IBM's Norwegian Grammar Project, 1988–1991

Jan Engh

Oslo University Library, Norway
jan.engh@ub.uio.no

Abstract. During the years 1988–1991, IBM Norway developed a broad-coverage grammar for Norwegian Bokmål as part of an international corporate effort to create writing tools for all platforms and for all major language communities where IBM had business at that time. The grammar was based on IBM's own lexicon and morphology modules and a key factor of the technology was the programming language PLNLP. The main project halted in 1990 because of the world's economic crisis. However, local development continued with a view to a different application: Machine translation between Norwegian Bokmål and Nynorsk. Unfortunately, even this project did not reach a natural conclusion for economic reasons. In addition to producing linguistic results, the project showed how difficult it is to rely on one unique source of corporate funding for a comprehensive long-term project. It also showed how a national subsidiary of an international corporation could not count on local public support.

Keywords: Broad coverage grammar, syntax, natural languages, Norwegian, machine translation, public funding

1 Introduction

During the period 1988–91, IBM Norway developed its own grammar for Norwegian Bokmål as a natural follow-up project after the company's lexicographical projects of the mid-1980s [1]. The project was funded by IBM's Advanced Systems' Development (ASD)[1] as part of a plan to create writing tools for all IBM platforms – for all major language communities of Western Europe, the Middle East, and East Asia.[2]

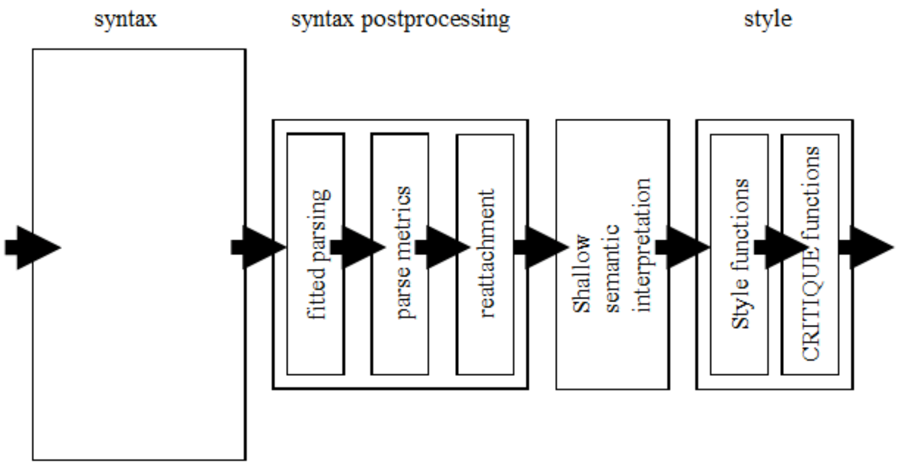
Initially, two main components were implied: A broad-coverage analysis grammar including a component for post syntax processing and a style module. Various other applications were planned. Since an unannounced product was involved, the project was confidential. Later, when international funding failed, IBM Norway decided to carry on with the project as an open research program at a reduced pace and with a different application as its primary objective [2, 3].

The basic idea behind the project was to link modules of syntax, semantics, and discourse analysis to form a unified whole on a firm morphological basis, using a

[1] Based in Bethesda, Maryland.

[2] With the notable exception of Finnish and Icelandic.

unified formalism called "Programming Language for Natural Language Processing" or PLNLP.



Unfortunately, the project was cancelled because of a lack of funding. At the time of its termination, though, the principal component of the grammar, the analytic syntax, had reached a comparatively high functioning level. However, only partial experimental implementations of the remaining components had been made.

In this article, I shall focus on the part of the grammar that was actually implemented, the syntax, and the national continuation project, a system for machine translation between Norwegian Bokmål and Nynorsk.[3] I shall only give a brief sketch of the general technology, which has been thoroughly represented elsewhere [5–9]. The implementation, including aspects of the linguistic development, will be given and I will account for the general experience gained from the project.

