DETECTING GRAMMAR ERRORS WITH LINGSOFT'S SWEDISH GRAMMAR CHECKER

Juhani Birn
Lingsoft, Inc.
jbirn@lingsoft.fi

Abstract

A Swedish grammar checker (Grammatifix) has been developed at Lingsoft. In Grammatifix, the Swedish Constraint Grammar (SWECG) framework has been applied to the task of detecting grammar errors. After some introductory notes (chapter 1), this paper explains how the SWECG framework has been put to use in Grammatifix (chapter 2). The different components of the system (section 2.1) and the formalism of the error detection rules (section 2.2) will be overviewed, and the relationship between grammar errors and disambiguation will be discussed (section 2.3). Work on the avoidance of false alarms is also described (chapter 3). Finally, test results are reported (chapter 4).

1. Introduction

The purpose of this paper is to explain how Grammatifix goes about its task of detecting grammar errors. The paper by Arppe (this volume) addresses the more general level design principles in the development of Grammatifix, and provides also a background to the field of Swedish grammar checking in general.

Grammatifix has checks on three kinds of phenomena: grammar errors, graphical writing convention errors, and stylistically marked words.[1] For these phenomena different detection techniques are used: SWECG, matching of regular expressions against character sequences, and lexical tagging, respectively. This paper is concerned with grammar error detection.

Prototypical grammar errors can be understood to be norm violations that are to be identified in contexts larger than the word (cf. spell-checking) where the contexts are morphosyntactically explainable. Of errors so defined, no computational grammar checker is able to control more than a (more or less) modest part. A realistic grammar checker concentrates on central categories of the language's grammar, and, within those categories, on common, simple patterns that allow precise descriptions. The error categories targeted by Grammatifix are presented in Arppe & al. (1999), for a listing with examples see also Arppe (this volume).

2. Constraint Grammar as a framework for grammar error detection

Constraint Grammar (CG) is a framework for part-of-speech disambiguation and shallow syntactic analysis, as originally proposed by Karlsson (1990). The basic principles and the formalism of CG are fully explained in Karlsson & al. (1995). A short presentation of SWECG is given in Birn (1998). In Grammatifix, the CG framework is used for the purposes of grammar error detection.

2.1. Overview of the error detector's components

The CG-based error detection system consists of five sequential components as listed below (1-5). In a formal sense the componets are the same as in SWECG, but, contentwise, the components of the two systems are not identical. There are some differences even in components (1, 2), some more in component (3), and components (4, 5) are wholly application-specific.

(1) Preprocessing
(2) Lexical analysis
(3) Disambiguation
(4) Assignment of the tags @ERR and @OK to each word
(5) Error detection rules, i.e. rules for the selection of @ERR

Proceedings of NODALIDA 1999, pages 28-40

Preprocessing. The preprocessor (or tokeniser) identifies words, abbreviations, punctuation marks, and fixed syntagms. A fixed syntagm is a multi-word expression identified as a lexical unit, e.g. the words *till hands* are identified as a unit, *till_hands*, analysed as an ADV[2]. This treatment entails that the error detector avoids false alarms that might follow (in unexpected contexts, e.g. *funnits till hands dygnet om*) if a genitive feature was present in the analysis of *till hands*.

The tasks performed by componets (2–5) will be illustrated with a stepwise analysis of the relevant (here boldfaced) parts of the example sentence given below. The error to be detected is the definite form *stavningen* as governed by the genitive *vilkas*. The analysis of the sequence *många engelska* also illustrates a relevant point.

Det finns ***många engelska lånord vilkas diskontinuerliga stavningen*** *inte tycks bereda språkbrukarna några problem.*
(From *Språket lever. Festskrift till Margareta Westman*. Norstedts 1996:68.)

