Interrupting Constructions in a Rejuvenated Amazon Grammar

Carla Schelfhout, Peter-Arno Coppen

Radboud University Nijmegen

Abstract

This paper reports on the latest rejuvenation of AMAZON, a structuralist parser for Dutch written sentences. Unlike older versions, the new AMAZON parser has been developed in a modular organization, with an empirical cycle containing evaluations on corpus material. This methodology facilitates the development by separate researchers, and it gives more insight into the actual performance of the parser, providing a useful means of measuring the improvement during development. In this paper, the evaluation method, and its outcome, is presented in general. As a more specific case study, the implementation of a separate module for interruption constructions is discussed.

1 Introduction

The AMAZON parser for Dutch ((Van Bakel 1975); (Van Bakel 1984); (Oltmans 1994); (Van Dreumel 1997); (Coppen 2002)) was originally developed to describe only grammatical, written Dutch sentences. Based on traditional structuralist theory (Rijpma and Schuringa 1968), the AMAZON parser aimed at an immediate constituent analysis of sentences in terms of structuralist fields (like Topicalization, Middle and Extraposition Field), without attempting to assign functional labels to the constituents. In the 1970s, theoretical coverage was the only research topic. The question was whether in principle the structuralist descriptive theory was adequate to cover all grammatical Dutch sentences. No attempt was made to determine the coverage of the parser on actual data.

This approach differs from more ambitious projects aiming at the development of a broad coverage —or more detailed— syntactic or semantic parser for Dutch (in (Bouma and Schuurman 1998), an overview of parsers currently available is given). Since all of these projects aim at different goals, a full systematic comparison is non-trivial. So far, such a full comparison has never been attempted, and we will not try to do so in this paper. In a special of the Dutch journal *Nederlandse Taalkunde* (Coppen and Cremers 2002), the results of four Dutch parsers on the same input are discussed.

In the course of time, the theoretical bias of the AMAZON grammar was replaced by more practical goals. First, the output of the AMAZON parser was used as input for a subsequent module aiming at a dependency structure (Van Bakel 1984) and second, the AMAZON grammar was provided with a robustness module (Oltmans 1994) to capture ungrammatical input. Finally, structural ambiguity in the AMAZON grammar was tackled ((Oltmans 1994); (Van Dreumel 1997); (Coppen 2002)), for instance by enriching the grammar with probability information, in order to make it possible to use the parser in practical applications (e.g. (Kerkhoff and Marsi 2002)).

From 1983 onwards, the AMAZON parser is organized as a two–level grammar that is converted into a parser by a parser generator (the AGFL system, (Koster 1991)). Since then, every once in a while, the grammar has been completely rejuvenated by rebuilding it from scratch (e.g. in ((Oltmans 1994); (Van Dreumel 1997)). In this paper we report on the latest rejuvenation (2001-2003).

We will show how AMAZON was rebuilt, and with what results. As a case study, we will focus on a separate module describing interrupting constructions.

2 Rebuilding the Amazon Grammar

Up until (Van Dreumel 1997), all AMAZON versions were developed in a purely linguistic way. That is to say, the grammar focused on the description of constructions on the basis of linguistic theory only. Although the parser seemed to perform reasonably well on unseen material, this was never evaluated systematically. In most cases, construed sentences were used to determine the parser's coverage. Evaluation merely meant a proof of principle. Whereas this was understandable from the initial purpose of the AMAZON parser (to be able to describe all sentences *in principle* and theoretically), it was not sufficient for realistic applications.

Another problem with the 1997 parser was the fact that sometimes it would behave unexpectedly. Although normally it would give 1 to 2 analyses per sentence within a second, for some sentences it would suddenly need minutes (or even hours) to run, or give 40 or more analyses.