2 The Project

2.1 Goals and Means

The main goal of the project was to create a grammar with the capacity of processing every possible input string of Norwegian words: To analyse fully any sequence generally recognised as correct, and to analyse as much as possible any other sequence. This was motivated, of course, by the simple fact that a grammar without a broad coverage is useless for commercial purposes. Developing such a grammar was quite a challenge, however, since one had to face the often unexpected variety and complexity of natural language. On the other hand, with a full-scale design, one

[3] For those unfamiliar with the language situation of Norway, see [4], pp. 53–57 and pp. 98–104.

would not run the risk of ending up in an "it doesn't scale up" situation. In the 1980s, most grammars developed in a scientific environment only worked on a very small subset of a given language.[4]

The task of the Norwegian development group was twofold. It consisted in the adaptation of the general technology provided by ASD, so that it could handle the peculiarities of Norwegian and, above all, it should implement the grammar rules based on thorough linguistic insight.

2.2 Technology

The key factor of the grammar technology was "Programming Language for Natural Language Processing" (PLNLP), a rule-based language specially designed for the purpose. All software was written in PLNLP and an entire development environment was available for grammar writing, including tools for debugging and regression testing. The PLNLP environment provided a shell into which the user loaded a grammar definition as a set of PLNLP rules. Subsequently, the program could decode a language sequence according to the rules of the grammar. Details of the decoding process were displayed by selecting from a variety of tracing options.

2.3 Linguistics

From a linguistics point of view, a grammar written in PLNLP was an augmented phrase structure grammar: It consisted of phrase structure rewriting rules "augmented" by specifications and conditions for the application of the rules. Independently of the linguistic meaning that the grammarian would like to assign to them, basic descriptive entities such as 'head', 'premodifier', and 'postmodifier' already had a rather concrete meaning in the PLNLP system. Apart from this, PLNLP could be considered neutral to linguistic theory. In principle, any theoretical approach or linguistic model could be implemented by means of PLNLP. Neither was PLNLP characterized by any latent bias as far as the structure of the language under analysis was concerned. In fact, PLNLP proved itself a valuable tool for the representation of languages as different as English, Arabic, and Japanese.

One important characteristic of actual use of the formalism in the grammar was the "relaxed" approach:[5] to parse as extensively as possible every sequence found in actual texts submitted for analysis and in an "extrinsic" way. No claim was made of creating any "intrinsic" representation of "how the language really is." The grammar was constructed based on actual language use and the surface text provided the clues for parsing. Thus, in order to identify any sentence level constituents, the parsing rules had to contain information about what (form of a) word could be followed by (what form of) another word.

---

[4] E.g. the only predecessor of IBM's Norwegian grammar [10].

[5] Adopted from the English PLNLP grammar, cf. [7].

2.3 The Lexicon

A prerequisite for high coverage as well as for correct parsing was an extensive lexicon and a corresponding morphology. Such data were provided by the current updated version of the Norwegian Bokmål base dictionary and morphology module [1]. The former contained more than 64,777 entry words and the latter was, in principle, complete.[6] Together, they covered close to 99 percent of the word forms of most running text.

Additional properties of words were stored in auxiliary files. They included traditional grammatical information such as valency, types of complement, and control characteristics for verbs, as well as collocation data and information that is more specific; these attributes are not usually found in linguistic literature. For instance, information about a given adverb's ability to appear in front of, in the middle of, or at the very end of a sentence, whether a given adjective could appear in an attributive position or not or whether an adverb could modify an adjective phrase as premodifier. As such, information for Norwegian was not accessible or simply inexistent at that time, quite a few person-months were invested in this type of basic descriptive linguistic research.

3 The Syntax

3.1 Architecture

The syntax had a sequential architecture: each step in the analysis added to the final structural description of the input string, thus producing a more precise and "deeper" analysis. When parsing began, the grammar received, from the lexicon, information about each word, in the form of collections of attribute-value pairs called records. During the processing, grammar rules combined the word records into constituent records, and then put these records together to form even larger record structures. Whether a particular rule would be applicable to a certain set of constituents was governed by the presence or absence of certain attributes, and their values, in the record structures. Some attributes were provided by the dictionary and some added by the rules themselves. Hence, the syntax produced a description of a sentence by incrementally building a record structure. Every possible combination of the records according to the syntax rules was tried until one (or more) successful parse(s) of the input string had been found. This meant that a high number of records that had not been used for the current successful parse(s) were created as well.

---

[6] Generating more than 485,000 unique (i.e. non-duplicate) wordforms. This particular dictionary module could easily be exchanged for a new, updated and still more extensive one. The last Bokmål module actually produced by IBM, contained approximately 160,000 lexemes (including some 30,000 proper nouns) generating 1,133,633 unique wordforms.