Lexical analysis. The main module here is the SWETWOL analyser (Karlsson 1992; cf. also Birn 1998). As illustrated below each word is here given one or more readings. For example, *många* has two readings, DET (implying modifier status) and PRON (implying head word status), and *engelska* has three readings, one of them N SG. The sequence *många engelska* illustrates why it was obvious from the start that disambiguation should be used: *många* is **PL** and *engelska* is N **SG** (inter alia), but flagging this as a number agreement error would be a false alarm, of course. Disambiguation is needed for the sake of precision.

```
"<många>"
    "mången" <ID> DET UTR/NEU INDEF PL NOM
    "mången" PRON UTR/NEU INDEF PL NOM
"<engelska>"
    "engelsk" A UTR/NEU DEF SG NOM
    "engelsk" A UTR/NEU DEF/INDEF PL NOM
    "engelska" N UTR INDEF SG NOM
"<lånord>"
    "lån_ord" N NEU INDEF SG/PL NOM
"<vilkas>"
    "vilken" <WH> <CLB> <MD> DET UTR/NEU INDEF PL GEN
    "vilken" <WH> <CLB> PRON UTR/NEU INDEF PL GEN
"<diskontinuerliga>"
    "diskontinuerlig" A UTR/NEU DEF SG NOM
    "diskontinuerlig" A UTR/NEU DEF/INDEF PL NOM
"<stavningen>"
    "stavning" N UTR DEF SG NOM
```

Disambiguation. The disambiguation rules of SWECG have been adopted to a large extent as such in Grammatifix, but, importantly, there are differences. The differences are a consequence of the efforts, in Grammatifix, to overcome certain disambiguation disturbances due to grammar errors (for more on this point see section 2.3). Full disambiguation is not a goal as such for Grammatifix, and some of the error detection rules are formulated so as to tolerate ambiguities or even incorrect disambiguations (section 2.3). In the example sentence of this section, the disambiguator selects the appropriate reading for each word, e.g. *engelska* is disambiguated as A PL as shown below.

```
"<många>"
    "mången" <ID> DET UTR/NEU INDEF PL NOM
"<engelska>"
    "engelsk" A UTR/NEU DEF/INDEF PL NOM
"<lånord>"
    "lån_ord" N NEU INDEF SG/PL NOM
```

Assignment of the tags @ERR and @OK to each word. In ordinary CG the component called 'Morphosyntactic mappings' assigns a number of syntactic tags (subject, object, premodifier, etc.)

to each remaining reading. In Grammatifix this component performs a trivial task: each reading is assigned two more tags, @ERR (error) and @OK (no error), as shown below for *många*.

```
"<många>"
    "mången" <ID> DET UTR/NEU INDEF PL NOM @ERR @OK
```

Error detection rules, i.e. rules for the selection of @ERR. In ordinary CG the component called 'Syntactic constraints' performs syntactic disambiguation, i.e. there are rules that try to select the contextually appropriate syntactic tags. In Grammatifix this component contains error detection rules, i.e. rules for the selection of the tag @ERR for those words where an error can be located. In the example, @ERR lands on *stavningen*, and all other words get @OK. The words with @ERR, possibly together with some of the surrounding words, are flagged to the user.

```
"<många>"
    "mången" <ID> DET UTR/NEU INDEF PL NOM  @OK
"<engelska>"
    "engelsk" A UTR/NEU DEF/INDEF PL NOM @OK
"<lånord>"
    "lån_ord" N NEU INDEF SG/PL NOM @OK
"<vilkas>"
    "vilken" <WH> <CLB> <MD> DET UTR/NEU INDEF PL GEN  @OK
"<diskontinuerliga>"
    "diskontinuerlig" A UTR/NEU DEF SG NOM @OK
"<stavningen>"
    "stavning" N UTR DEF SG NOM @ERR
```

The selection of @ERR is performed by rules which use the CG disambiguation rule formalism (section 2.2). For the above case the rule is in basic outline as shown below. This formulation, a formally valid CG rule, is simplified in the sense that here are not included any of the additional conditions used for the avoidance of false alarms (chapter 3).

```
Error detection rule (simplified):
(@w =s! (@ERR)          ;Read: For a word (@w), select (=s!) the error tag (@ERR),
        (0 N-DEF)       ;if the word itself is a noun in definite form (0 N-DEF), and
        (-2 GEN)        ;if the second word to the left is a genitive (-2 GEN), and
        (-1 A-DEF))     ;if the first word to the left is an adjective in definite form (-1 A-DEF).
```