In order to identify the cause of these problems, we decided in 2001 to rebuild the AMAZON grammar in a modular design, meaning that the grammar is generated from separate modules, which can be plugged in or replaced. Modules are not entirely independent in that they may refer to constituents defined in other modules. For example, the module describing prepositional phrases (the PP module) does not contain rules describing noun phrases, but it refers to the NP module, in which noun phrases are described. However, the PP module can be replaced by another PP module containing rules for all PPs referred to in other modules. New modules were carefully added incrementally, using regression tests on corpus material to monitor the performance of the evolving parser.

After building the description of the verbal structure (Van Dreumel and Coppen 2003), separate modules were added for the major constituents (NP, PP, AP), for the basic structuralist fields (Middle Field, Topicalization Field, Extraposition Field), and for peripheral fields (Left and Right Dislocation Fields). Preference measures were added to the rules, to favor more likely constructions, using standard AGFL mechanisms (cf. (Koster 1991)). Apposition and coordination were treated pragmatically: rather than enriching the grammar to determine the proper attachments, or underspecifying the structure, we decided to use a global attachment strategy (viz. maximal attachment/early closure for major constituents, minimal attachment/late closure for minor constituents such as noun-noun coordination). These attachment strategies were implemented by enriching the major structuralist fields with context information. For instance, NP postmodifiers will be accepted in a topicalization field, but not in the middle field. A PP will only be accepted at the end of the middle field if the verbal cluster is non-empty. Otherwise, the PP is attached to the extraposition field.

These pragmatic choices seem justified (cf. (Coppen 2002)) because in subsequent modules, the structural environment for attachment problems can be easily recognized, so that the attachment can be adapted when necessary. For instance, any PP following an NP is a possible candidate for appositional attachment. Whether it is an appropriate candidate depends on matters like subcategorization of the verb, semantic content and the like. In (Van Bakel 1984), the module CASUS is described that deals with these matters. Without entering into too much detail, this process can be characterized as a transformational grammar recognizing a structural description and changing the attachment whenever necessary (i.e. whenever the PP cannot be interpreted as an object or an adverbial).

As a basis for the lexicon, for the open word classes N, V and A, the CELEX lexicon was used. In addition, wild card rules were added to the grammar to cope with unknown words, using standard AGFL mechanisms (cf. (Koster 1991)).

Initially, for development purposes, we used two documents with the Dutch State of the Nation ("troonrede") from 2000 (initially) and 2003 (later on). At the end, the versions from 2001 and 2002 were used to determine the total performance, and to add some final tuning.

This methodology, incrementally adding separate modules and testing them on corpus material, enabled us to identify, and tackle, ambiguity problems one by one, and independently. This way, the problems of the older parser were all prevented.

3 Evaluating the Amazon parser

Evaluation of the parser during development consisted of a thorough manual judgment of the quality of the analyses of all sentences from the data. As the system evolved, we used an automated measure of coverage to be able to determine the performance on larger corpora.

In order to test the performance of the parser, we collected a number of corpora with different text types (cf. Table 1), from a children's story "Jip & Janneke" which consisted almost only of dialog to some editorials from a high quality news paper (the NRC)[1].

Since the latest AGFL version (2.3), analysis time does not seem to be an issue any more. Although the word throughput on various text types varies from 223 words per second in the Daily News section of the Eindhoven Corpus to 576 words per second on the child story "Jip & Janneke", the worst performance still parses the entire Daily News section of the Eindhoven Corpus (Uit den Boogaart 1975) in less than 10 minutes on a modest 800 Mhz PC.

Ambiguity was almost completely eradicated from the parser, by applying global attachment strategies and employing preference measures (cf. section 2). Of course, this way of eradicating ambiguity will sometimes result in the wrong parse, or an incomplete parse. It is the purpose of evaluation measures as discussed

[1] The material also included a corpus of "unedited prose" (Van Halteren 2004) which consisted of raw text fragments collected from informal, diary-like documents, and the Daily News Section of the Eindhoven corpus (cf. (Uit den Boogaart 1975)).