3.2 Records

Record structures could have as many attributes as necessary. Fig. 1 illustrates one example. On the left-hand side, are attribute names; to the right of each attribute name is its value.[7] Values could be either simple or complex.

```
Skriv en setning på norsk eller en kommando:

(prtrec 1)

            SEGTYPE    'FHSETN'
            STR         " Den ikke altfor flinke snekkeren sendte Kari en vakker faktura ."
            RULES      2040 2500 2500 2400 2900
            RULE        2900 SNTBEG1 VP1 PUNC1
            COPYOF     VP1 "Den ikke altfor flinke snekkeren sendte Kari en vakker faktura" ' SENDE'
            BASE        'SENDE'
            DICT        'sendte'
            INDIC      PRET V3
            PRMODS     NP1 "Den ikke altfor flinke snekkeren" 'SNEKKER'
            HEAD        VERB1 "sendte" 'SENDE'
            PSMODS     NP2 "Kari" 'KARI'
            PSMODS     NP3 "en vakker faktura" 'FAKTURA'
            PSMODS     PUNC1 "." '.'
            INDOBJ     NP2 "Kari" 'KARI
            FREMSTEV   VERB1 "sendte" 'SENDE'
            OBJEKT     NP3 "en vakker faktura" 'FAKTURA'
            TOOBJ      1
            NOKOBJ     1
            SUBJEKT    NP1 "Den ikke altfor flinke snekkeren" 'SNEKKER'
            HOVEDV     VERB1 "sendte" 'SENDE'
            PARSENO    1
            PRED        'SENDE'
            DSUB        NP1 "Den ikke altfor flinke snekkeren" 'SNEKKER'
            DIND        NP2 "Kari" 'KARI'
            DOBJ        NP3 "en vakker faktura" 'FAKTURA'
            XVPMODS    1
```

Fig. 1. Record structure with attributes.

Many of the values are themselves other records. Five attributes were essential: PRMODS 'premodifiers', HEAD, PSMODS 'postmodifiers', SEGTYPE 'segment type', and STR '(input) string'. Of the remaining attributes, some were provided by the system, the others chosen by the grammarian. The RULES attribute gave the derivational history of the parse by displaying an ordered list of rules, which had been applied at that level of constituent analysis. Feature markings on words and phrases were shown by the INDIC attribute. BASE showed the lemmatised form of the head word for any constituent. A POS attribute would have told what possible parts of speech that were returned from the lexicon for any given word. Functional information was also added to the record whenever possible. In the case above, both the subject, the direct and the indirect object of the sentence have been identified as the values of the SUBJEKT, OBJEKT, and INDOBJ attributes, respectively.

3.3 Rules

With very few exceptions, all the rules of the grammar were binary. As opposed to configurational rules, they have the advantage of predicting correctly the flexible order and the theoretically unlimited branching capacity of natural language constituent structures. Another characteristic of the rules was recursion. In general,

[7] "NP" and "VP" are abbreviations for 'noun phrase' and 'verb phrase' respectively.

PLNLP rules could apply several times. Further, attributions made at the first application could be changed at a later one. For instance, the verb complement pick-up rule was not only intended to handle direct object attachment, but also attachment of indirect object as well as subject and object predicatives. Each time the rule applied, the (preliminary) role attributed to each of the NPs of the VP was reconsidered in the light of information provided by the newly attached NP: An NP would be given the role of direct object in case there was not one there already, i.e., in the first pass of the rule. In a possible second pass, an NP already attributed the role of direct object, would be re-evaluated and given the role of indirect object, and the NP attached in the current pass would be marked as a direct object instead – unless the main verb required an object predicative. In the latter case, the NP attributed the function of direct object in the first pass, would continue to be the direct object, and the new NP attached in the second pass, would be marked as object predicative. Cf. the extract from the rule appears in Fig. 2.

```
(2500)    VP((---)
                  /// for å få til at 2500 virker to ganger /////////
                  /// etter verb som tar objektspredikativ //////////
              <OBJPRED!^METT>,
                   (---)
                  /// for å hindre "(gi den ut)VP nåNP"     /////////
              ^NOKOBJ,
                   (---))

              NP(---)

           -->  VP(
                   (---)
                  /// får til indirekte og direkte objekt  //////////
              <^OBJPRED,INDOBJ=OBJEKT,OBJEKT=NP,+TOOBJ>,
                  /// får til direkte objekt og objektspred. ////////
              <OBJPRED,
              <OBJEKT,OBJPREDIK=NP>,
              <^OBJEKT,OBJEKT=NP>>,
                   (---))
```

Fig. 2. Rule extract example.