The current description contains 659 @ERR rules. After all the @ERR rules have been tried, there is one final "rule" that picks @OK for all the remaining words. (No word has the feature DUMMY referred to in the rule.)

```
(@w =s! (@OK)               ;Read: For a word (@w), select (=s!) the @OK tag,
        (NOT 0 DUMMY))      ;if the word does not have the feature DUMMY.
```

What the actual CG components are used for in Grammatifix has been explained above. – To each @ERR rule is attached (a number that refers to) an error message. An error message consists of an error title, a short explanation, a correction scheme (when possible), and (behind a button) a longer explanation of the grammar point mentioned in the title. Below is given the error message, except for the longer explanation, attached to the @ERR rule presented above. Triggered by the above example sentence, the position slots (0) and (-2) in the explanation are filled by the words *stavningen* and *vilkas*, respectively. The correction means that the DEF form of the noun in position (0) is transformed into INDEF, so the correction suggested to the user is *stavning*.

Error title: *Substantivets bestämdhetsform*
Explanation: *Kontrollera ordformen (0). Om ett substantiv styrs av en genitiv, t.ex. (-2), bör det stå i obestämd form.*
Correction: *(0 N DEF) => (0 N INDEF)*

2.2. Overview of the error detection rule formalism

As noted, Grammatifix error detection (i.e. @ERR selection) rules use the CG rule formalism. For a full explication of the CG rule formalism see chapter 2 in Karlsson & al. (1995) – as a companion to the study of that chapter 2, below is given a convenient overview of the rule formalism as applied to @ERR selection. The example rule is already familiar (see section 2.1). After the overview follow some more examples of the ways in which the formalism can be used for error detection.

A Constraint Grammar error detection rule consists of four parts:

	Domain	**Operator**	**Target**	**Context condition(s)**
Example:	(@w	=s!	(@ERR)	(0 N-DEF) (-2 GEN) (-1 A-DEF))

Where:

Domain: @w (any word-form) or "<...>" (a specific word-form, e.g. "<ett>").
Operator: =s! (select) or =s0 (remove)
Target: @ERR or @OK.
Context condition: Polarity Position(Careful-mode) Set (Linked-position).

Polarity: Positive or negative (NOT). Examples:
(1 N) = the word in position 1 is N (i.e. has a N reading).
(NOT 1 N) = the word in position 1 is not N (i.e. does not have a N reading).

Position:

Target: 0.
Absolute: 1, 2, 3 etc., and -1, -2, -3 etc., in relation to the target. Examples:
(1 V) = the first word to the right from the target is V.
(-2 V) = the second word to the left from the target is V.
Unbounded: *1, *2, *3 etc., and *-1, *-2, *-3 etc., in relation to the target. Examples:
(*1 V) = a V one or more words rightwards from the target.
(*-2 V) = a V two or more words leftwards from the target.
Linked: R+1, R+2, R+3 etc. and *R, and L-1, L-2, L-3 etc. and *L, starting from a word found in some unbounded '*' position. Examples:
(*1 V R+1) (R+1 N) = somewhere to the right (*1) from the target is found a V, and the next word to the right (R+1) from that V is an N (R+1 N).
(*1 V L-1) (L-1 N L-1) (L-1 A) = somewhere to the right (*1) from the target is found a V, and the next word to the left (L-1) from that V is an N (L-1 N), and the next word to the left (L-1 again) from that N is an A (L-1 A). (Several linkings are possible.)
(*-1 AUX *R) (NOT *R INF) = somewhere to the left (*-1) from the target is found an AUX and to the right (*R) from that AUX there is no INF preceding the target (NOT *R INF).