Table 1: Test corpus characteristics

Corpus	sentences	words	w/s
child story ("Jip & Janneke")	267	1580	5.92
fairy tales	302	3618	11.98
internet news	1426	20718	14.53
state of the Nation (development)	321	4946	15.41
unedited prose	4488	70027	15.60
state of the Nation (test)	325	5207	16.02
NRC editorial	75	1225	16.33
Daily News Eindhoven corpus	7137	126932	17.78

below to determine these costs. Furthermore, this strategy relies on subsequent modules that have to be evaluated in the future.

Testing on large corpora showed a mean number of 1.39 parses per sentence (with 75% of the sentences receiving 1 parse), ambiguity almost always resulting from lexical sources. In the future, we will employ statistical means (e.g. adding lexical probability taken from the CELEX lexicon, or using the output of a part–of–speech tagger as input to AMAZON) to get rid of this ambiguity as well.

We determined the AMAZON performance on these corpora, first with a rough measure, distinguishing just three possibilities for a sentence: either a full sentence analysis from the AMAZON core grammar, or a result from the robust module, in which the sentence is analyzed as an ellipsis consisting of (as large as possible) constituent chunks. A third possibility is that the parser produces no analysis at all. Using this measure, we get results as in Table 2.

Table 2: AMAZON Performance Statistics

Corpus	analysis		
	Full	Elliptic	None
Daily News Eindhoven corpus	4305 (60%)	2822 (40%)	10
fairy tales	213 (71%)	89 (29%)	0
NRC editorial	53 (71%)	22 (29%)	0
unedited prose	3379 (75%)	1107 (25%)	2
internet news	1123 (79%)	303 (21%)	0
child story ("Jip & Janneke")	229 (86%)	38 (14%)	0
state of the Nation (test)	285 (88%)	40 (12%)	0
state of the Nation (development)	317 (99%)	4 (1%)	0

Of course, a full analysis must not be identified with a correct analysis, and an elliptic analysis is not always an inferior result. Note that in some cases (especially in child stories, fairy tales and unedited prose) the input is indeed elliptic, which makes the elliptic analysis the only possible one (even for a human parser) and therefore, the correct one. A full sentence analysis obviously does not necessarily imply a fully correct analysis. However, random spot checks suggest that full analyses are for the most part correct or at least defendable. A more detailed error analysis will have to determine the quality of full sentence analyses in the future.

In order to obtain more insight into the qualitative performance of the parser, a full comparison with a gold standard analysis is necessary. However, for practical reasons[2], we decided to perform only partial evaluations. First, we manually extracted all verbal structures (in Dutch: *werkwoordelijk gezegde*) from the fairy tale subcorpus and the NRC editorial subcorpus. Verbal structures were defined in a traditional sense, as a main verb or copular verb possibly modified by auxiliaries. We compared these to the AMAZON results. The two subcorpora were chosen because the former is one of the simplest in terms of verbal constructions, whereas the latter is the most complex. Results are in Table 3.[3]

Table 3: AMAZON performance on verbal cluster

Fairy tale subcorpus						
target	correct	false	not found	precision	recall	F-score
851	741	100	110	0.88	0.87	0.88
NRC subcorpus						
target	correct	false	not found	precision	recall	F-score
128	100	13	28	0.88	0.78	0.83

As expected, AMAZON scores a little bit worse on the more difficult corpus with an F-score[4] of 0.83, whereas on the “easy” corpus the F-score is 0.88. The difference is entirely due to the lower recall. Lower recall results from the fact that elliptic analyses of sentences with single word verbal structures will in many cases have detected this single verb correctly, whereas multiword verbal structures will not be detected so easily in elliptic sentences. The fairy tale corpus contains more sentences with single word verbal structures than the NRC corpus. Therefore, recall of verbs will be higher.

A second test was performed by inspecting base NPs[5] in three subcorpora: the

[2] We were not able to match existing treebanks, like the ALPINO treebank or the CGN corpus, with the structural description that AMAZON provides. One of the main reasons for this was that the treebank analyses were dependency structures, whereas AMAZON aims at constituent structures.