4 Development

Writing the syntactic rules constituted the central activity of the project. This is why I shall elaborate on rule writing, while just mentioning the post syntax components in passing.

4.1 Staff

The project leader[8] acted as the “grammarian,” i.e. the person actually implementing the rules and testing them. He also coordinated the maintenance of the lexicon and the morphology as well as the development of the auxiliary material (syntactic information, corpora etc.). Research assistants were instructed to find and document

[8] Jan Engh.

combinability properties of words as well as information that might be of value for the future style component.

4.2 Rule Writing

To limit the scope of the phrase structure rules, thus preventing incorrect and otherwise unwanted parses, syntactic markers and conditions were used. In practice, the grammar writing was carried out as a process of rule writing and subsequent testing, involving both individual utterances and corpora.[9] Debugging functions allowed the user to pinpoint the exact place in a rule where a parse failed, or the exact differences between two ambiguous parses.

First, a rule was set to apply to a certain linguistic structure. Then the grammarian's task was to make that rule as precise as possible and to put limits to its scope so that it actually parsed the type of linguistic structure that the grammarian had in mind and nothing else. One simple example is direct object selection. For instance, the verb KJØPE 'buy' cannot take an infinitive as its direct object, while ØNSKE 'wish' can. In order to block object infinitives, a simple condition had to be written in the verb complement pick-up rule.

The first task was carried out partly based on the grammarian's own linguistic competence as a native language user and partly based on documented linguistic knowledge. Now, it is common knowledge that the description of all natural languages is far from complete. Based on what is actually stated in syntax literature, rules will never cover nor provide adequate structural descriptions of the sentences produced. This even holds for rules based on introspection. Hence, the use of corpora was essential.

Inevitably, rules may also produce unforeseen results, not least given the rapid complication of the rule structure itself. A consequence of the extrinsic and relaxed approach of rule writing was, thus, a proliferation of unintended parses. So, in a second phase, the grammarian had to set conditions – often in other rules – that would block these partly unpredictable and in any case unwanted parses. Some incorrect, others, in fact, correct, however inconvenient. In fact, a few central rules carried a heavy burden containing numerous filtering conditions, simply because they were to be applied after most other rules, offering the possibility to "control" the well-formedness of the input graphs generated by previous rules. These conditions were of a general nature, as already illustrated by the complement selection. However, there were exceptions. One was the case of "man."

Although there are sentences whose subject is an indefinite singular, such as (1),

---

[9] Two types of corpora were created for the purpose**.** One constructed with systematically fully expanded Norwegian sentences selected to contain well identified structures – each sentence with variants containing all possible word orders (both grammatical and ungrammatical). One "authentic" with ordinary Norwegian texts drawn from various areas of language use, partly typed in, as no Norwegian common language corpus was available to the project (examples from linguistics literature, samples of literary texts etc.), partly harvested from corporate business correspondence. Finally, a corpus consisting of grammar school pupils' essays – uncorrected and containing authentic errors and weaknesses – was purchased.

(1) Regn faller fra himmelen.
'Rain falls from the sky'[10]

"man", the indefinite singular a form of MAN 'mane' – a homonym of the indefinite pronoun MAN 'one' – had, in fact, to be blocked in subject position, i.e. an ad hoc blocking. This was done to prevent a highly improbable alternative parse of the sentence (2).

(2) Man hørte rop i det fjerne.
'One heard shouts in the far', i.e. 'One heard distant shouts'

The last example illustrates yet another characteristic of PLNLP-based syntax: information usually attributed little importance – or ignored – by grammarians is quite useful. One more example: There are lexemes (word types) that never appear as the subject of a sentence, for instance, the one marked for 'reciprocity'. Why bother? The syntax will never be given an absurd sentence such as (3) as input. However, the feature 'reciprocity' will provide a clue for producing only the correct parse of sentences such as (4). Without this clue, a second – and incorrect – parse would also be possible (5). [11]

(3) *Hverandre kjøpte mat.
'Each other bought food'.