Careful mode: A position may have C for 'careful mode', meaning that the condition is satisfied only in an unambiguous context. Example:
(1C N) = the word in position 1 has no other readings than N.

Set: Anything referred to in the context conditions must initially be declared as a set. Examples:

Set	**Set elements**
(GEN	GEN)
(N-NEU	(N NEU))
(A-DEF	(A DEF) (A DEF/INDEF))
(MOD-AUX	"kunna" ("vilja" V) ...)

Below are given four more illustrations of the error detection properties of the rule formalism. The rules here are simplified in the same sense as the @ERR rule in section 2.1, i.e. we ignore here the additional (sometimes highly specific) context conditions used for false alarm avoidance in the real rules. – The first rule below illustrates that the **domain** of a rule can be a specific **word form**, in this case "<ett>". The C as in 1C stands for **careful mode** (unambiguous analysis required), used in a majority of the @ERR rule context conditions.

Example: *Ett*(@ERR) *högtrycksrygg förskjuts norrut.*

Error detection rule (simplified):

```
("<ett>" =s! (@ERR)       ;Read: For the word-form Ett/ett, select (=s!) the error tag (@ERR),
        (1C N-UTR))  ;if the next word to the right is an unambiguous utrum noun (1C N-UTR).
```

The above rule uses a (maximally) close context. The following rules show that you can also refer to more comprehensive contexts. You may want to check e.g. that the whole sentence lacks some feature, e.g. the feature 'finite verb' as in the rule below. The rule illustrates **unbounded positions**, in this case *-1 (anywhere leftwards starting from position -1) and *1 (anywhere rightwards starting from position 1). **Negative** conditions (NOT) are often crucial.

Example: *Pulsen bli*(@ERR) *för kraftig.*

```
Error detection rule (simplified):
(@w =s! (@ERR)              ;Read: For a word (@w), select (=s!) the error tag (@ERR),
        (0C V-INF)          ;if the word itself is anunambiguous infinitive (0C V-INF), and
        (NOT *-1 V-FIN)     ;if there is no finite verb to the left (NOT *-1 V-FIN), and
        (NOT *1 V-FIN))     ;if there is no finite verb to the right (NOT *1 V-FIN).
```

A common situation is that you want to restrict a search to some specified portion of the sentence. This is illustrated by the last two examples. In the next example, the rule checks that there is no verb (especially, no instance of *ha*) between *skulle* and *skrivits*.[3] The rule illustrates the use of **linked conditions**, the link provided by an identical hook in the conditions, in this case by *R. Starting from a supine (*skrivits*) the first search here is for a modal auxiliary to the left and, when the first instance (*skulle*) is found, the second search starts for the non-occurrence of verbs between the modal auxiliary and the supine.

Example: *Så kom en flicka, som Göran höll av, in på Bibliotekshögskolan i Borås och hade hon inte gjort det skulle kanske denna artikel aldrig skrivits*(@ERR).

```
Error detection rule (simplified):
(@w =s! (@ERR)              ;Read: For a word (@w), select (=s!) the error tag (@ERR),
        (0C V-SUPINE)       ;if the word itself is un unambiguous supine (0C V-SUPINE), and
        (*-1 AUX-MOD *R)    ;if there is a modal auxiliary (AUX-MOD) to the left (*-1) and if to the right (*R) of it
        (NOT *R V))         ;there is no verb preceeding the word itself (NOT *R V).
```

In the last example, the word *inte* gets @ERR because of its placement after the finite verb in a subordinate clause.[4] The rule illustrates that you can **link several conditions**. The four conditions in the chain make three pairs of linked conditions, the first pair hooked together by *R, the second pair by R+1, and the third pair by *R. The sets in the rule are: ADV-CLAUSAL = clausal adverb, covering a number of common adverbs (e.g. *inte*, *aldrig*, *alltid*) used typically as "satsadverbial" (rather than as "särskilda satsadverbial"); SC = subordinating conjunction; NP-HEAD = nominal phrase head, e.g. N; V-FIN = finite verb; V = verb.