[3] In this and following tables, *target* is the number of constructions to be detected, *correct* is the number of correct detections, *false* is the number of false detections, and *not found* is the number of (target) constructions that remain undetected. So, *target* is the sum of *correct* and *not found*, *precision* is the division of *correct* by the sum of *correct* and *false*, and *recall* is the division of *correct* by *target*.

[4] The F-score is computed by doubling the division of the product of Precision and Recall by their sum (“harmonic mean”). The F-score ranges from 0 to 1.

[5] Base NPs are Noun Phrases without postmodifiers. Identifying Base NPs is a well-known task in the field of NLP (e.g. (Tjong Kim Sang 2000)). F-Score results are usually in the range of 0.87 to 0.95. This is slightly better than the AMAZON performance, but note that this result is achieved on tagged

fairy tale subcorpus, the NRC subcorpus and the State of the Nation 2003. The former two were chosen because they were expected to contain the most simple and the most complex utterances respectively, and the latter corpus was added for reference, since it was part of the development corpus. This should give us an idea of the best performance. We manually counted all base NPs in the first 50 sentences of all three corpora. Results are in Table 4.

Table 4: AMAZON performance on base NP

Fairy tale subcorpus						
target	correct	false	not found	precision	recall	F-score
218	195	39	23	0.83	0.89	0.86
NRC subcorpus						
target	correct	false	not found	precision	recall	F-score
217	199	44	18	0.82	0.92	0.87
State of the Nation 2003						
target	correct	false	not found	precision	recall	F-score
215	203	18	12	0.92	0.94	0.93

As it appears, AMAZON scores slightly better on formal prose, which is understandable since this is the text type that the original AMAZON description was based on.

A final test was performed by comparing the Noun Phrase detection by AMAZON with the Newspaper part of the Eindhoven corpus, as annotated in the CD-ROM version of the ALPINO Treebank (Van der Beek et al. 2001). Although the ALPINO Treebank does not give a real constituent analysis (it gives a dependency structure, in which constituents may be formed from words that are not adjacent in the original word order), the syntactic annotation of noun phrases seems to follow the original word order in the sentence. We extracted only Base NPs, without postmodifiers, and compared them with the AMAZON analysis (cf. Table 5).

We compared the results in three ways: first, we compared only head detection (how many Noun Phrase heads were detected correctly), and then full (base) Noun Phrases. Since it seemed that many Noun Phrases were detected almost correctly, we also computed a third measure in which detection was compared at word level. Every word from a target NP also included in a detected NP was counted as correct, even if the detected NP was not identical to the target. For instance, if ALPINO considers *nog een ruime marge* as a NP and AMAZON decides that only *een ruime marge* is a NP, a word measure count will score 3 correct words on a target of 4, no false hits, and one word missed.

It should be noted that these figures cannot be taken as an absolute performance measure, but rather as an indication of the agreement between AMAZON and the ALPINO treebank. Upon random inspection it seems that some decisions in the

material. AMAZON runs on untagged text.

Table 5: AMAZON performance on NP detection in ALPINO Treebank

On NP head						
target	correct	false	not found	precision	recall	F-score
37266	30925	8703	6341	0.78	0.83	0.80
On full (base) NP						
target	correct	false	not found	precision	recall	F-score
37266	29207	13519	8059	0.68	0.78	0.73
On NP words						
target	correct	false	not found	precision	recall	F-score
70719	66181	3196	4538	0.95	0.94	0.94

ALPINO treebank can be seriously questioned (and actually, have been altered in the past). For instance, it seems that in some cases the human ALPINO annotators at first decided to consider certain adverbials as focus adverbials, to be attached to the NP, like in the following example 1, the very first sentence of the corpus[6]:

(1) De verzekeringsmaatschappijen verhelen niet dat *ook de*
the insurance companies hide not that also the
rentegrondslag van vier procent *nog een ruime marge* laat
interest base of four percent yet a considerable margin leaves
ten opzichte van de thans geldende rentestand .
compared to the current interest rate.