(4) De så på hverandre tenkte sitt og gikk hvert til sitt.
[De så på hverandre] [(tenkte sitt] [og gikk hvert til sitt].
'[They looked at each other], [made up their own minds], [and went away]'

(5) *[De så på] [hverandre tenkte sitt] [og gikk hvert til sitt].
'[They looked at] [each other made up their own minds] [and went away]'

As already mentioned, an amazingly high number of unexpected parses turned out to be – just correct parses. For instance, (6) has, surprisingly, two correct interpretations, (7) and (8), as shown by the analysis trees in Fig. 3.

(6) De kjøpte huset.

(7) DeNP/SUBJECT kjøpteVP husetNP/DIRECT OBJECT.
'They bought the house'

(8) [De kjøpte]NP/SUBJECT husetVP.

[10] Here and later, the expression in single quotes represents a rough, word-by-word English gloss of the content of the Norwegian example – regardless of the possible ungrammaticality of the English rendering.
[11] Theoretically, one might have avoided the problem in this particular case by means of a comma rule. However, that would have made it more difficult to process sentences with a comma error, which, among other things, would have made the syntax unsuited as the base of a text critiquing system.

‘[The bought ones] gave shelter’

```
Skriv en setning på norsk eller en kommando:

De kjøpte huset.

----------------------------------------------------------
FHSETN1NP1               ADJ1          "De"
                                       VERB1*       "kjøpte"
                         VERB2*        "huset"
                         PUNC1         "."
----------------------------------------------------------
FHSETN2NP2               PRON1*        "De"
                         VERB1*        "kjøpte"
                         NP3           NOUN1*       "huset"
                         PUNC1         "."
----------------------------------------------------------
```

Fig. 3. Analysis trees.

It would be wrong to block the unexpected parse since, theoretically, it will always be possible to use (8), for instance as the response to a question (9), which, in turn, cannot be discarded on syntactic grounds. Cf. its structural resemblance with sentences such as (10).

(9) Hvem var det som huset [dem]?
‘Who was it that let them stay in their house?’

(10) [Hva var det de oppsagte arbeiderne gjorde?] De oppsagte streiket.
‘[What was it that the dismissed workers did?] The dismissed (ones) started a strike’

However, not all unexpected correct parses were that awkward. One perfectly normal sentence was (11), where the preposition phrase may be interpreted in two distinct ways, cf. (12) and (13), according to the context.

(11) Han spiste det brødet han bakte i går.
‘He ate the bread that he baked yesterday'

(12) Han spiste [det brødet [han bakte [i går]ADVP]RELATIVECLAUSE]NP/DIRECT OBJECT.

(13) Han spiste [det brødet [han bakte]RELATIVECLAUSE]NP/DIRECT OBJECT. [i går]ADVP.

In fact, the most common type of real ambiguity concerns attachment of this kind, which was marked in the parse tree to be solved later.

5 Post Syntax Processing

In a number of cases, one may find sentence internal clues for adjunct attachment. Cf. (14) and the two possible attachments for the preposition phrase, (15) and the somewhat awkward (16).

(14) Hun renset fisken med en kniv.

(15) Hun renset fisken [med en kniv]PP.
'She cleaned the fish by means of a knife'

(16) Hun renset [fisken med en kniv]NP.
'She cleaned the fish which had a knife'

Such cases were supposed to be taken care of by the reattachment component, which would use the information from a conventional lexicon to decide which one of the parses was the more probable. In its most primitive manner, simply by looking for the possible co-occurrence of "clean (a fish)" and "knife" within the same dictionary definition. However, this subcomponent was not fully developed. Neither were the ones for fitted parsing (assigning some reasonable structure to non-parsed input by the use of a "fitting" procedure, see [11]) and parse-metrics (evaluation of multiple correct parses for applications requiring unique parses, see [12, 13]). In addition, the component for surface structure interpretation, operating on the output of the reassignment components and intended to serve as the basis for further semantic processing, did not exceed the stage of prototype.

6 Style Component

The style component had two subcomponents, the style procedures and the features pertaining to the style critiquing application system, CRITIQUE. The former would detect stylistic errors. The latter would produce a set of explanations, ranging from the simple identification of the style error via advice for action to the relevant paragraph in an on-line textbook on correct style. Whenever required, a syntactic change such as moving a constituent was prepared – for later execution on request from the application user. For instance, one could identify "heavy" adjuncts between the finite and the infinite verb of (17), proposing a move to the front of the sentence (18).