Example: *Söndagens lopp bevisade också att det spelar inte*(@ERR) *någon roll hur väl förberedd man är.*

```
Error detection rule (highly simplified):
(@w =s! (@ERR)              ;For a word (@w), select (s=!) the error tag (@ERR),
        (0 ADV-CLAUSAL)     ;if the word itself is an ADV-CLAUSAL (i.e. belongs to this set), and
        (*-1C SC *R)        ;if there is a SC to the left (*-1) and if to the right (*R) of the SC
        (*RC NP-HEAD R+1)   ;there is a NP-HEAD and the next word to the right (R+1) from the NP-HEAD
        (R+1C V-FIN *R)     ;is a V-FIN and if to the right (*R) of the V-FIN
        (NOT *R V)          ;there is no V preceeding the word itself (NOT *R V), and
        (-1C V-FIN))        ;if the next word to the left from the word itself is V-FIN (-1C V-FIN).
```

In more complex cases there are several chains of linked conditions, both leftwards and rightwards from the target, and there may also be any number of conditions on the words in absolute positions.

It is generally assumed, and it would seem to be an uncontroversial point, that grammar error detection has to be based on syntactic parsing, not necessarily of whole sentences but at least of those parts where errors are anticipated by the system – for instance, in order to be able to find noun phrase internal errors, the system would first have to parse noun phrases. (For systems that build on

some measure of syntactic parsing see e.g. Sågvall Hein 1998, Knutsson 1998, Cooper 1998, Cornu & al. 1996, Bustamante & León 1996.) It is therefore a noteworthy feature of the Grammatifix error detection system that it does **not** build on the output of a syntactic parser. What takes the place of syntactic parsing as such are the context conditions in the @ERR selection rules; in a way, each rule does its own syntactic analysis of a specific sequence of elements and, typically, of the context where it occurs. This is perhaps a burdensome way of writing error detection rules, but, on the other hand, we are saved the trouble of working on parsing rules and their relaxations (cf. Bustamante & León 1996). Anyhow, the conditions for the avoidance of false alarms often being pattern-specific (cf. chapter 3), it is convenient to have pattern-specific error detection rules where to incorporate such conditions. The more local a phenomenon is, the easier it is to control with CG rules.

2.3. Grammar errors and disambiguation

The relationship between disambiguation and grammar error detection is intricate. On the one hand, it is obvious that disambiguation is a prerequisite for any effort at precise error detection. On the other hand, a grammar error may disturb the disambiguation, with either a disambiguation error or remaining ambiguity as a consequence, and this in turn may disturb the error detection.

This section illustrates the methods we use in Grammatifix in order to overcome effects of disambiguation disturbances caused by grammar errors. You can either take the disambiguator's disturbed output as it is and write error detection rules based on that (methods 1 and 2 below), or you can make changes in the disambiguation rules as used by the error detector (method 3). Considering the kinds of Grammatifix rules involved we can speak of three methods: (1) word-form-specific @ERR rules, (2) @ERR rules for ambiguous words, and (3) adjustment of the disambiguation rules. The methods are illustrated in turn below.

Word-form-specific @ERR rules. In the following example (used earlier in chapter 2.2), the correct analysis of the word *ett* would be DET, so in the Grammatifix analysis shown below we have a disambiguation error: PRON instead of the intended DET.

Example: *Ett*(@ERR) *högtrycksrygg förskjuts norrut.*

```
Lexical analysis of ett:
"ett"  <NUM/ART> <ID> DET NEU INDEF SG NOM      ;Correct analysis in modifier use.
"ett"  <NUM> PRON NEU INDEF SG NOM              ;Correct analysis in head use.

Grammatifix analysis:
"<ett>"
    "*ett" <**c> <NUM> PRON NEU INDEF SG NOM @ERR
"<högtrycksrygg>"
    "högtrycksrygg" N UTR INDEF SG NOM @OK
```

The grammar error in the above example is – using the grammatically proper terms to describe it – that a neuter **determiner** (DET) is combined with an utrum noun. In the @ERR rule we can not describe the error in those terms, however, because the disambiguator discards the DET reading. What we have here instead is an @ERR rule with the word-form domain "<ett>" (for the rule formulation see chapter 2.2). Grammatifix tolerates the disambiguation error (PRON) simply by ignoring it. Word-form-specific rules are used especially with many common determiners.