Whereas the attachment of the modifier *ook* to the NP *de rentegrondslag* may indeed be defended[7], attaching *nog* to *een ruime marge* is certainly not the best option[8]. Since AMAZON structurally does not attach these modifiers to the NP (except when they occur within PP or in topicalized position), its NP precision will decrease, but on a word level the effect will be less strong.

It may be expected that AMAZON performs better with respect to NP detection in sentences with a full analysis. If we compare only the sentences with full analysis (60% of the corpus), the F-score on NP head detection increases from 0.80 to 0.84, on full NP detection it increases from 0.73 to 0.78, and on word level, the F-score increases from 0.94 to 0.95. This effect is mainly due to the improvement in precision. This is understandable, since in an elliptic analysis, AMAZON often decides on a noun analysis in case of a lexically unknown word. Therefore, more nouns will be wrong in elliptic analyses. It may be expected that the

[6] At least in the CD-ROM version. On the website, the analysis has been adapted. In this example, the appositional PP *van vier procent* is attached to the NP *ook de rentegrondslag* by ALPINO. Recall that AMAZON does not attach these PPs to the NP.

[7] The whole NP *ook de rentegrondslag van vier procent* may be preposed. However, *ook* may also be a separate adverbial. This can be argued by the observation that an adverbial like *volgens hen* "according to them" can occur at this position. Such an adverbial is uncontroversially non-appositional.

[8] The whole NP *nog een ruime marge* cannot be moved in this sentence.

performance improves when the AMAZON input is filtered by a statistically based part–of–speech tagger[9].

The tests on NP detection and verbal cluster analysis indicate that the grammar performs reasonably well on a basic level. For special constructions, similar tests have to be carried out. When new parts of the grammar have been developed, they can be evaluated by performing these tests and determining whether the adaptations resulted in a real improvement of the parser's performance. We will show an example of such an evaluation in the next section, with respect to the implementation of the interruption construction.

4 Interruptions in Amazon

An immediate constituency grammar like AMAZON runs into problems when it encounters a construction that is not described by the rules. This can happen whenever this construction does not really form a part of the clause but is more like a comment to it, as is the case with finite comment clauses (or: parentheticals), reporting clauses, interjections or forms of address. These constructions are illustrated in examples 2–4.

(2) Dat is de man, *denk ik*, die gisteren mijn arme kat een schop gaf.
That is the man think I who yesterday my poor cat a kick gave.
'That is the man, I think, who kicked my poor cat yesterday.'

(3) "Dat is hem," *zei hij*, "hij schopte gisteren mijn arme kat."
That is him said he he kicked yesterday my poor cat.
"'That's him," he said, "he kicked my poor cat yesterday."'

(4) Waarom heb je dat *verdomme* gedaan?
Why have you that damn done?
'Why the hell did you do that?'

Such constructions merely interrupt the clause rather than that they are part of it. However, since the examples are perfectly grammatical Dutch, in our description of Dutch we have to include interruption constructions. In order to do so, we need the answers to two questions: at which positions in the sentence do interruption constructions occur and in which forms do they occur? Previous studies ((Schelfhout 1999); (Schelfhout, Coppen, and Oostdijk 2003); (Schelfhout, Coppen, and Oostdijk n.d.)) into finite comment clauses (as in example 2), reporting clauses (as in example 3) and interjections (as in example 4) have shown that these three constructions tend to occur exactly on the boundaries of the fields described by structuralist theory, with the exception of the position between the Middle Field and the verbal cluster. In addition, interruptions occur at a limited number of positions within the Middle Field.

[9] This research is currently being carried out in an undergraduate project by MA student Herman Heringa.

Obviously, all three constructions also occur at the end of the sentence. Only interjections are allowed at the beginning of the sentence. Interjections can also form utterances in themselves.

About the form of interruption constructions the studies report that interjections can be single words (*ja*, “yes”), multiwords (*kom nou*, “come on”) or a combination of interjections (*ja ja*, “yes yes”), possibly separated by commas. Finite comment clauses and reporting clauses are very much alike: they consist of a finite verb and the subject, optionally preceded by the word *zo* “so”, and optionally followed by objects, modifiers and other verbs. Some examples are given in 5–8.