(17) (?) Jeg ble i dag morges påkjørt av en sykkel.
'I was this morning run into by a bicycle.'

(18) I dag morges ble jeg påkjørt av en sykkel.

This example also illustrates the unclear border between style and syntax. On the other hand, it also shows how the style procedures could be used for borderline cases.

7 Project History

The project started in the late autumn of 1988, and was carried out in close cooperation with its sister projects abroad. In addition to the project leader/grammarian, five part-time assistants participated in the development, which was carried out on an IBM 370 mainframe under VM/CMS at IBM Norway premises. The development work went on until ASD cut the funding prematurely as of 3 August 1990. Still, significant results had been made. Most important: the first broad coverage analytic syntax for Norwegian.

8 The Continuation Project – Machine Translation

After the corporate funding cut, IBM Norway decided to continue the grammar project on its own. As a continuation of the writing tool development was out of question, grammar development was continued in order to produce a commercial system for automatic translation from Norwegian Bokmål to Nynorsk.

The machine translation technology adopted was a transfer based system developed at IBM Portugal [14–16]. A rough sketch of its architecture and the dataflow appears in Fig. 4.

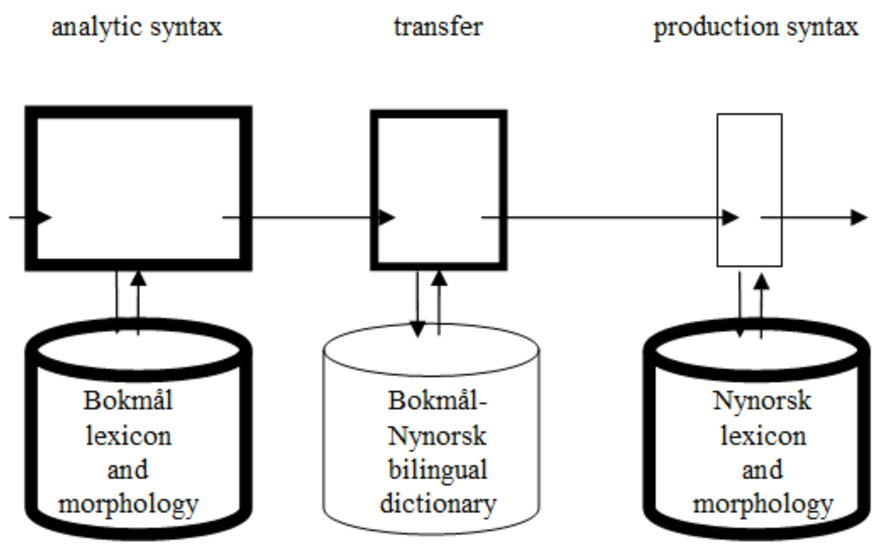


Fig. 4. Architecture and dataflow.

The width of the lines indicates the degree of completion at the time of the conclusion of this project. The Bokmål and Nynorsk lexica and morphologies were ready, the analytic syntax was extensive but not yet ready for commercial purposes, the transfer rules were also more or less ready, and the production syntax development was under way. The bilingual dictionary, however, was small.

The Bokmål-Nynorsk machine translation engine was a possible product with good market perspectives (in public sector administration and, above all, in the production of schoolbooks). Unfortunately, even this project turned out to be too expensive, and IBM Norway invited The Ministry of Administration as a partner in the application for a grant from Statens industrifond,[12] NOK 1,300,000, intended to finance Portuguese education and support and, above all, the compilation of an extensive bilingual Bokmål-Nynorsk dictionary. After half a year's reflection, the invitation was turned down (1991) on the grounds that IBM had not proved that it was possible to realise the project – strangely enough, since the prototype with fully developed system modules had been demonstrated to top officials and their technical aids.

9 Experience Gained

As for linguistics, unknown aspects of Norwegian syntax were discovered, relating to valency, preposition selection, and subject predicative agreement etc. One general observation: Actual Norwegian sentences are far more open to interpretation than usually acknowledged, as long as a strict separation of syntax and semantics is observed. From a methodological point of view, it became clear that a working broad coverage grammar could, in fact, be made without drawing heavily on modern linguistic syntactic and semantic theory. What turned out to be essential was extensive descriptive knowledge of Norwegian, for instance based on traditional and structuralist (positional) syntax.