Word-form-specific rule can also be formulated so as to cover a set of word forms. In this case the domain of the rule is @w, and the set is used in e.g. the target position (0). For example, in the formulation (@w =s! (@ERR) (0 POSS-UTR) ...), the set POSS-UTR covers utrum forms of possessive determiners, e.g. *sin*. In the sentence *Han har sin*(@ERR) *företag att tänka på*, the disambiguator leaves *sin* three-way ambiguous (DET|PRON|ABBReviation). The ambiguity does not disturb the error detection because the @ERR rule refers to the set POSS-UTR, i.e. ultimately to the word form *sin* in this case.

@ERR rules for ambiguous words. When developing the @ERR rules we noticed cases where words remained ambiguous in some systematic way in certain targeted patterns, and rules were then formulated so as to accept the ambiguity. The following example illustrates the idea. When combined with certain verbs (e.g. *uppges* below) and not preceeded by *ha*, the word (*orsakat* below) whose correct analysis would be supine remains ambiguous between supine and past participle (<PCP2>, with A as part-of-speech tag). (For the message cf. note 3.)

Example: *Slarv uppges orsakat*(@ERR) *branden*

Grammatifix analysis:
"<slarv>"
    "*slarv" <**c> N NEU INDEF SG NOM @OK
"<uppges>"
    "uppge" V PASS PRES @OK
"<orsakat>"
    "orsaka" V ACT **SUPINE** @ERR
    "orsaka" **<PCP2>** **A** NEU INDEF SG NOM @ERR
"<branden>"
    "brand" N UTR DEF SG NOM @OK

What we in this case want to refer to in the @ERR rule is precisely the ambiguity of the target word. This is done with the description (@w =s! (@ERR) (0C SUPINE/PCP2) (0 SUPINE) ...), where the sets are (SUPINE/PCP2 SUPINE <PCP2>) and (SUPINE SUPINE). According to the condition (0C SUPINE/PCP2) the target word must have a SUPINE reading or a <PCP2> reading or both, and according to the condition (0 SUPINE) it must have at least a SUPINE reading. This excludes words that are disambiguated unambiguously as <PCP2>, e.g. premodifiers of nouns. The above target description accepts also words that are unambiguously SUPINE (e.g. *skrivit*).

Adjustment of the disambiguation rules. It was noted in section 2.1 that the disambiguation rules of SWECG have been adopted to a large extent as such in Grammatifix, but, importantly, there are differences. (At present there are some 50 points of difference; if it was not for the two methods discussed above, there would have to be many more.) The most important ones of the differences involve the following scenario. Using the original SWECG disambiguation rules we noticed that certain common open-class words, lexically ambiguous in some systematic way, regularily lost their intended reading in a certain error pattern where the intended reading would have been needed for the @ERR rule to apply. Disambiguation rules were then adjusted in Grammatifix for the recovery of the needed reading. An illustration follows.

It was noticed that common indefinite adjective forms (e.g. *kall* "cold") with a competing indefinite noun reading (*kall* "vocation") regularily lost their adjective reading in the error pattern GEN + A-INDEF(@ERR) + N-INDEF. In the @ERR rule for this pattern, the target word is described as (@w =s! (@ERR) (0C A-INDEF) ...). The rule detected the error e.g. in *Hennes vacker*(@ERR) *hand*, where vacker is A-INDEF, but in the following example the rule did not detect the error because the disambiguator selected the N reading of *kall* instead of the A reading presupposed by the rule.