(5) Hij was bang, *denk ik*, dat dat geen goed idee was.
He was afraid, think I, that that no good idea was
‘He was afraid, I think, that that was not a good idea.’

(6) Hij was bang, *zo denk ik*, dat dat geen goed idee was.
He was afraid, so think I, that that no good idea was
‘He was afraid, so I think, that that was not a good idea.’

(7) Hij was bang, *zei hij in de trein*, dat dat geen goed idee was.
He was afraid, said he in the train, that that no good idea was
‘He was afraid, he said in the train, that that was not a good idea.’

(8) Hij was bang, *zo zei hij in de trein*, dat dat geen goed idee was.
He was afraid, so said he in the train, that that no good idea was
‘He was afraid, so he said in the train, that that was not a good idea.’

Each type of clause also has a special, more formal variation: reporting clauses can take the form of the word *aldus* “according to” followed by a noun phrase, and finite comment clauses can consist of an optional *zo* “so”, followed by a copula, optionally followed by a clitic. Like in the standard forms, modifiers are possible in these special forms as well. Some examples are given below:

(9) Hij was bang, *zo bleek gisteren*, dat dat geen goed idee was.
He was afraid, so appeared yesterday, that that no good idea was
‘He was afraid, so it appeared yesterday, that that was not a good idea.’

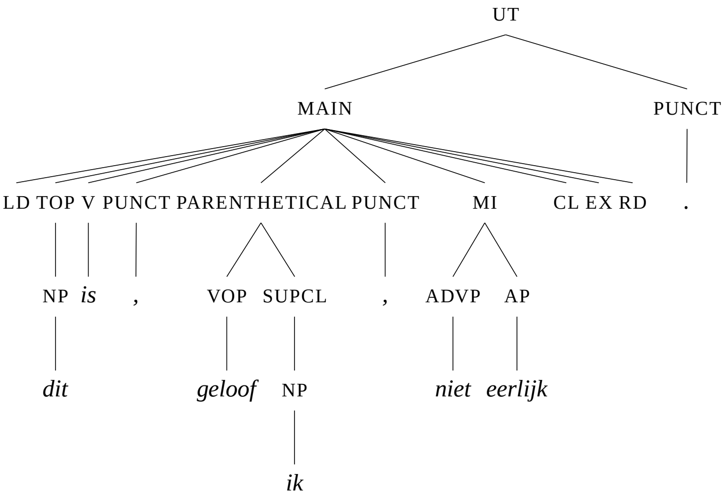
(10) Hij was bang, *bleek het*, dat dat geen goed idee was.
He was afraid, seemed it, that that no good idea was
‘He was afraid, it seemed, that that was not a good idea.’

(11) Hij was bang, *aldus zijn broer*, dat dat geen goed idee was.
He was afraid, according to his brother, that that no good idea was
‘He was afraid, according to his brother, that that was not a good idea.’

Parallel to developments on other parts of the new AMAZON grammar, these findings were described in a separate grammar module, called the *interruption* module. The development of this module was organized in the same cyclic method as the total AMAZON system: first we implemented the results of the descriptive studies, analyzed the corpus sentences that were used in these studies with the new parser and checked whether our implementation was complete and correct by performing a manual check of the analyses. Second, we analyzed new material with the new parser and extended the interruption module with types of interruption constructions that were not described in the literature but found in a manual check of corpus material.

Because of the similarities between finite comment clauses and reporting clauses, they were implemented together under the term ‘parenthetical’. For an example analysis according to the interruption module see Figure 1.