As far as economy was concerned, the final, "national" phase of the project showed that relying on public grants for this type of development was a daring venture. The project obviously stumbled owing to higher public management's attitudes and personal antipathies, irrational factors that it will always be difficult to control. On the rational side, however, the main experience was probably that a huge and expensive enterprise such as the entire broad-coverage grammar project ought to be financed in several stages: Funding needs to be generated from successive implementation of modules in products and services already from an early phase of the project. The IBM grammar project depended on one unique corporate source of long term funding. There were no plans to generate funding before the very end of the project and this became critical when the world's economic crisis hit the corporation.

[12] The Norwegian State's Fund for Industrial Development.

Acknowledgments. I would like to thank Diana Santos for having read the manuscript and confirmed my recollections.

References

1. Engh, J.: Lexicography for IBM. Developing Norwegian Linguistic Resources in the 1980s. In: Impagliazzo, J., Järvi, T., Paju, P. (eds.) History of Nordic Computing 2. Second IFIP WG 9.7 Conference, HINC2, Turku, Finland, August 21–23, 2007, Revised Selected Papers. (IFIP advances in information and communication technology 303) Springer, Boston 258–270 (2009)
2. Engh, J.: Språkforskning i IBM Norge. [Paper read at Møte om norsk språk (MONS) IV, Oslo 15.-17.11.1991.] Printed in NORSKRIFT 72, pp. 16–36 (1992)
3. Engh, J.: Developing Grammar at IBM Norway 1988–91. Unpublished report. Oslo (1994)
4. Vikør, L.: The Nordic languages. Their Status and Interrelations. [3rd edition] (Nordic Language Secretariat. Publication 14) Novus, Oslo (2001)
5. Heidorn, G.: Natural language inputs to a simulation programming system. Ph.D. dissertation. Technical report, NPS-55HD72101A. Naval Postgraduate School, Monterey. Monterey, Cal. (1972)
6. Heidorn, G.: Augmented Phrase Structure Grammars. In: Nash Webber, B.L., Schank, R. (eds.) Theoretical Issues in Natural Language Processing. ACL 1–7 (1975)
7. Jensen, K.: PEG 1986: a Broad-coverage Computational Syntax of English. IBM research report. T.J. Watson research center, Yorktown Heights, N.Y. (1986)
8. Jensen, K., Heidorn, G., Richardson, S., Haas, N.: PLNLP, PEG, and CRITIQUE: Three Contributions to Computing in the Humanities. In: Proceedings of the 8th International Conference on Computers and the Humanities. Toronto, Canada (1986)
9. Jensen, K., Heidorn, G., Richardson, S. (eds.): Natural Language Processing: The PLNLP Approach. (Natural Language Processing and Machine Translation. The Kluwer International Series in Engineering and Computer Science 196) Kluwer, Hingham, Mass. (1993)
10. Lie, S.: Automatisk syntaktisk analyse. 1, Grammatikken (Rapportserie 17) NAVFs EDB-senter for humanistisk forskning, Oslo (1980)
11. Jensen, K., Heidorn, G., Miller, L., Ravin, Y: Parse Fitting and Prose Fixing: Getting a Hold on Ill-formedness. American Journal of Computational Linguistics 9, 3–4 (1983)
12. Heidorn, G.: An Easily Computed Metric for Ranking Alternative Parses. Paper presented at the 14th annual meeting of the ACL (1976)
13. Heidorn, G.: Experience with an Easily Computed Metric for Ranking Alternative Parses. Proceedings of the 20th Annual Meeting of the ACL (1982)
14. Santos, D.: An MT Prototype from English to Portuguese. In: Proceedings of the IBM Conference on Natural Language Processing (Thornwood, USA, October 24–26, 1988) (1988)
15. Santos, D.: Broad-coverage Machine Translation. In: Jensen, K., Heidorn, G., Richardson, S., (eds.): Natural Language Processing: The PLNLP Approach, pp. 101–115(118). Kluwer, Hingham, Mass. (1993)
16. Engh, J., Santos, D: Use of PORTUGA for the two Norwegian Written Standards. In: Jensen, K., Heidorn, G., Richardson, S., (eds.) Natural Language Processing: The PLNLP Approach, pp. 115–118. Kluwer, Hingham, Mass. (1993)