Example: *Hennes kall hand*

Original SWECG disambiguation:
"<*hennes>"
    "hon" <**c> <PERS-SG3> DET UTR DEF SG GEN
"<kall>"
    "kall" N NEU INDEF SG/PL NOM
"<hand>"
    "hand" N UTR INDEF SG NOM

In SWECG the N analysis above is understandable because *Hennes* + N-INDEF does, whereas *Hennes* + A-INDEF does not, make grammatical sense. In Grammatifix, in order to detect the error, we did the following. First, we defined a set, KALL-ETC, covering words that are ambiguous between A and N in the same way as *kall*, e.g. *besk*, *briljant*, *kall*, *intern*, *sval*, all in all some 60 common adjectives. Then (with due attention to additional details), we wrote a rule specifically for the disambiguation of the KALL-ETC words as A in the context (-1C GEN) (1C N-INDEF). The same @ERR rule that detected the error in *Hennes vacker hand*, now detected the error also in *Hennes kall hand*, as shown below.

Grammatifix analysis:
"<hennes>"
    "*hon" <**c> <PERS-SG3> DET UTR DEF SG GEN @OK
"<kall>"
    "kall" A UTR INDEF SG NOM **@ERR**
"<hand>"
    "hand" N UTR INDEF SG NOM @OK

Using the methods illustrated above we have come to grips with a number of disambiguation disturbances caused by grammar errors, but we are also aware that there are many cases that we have not tackled yet. Some amount of disambiguation errors, be they due to grammar errors or other factors, will always remain a feature of the output of a computational analyser.

3. Notes on the process of refining the error detection rules

The challenge for an error detector is not only to detect errors but also to avoid false alarms. This chapter describes work done in Grammatifix on the avoidance of false alarms.

The issue can be introduced by way of a two-point example. (1) You want to detect the error in the phrase *vilkas diskontinuerliga stavningen*(@ERR), so you write the rule (@w =s! (@ERR) (0 N-DEF) (-2 GEN) (-1 A-DEF)), presented in chapter 2.1. (2) You become aware of cases where you want to **avoid false alarms**, e.g. *Strindbergs Röda rummet*(@OK) and *Dostojevskijs berömda Idioten*(@OK), so you **add conditions** to the @ERR rule to the effect that it does not apply in these cases. The two simple conditions added to the above rule due to these two cases are (NOT -1 CAP) and (NOT 0 CAP), respectively. The set CAP refers to words written with an initial capital letter, e.g. *Röda* and *Idioten*. It is not feasible to list all proper names (e.g. titles of literary works).

The two points in the above introduction correspond to two stages in the process of developing the @ERR rules: (1) constructing a set of basic, rather unconstrained rules, and, based on corpus testing, (2) refining the basic rules. The basic rules were "rather unconstrained" in the sense that they might flag (almost) all instances of a potential error pattern – say, all instances of a noun in definite form preceeded by a genitive, or all instances of *detta* in front of an utrum noun. It was obvious even prior to testing that, of those instances, quite a number would be false alarms. False alarm cases are often so marginal structurally that they easily escape the rule writer's intuitive attention. The purpose of corpus testing was to bring false alarm cases more effectively into our attention, the task then being to refine the rules so that false alarms be eliminated.

The main corpus we used for the purpose explained above was a 1.6 million word collection of published texts, mainly from newspapers and periodicals. (The corpus was compiled by Fredrik Westerlund.) The testing procedure was as follows: the corpus was divided into five parts, and with each part we ran through the same three steps as described below for part 1, except that after part 5 there was no "next part of the corpus" to apply the refined rules to.

The basic rule set applied to part 1 of the corpus
- Each @ERR alarm studied: good or false?
- **False alarms eliminated as far as reasonable**
- Result: a refined rule set, applied to the next part of the corpus

The main point here is that we eliminated false alarms, for examples see below. The reason for treating the corpus in parts was that we wanted to verify that the precision of the rules (ratio of good alarms to all alarms) was improving after each round of rule refinements. It may be noted that when the basic rule set, prior to any refinements, was applied to part 1 of the corpus, the precision was 38%. The precision of the current rule set is reported on in chapter 4.