Figure 1: An example analysis of a sentence containing an interruption



We tested the interruption module on new material. From the internet we derived a small corpus of texts with their origin in print: 3 essays, 401 sentences in total, 3 interviews, 555 sentences in total and 3 short stories, 761 sentences in total. These text types were chosen because a previous study (Schelfhout, Coppen and Oostdijk 2003) showed that finite comment clauses and interjections occur relatively frequently in these types of text. These texts were automatically preprocessed using a tokenizer developed for English and Dutch (Van Halteren, personal communication): they were split up into sentences, and diacritic symbols were removed[10]. The total number of sentences is 1717; the total number of words is 26,527. A manual check of the preprocessing revealed some unexpected behavior of the preprocessing module. As it appeared, some 5% of the sentences were split at a point that did not conform to the structuralist description AMAZON was based

[10] The preprocessor does not accept higher ASCII signs, so accents, diaereses and the like had to be removed.

on. Besides that, although the test material had been edited before publishing, some spelling errors have remained. It should be remarked that this puts an upper bound on the AMAZON performance results.

In order to test the effect of the new module on the total parser, we analyzed this material with the interruption module switched off and on. The rough coverage results are in Table 6.

Table 6: Interruption Module Performance Statistics

Corpus	analysis		
	Full	Elliptic	None
without interruption module	1231 (72%)	484 (28%)	2 (0%)
with interruption module	1260 (73%)	455 (26%)	2 (0%)

It appears that the AMAZON parser with the new interruption module is able to attain more full sentence analyses than without it. This quantitative improvement does not seem spectacular, due to the relatively low frequency of interruption constructions on the one hand and the upper bound effect from the preprocessor on the other hand (recall that 5% of the sentences after preprocessing did not conform to the AMAZON description). However, it can be expected that there is also a qualitative improvement in that more constructions are recognized as interruptions and not erroneously parsed as other constituents.

In order to determine this qualitative improvement, we manually counted the parentheticals and interjections in our test corpus[11]. This table does not have figures for the results without interruption module because, of course, in that case no parentheticals or interjections are detected. The results are in Table 7.

Table 7: AMAZON performance on interruption constructions

Parentheticals						
target	correct	false	not found	precision	recall	F-score
62	54	29	8	0.65	0.87	0.74
Interjections						
target	correct	false	not found	precision	recall	F-score
65	49	7	16	0.88	0.75	0.81

As can be seen, the interruption module reaches an F-score of 0.74 on parentheticals and 0.81 on interjections. On parentheticals, precision is low, because too many cases are considered parenthetical, and on interjections, recall seems to be problematic. This may be a lexical problem[12], that will be tackled in the future

[11] A combination of adjacent interjections was counted as one interjection.

[12] A spot check gave *teeeering* which—unlike its base form *tering* “hell”—is not in the lexicon.

by adding statistical information from the CELEX lexicon or by using statistically based part-of-speech tagging. On the whole, these scores imply that the quality of the analyses of sentences that do contain an interruption has indeed become better.

5 Conclusion

In this paper we looked at two methodological issues in the rejuvenation of the AMAZON parser: the modular design and the evaluation on actual data. The new modular organization enables individual researchers to work on separate projects simultaneously, and it facilitates evaluating the parser's performance on corpus material by switching separate modules on and off. This way the influence of a separate module can be determined precisely.

A number of evaluation measures on actual data have been used in the development of the new AMAZON parser. In addition to a thorough manual inspection of all analyses, a rough coverage measure has proved to be useful. In order to determine the quality of the parser's performance, some partial evaluations have been performed manually. Automatic evaluation on the basis of a gold standard proved to be difficult, because of the lack of a treebank which is syntactically annotated in the structuralist style. However, tentative experiments were performed on the ALPINO treebank.

The partial evaluation experiments show an AMAZON performance that differs slightly for various text types, with F-scores in the range of 0.86 to 0.93 (manually counted base NPs and verbal constructions). A worse performance on full NP detection seems the result of an automated comparison with the ALPINO treebank. However, the word measure reached an F-score of 0.94, suggesting that there may be some structural differences in syntactic annotation involved. In the future, we will attempt to improve these scores by enhancing the lexical module with probability information. Also, more research is needed on treebank evaluation.

References

Bakel, J. van (1975). *Automatische zinsontleding met de computer*, internal publication, University of Nijmegen.

