inspection of the sentences that earlier did not get a parse, but which after the changes to the grammar got a parse, shows that about two out of three sentences can be attributed directly to the added analysis. The last third is mainly made up of longer sentences for which the chart size after the changes to the grammar no longer exceeds 20 megabytes.[10] There is also a set of sentences which after the changes to the grammar exceeds the 20 megabyte limit, and these sentences therefore do not get an analysis. The measures are therefore not completely precise, but still they give a clear indication of the impact of the added analyses.

It should also be mentioned that a number of sentences which earlier were given an analysis for the wrong reason, got the correct analysis after the changes to the grammar. These changes are however difficult to measure.

6 Conclusion

Using the lexical categories of NorGram in the development of Norsyg has proved to be a fruitful exercise. As a grammar writer, I can foresee what grammatical phenomena the grammar I am developing can account for. However, it is harder to see exactly which changes will amount to the greatest gain of coverage. Here, the well-documented lexical categories of NorGram have been very useful. By looking at their frequencies in the NorGramBank corpus, and consulting the detailed online documentation and the analyses provided by NorGram, I have been able to pick five categories that represent phenomena not covered by the grammar and increase its coverage by 1.41%.

Acknowledgments

I would like to thank the research group Språk og samfunn at Western Norway University of Applied Sciences, two anonymous reviewers, and Victoria Rosén for very useful comments and suggestions for this paper. I would also like to thank Helge Dyvik for being a great inspiration and for his contribution to the field of computational linguistics.

References

Bender, Emily M., Dan Flickinger, and Stephan Oepen (2002). “The Grammar Matrix: An Open-Source Starter-Kit for the Rapid Development of Cross-Linguistically Consistent Broad-Coverage Precision Grammars”. In: *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*. Ed. by John Carroll, Nelleke Oostdijk, and Richard Sutcliffe. Taipei, Taiwan, pp. 8–14.

Bresnan, Joan (2001). *Lexical-Functional Syntax*. Malden, MA: Blackwell.

Butt, Miriam, Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi, and Christian Rohrer (2002). “The Parallel Grammar Project”. In: *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Com-*

[10] The maximum chart size is set to 20 megabytes during batch parsing of the 14770 ‘gold’ sentences in order to limit the batch parsing time to about 30 minutes. By setting the chart size to 200, the number of sentences parsed is increased by 479, but this also increases the batch parsing time to almost three hours.

putational Linguistics (COLING), Taipei, Taiwan. Ed. by John Carroll, Nelleke Oostdijk, and Richard Sutcliffe. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 1–7.

Copestake, Ann (2002). *Implementing Typed Feature Structure Grammars.* Stanford, CA: CSLI publications.

Copestake, Ann, Dan Flickinger, Carl J. Pollard, and Ivan A. Sag (2005). "Minimal Recursion Semantics: an Introduction". In: *Research on Language and Computation* 3.4, pp. 281–332.

Dyvik, Helge (1989). *The PONS Project.* Tech. rep. Department of Linguistics, University of Bergen.

– (2000). "Nødvendige noder i norsk: Grunntrekk i en leksikalsk-funksjonell beskrivelse av norsk syntaks [Necessary nodes in Norwegian: Basic properties of a lexical-functional description of Norwegian syntax]". In: *Menneske, språk og felleskap.* Ed. by Øivin Andersen, Kjersti Fløttum, and Torodd Kinn. Oslo: Novus forlag, pp. 25–45.

Dyvik, Helge, Paul Meurer, Victoria Rosén, Koenraad De Smedt, Petter Haugereid, Gyri Smørdal Losnegaard, Gunn Inger Lyse, and Martha Thunes (2016). "NorGramBank: A 'Deep' Treebank for Norwegian". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16).* Ed. by Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk, and Stelios Piperidis. ELRA. Portorož, Slovenia, pp. 3555–3562.

Haugereid, Petter (2017). "An incremental approach to gapping and conjunction reduction". In: *Proceedings of the 24th International Conference on Head-Driven Phrase Structure Grammar, University of Kentucky, Lexington.* Ed. by Stefan Müller. Stanford, CA: CSLI Publications, pp. 179–198.

Haugereid, Petter and Mathieu Morey (2012). "A left-branching grammar design for incremental parsing". In: *Proceedings of the 19th International Conference on Head-driven Phrase Structure Grammar, Chungnam National University Daejeon.* Ed. by Stefan Müller. Stanford, CA: CSLI Publications, pp. 181–194.

Oepen, Stephan, Erik Velldal, Jan Tore Lønning, Paul Meurer, Victoria Rosén, and Dan Flickinger (2007). "Towards hybrid quality-oriented machine translation. On linguistics and probabilities in MT". In: *In Proceedings of the 11th Conference on Theoretical and Methodological Issues in Machine Translation.*

Rosén, Victoria, Koenraad De Smedt, Helge Dyvik, and Paul Meurer (2005). "TREPIL: Developing methods and tools for multilevel treebank construction". In: *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005).* Ed. by Montserrat Civit, Sandra Kübler, and Ma. Antònia Martí, pp. 161–172.

Rosén, Victoria, Koenraad De Smedt, Paul Meurer, and Helge Dyvik (2012). "An Open Infrastructure for Advanced Treebanking". In: *META-RESEARCH Workshop on Ad-*

vanced Treebanking at LREC2012. Ed. by Jan Hajič, Koenraad De Smedt, Marko Tadić, and António Branco. Istanbul, Turkey, pp. 22–29.