Bakel, J. van (1984). *Automatic Semantic Interpretation, A Computer Model of Understanding Natural Language*, Foris, Dordrecht.

Beek, L. van der, Bouma, G., Malouf, R. and Noord, G. van (2001). The Alpino Dependency Treebank, *in* Theune, M., Nijholt, A. and Hondorp, H. (eds), *Computational Linguistics in the Netherlands 2001. Selected Papers from the Twelfth CLIN Meeting.*, Rodopi, Amsterdam, New York, pp. 8–22. No 45 of *Language and Computers: Studies in Practical Linguistics* (edited by Aarts, J. and Meijs, W.).

Bouma, G. and I. Schuurman (1998). *De positie van het Nederlands in Taal- en Spraaktechnologie* Report Nederlandse Taalunie,
`www.taalunie.org/_/publicaties/rapporten/01/rapport.pdf`.

Coppen, P.A. (2002). Het Geheim van de Oude Dame. De Nijmeegse Parser Amazon, *Nederlandse Taalkunde*, Vol 7, No 4, pp 312-334.

Coppen, P.A. and Cremers, C. (2002). Parseren in de polder. Nederlandse taaltechnologie in perspectief, *Nederlandse Taalkunde*, Vol 7, No 4, pp 305-311.

Dreumel, S. van (1997). A Robust parser for Dutch Sentences, abstract PhD project,
`http://lands.let.kun.nl/˜dreumel/PhD_project.nl.html`.

Dreumel, S. van and Coppen, P.A. (2003). Surface Analysis of the Verbal Cluster in Dutch, *Linguistics*, Vol 41, No 1, pp. 51–81.

Halteren, H. van (2004). Detection of Plagiarism in Student Essays, *in* Decadt, B., De Pauw, G. and Hoste, V. (eds), *Computational Linguistics in the Netherlands 2003. Selected Papers from the Fourteenth CLIN Meeting.*

Kerkhoff, J. and Marsi, E. (2002) NeXTeNS: a New Open Source Text-to-speech System for Dutch, abstract CLIN 2002.

Koster, C.H.A. (1991). Affix Grammars for programming languages, *in* Alblas, H. and Melichar, B. (eds), *Attribute Grammars, Applications and Systems, International Summer School SAGA, Prague, Czechoslovakia, June 1991. Lecture Notes in Computer Science, volume 545*, Springer-Verlag, pp. 469–484.

Oltmans, J.A. (1994). *Amazon in AGFL: een contextvrije herschrijfgrammatica voor de structurele module van het AMAZON/CASUS-systeem, beschreven in het AGFL-formalisme* , undergraduate report, University of Nijmegen.

Rijpma, E. and Schuringa, F.G. (1968). *Nederlandse spraakkunst*, ed. by Bakel, J. van, Wolters-Noordhoff, Groningen.

Schelfhout, C. (1999). *Een onderzoek naar plaats en vorm van de reporting clause in parenthetische directe rede-constructies*, Master's thesis, University of Nijmegen.

Schelfhout, C., Coppen, P.A., and Oostdijk, N. (2003). Positions of parentheticals and interjections: A corpus-based approach, *in* Fikkert, P. and Cornips, L. (eds.), *Linguistics in the Netherlands 2003*, John Benjamins Publishing Company, Amsterdam, pp. 155–66.

Schelfhout, C., Coppen, P.A., and Oostdijk, N. (n.d.). Finite comment clauses in Dutch: a corpus-based approach, submitted to *Journal of Germanic Linguistics*, 2004.

Tjong Kim Sang, E. and Buchholz, S. (2000). Introduction to the CoNLL-2000 Shared Task: Chunking, *in Proceedings of the Fourth Conference on Computational Language Learning and of the Second Learning Language in Logic Workshop*, 13-14 September 2000, Lisbon, Portugal, pp. 127-133.

Uit den Boogaart, P.C. (1975). *Woordfrequenties in geschreven en gesproken Nederlands*, Oosthoek, Scheltema and Holkema, Utrecht.