Below are given some examples (1–9) where a Swedish error detector, if not careful enough in assuming NP internal error patterns, might be tempted into giving false alarms. As for Grammatifix, this is a small sample of cases where the unconstrained rules initially gave a false @ERR alarm but where the refined rules now give @OK (no false alarm). Some notes follow. What these examples signify is an atomistic kind of work process, i.e. cases to be taken into account individually.

(1) *Sveriges* ***Televisions Antikrundan***(@OK) *har slagit ...*
(2) *... Carlos Menem beordrade i* ***fredags flottan***(@OK) *att avföra ...*
(3) *... till* ***dess barnet***(@OK) *fyller 18 år.*
(4) *... har ett* ***slags konstnärlig***(@OK) *frihet att ...*
(5) *... som inte tyckte om sin före* ***detta***(@OK) ***flickväns*** *nye man.*
(6) *... presenterat* ***en***(@OK) ***handfull***(@OK) ***program*** *med samma ...*
(7) ***Obetald***(@OK) ***omslags-*** *eller sällskapsflicka ...*
(8) *Walters far är gammelkommunisten för* ***vilken demokratin***(@OK) *börjar utanför dörren.*
(9) *Från början var* ***det stora problem***(@OK) *att få i Johan tillräckligt med mat.*

Notes on (1–9):

- The potential false alarm sources in (1–9) are: a noun in definite form is preceeded by a a word in genitive form (1–3); an adjective in indefinite form is preceeded by a word in genitive form (4); an utrum noun is preceeded by *detta* (5); a neuter noun is preceeded by *en* and an utrum adjective (*handfull*) (6); a neuter noun is preceeded by an adjective in utrum form (7); a noun in definite form is preceeded by *vilken* (8); a noun in indefinite form is preceeded by *det* and a potential definite form of an adjective (9).
- Example (2). The best way to avoid a false alarm in (2) is probably to treat *i freadgs* as a fixed syntagm, i.e. *"i_fredags"* ADV, cf. the notes on preprocessing in chapter 2.1.
- Example (3). This example can be used to illustrate a kind of chain reaction that may occur as a consequence of adding a condition to a rule. First, a general rule for GEN + N-DEF(@ERR) detects the error e.g. in *Onsdagens finalen*(@ERR) *visas i TV.* In that general rule we have added the condition (NOT -1 DESS) in order to avoid a false alarm in (3). Then, in order to detect the error e.g. in *Dess framtiden*(@ERR) *är osäker*, we have written a rule specifically for *dess* + N-DEF(@ERR), and (only) in that rule we use the condition (NOT -2 TILL), again in order to avoid a false alarm in (3). This kind of chains may sometimes be the only way of achieving the desired pattern- or word-specific effects.
- Example (6). The word *handfull*, classified as an adjective in SWETWOL, was excluded from the adjective slot in the relevant @ERR rules for the avoidance of false alarms in cases like (6) (quite frequent in texts). A syntactically more perceptive solution would have been to provide *handfull* with a noun reading (cf. *ett antal program*).
- Example (8). In (8) and (9), more than in the previous examples, the false alarm conditions are clause-structurally oriented. (8) can be compared with the following sentence where Grammatifix properly detects the error: *Saab var riktmärket för* ***vilken bilmodellen***(@ERR) *var och en skulle ha.* In this sentence the sequence *vilken* + N-DEF is followed by a NP boundary (*var och en*), whereas this is not the case in (8), this distinction taken into account in the rule conditions.
- Example (9). The single most problematic word for Swedish (agreement) error detection cum false alarm avoidance is *det*. This is because of the many uses of *det* as an independent clausal element particularily in a position after the finite verb. Let us consider one of the potential error patterns, viz. the one exemplified by *det stora problem* in (8) (*det* + A-DEF/INDEF-SG/PL + N-NEU-INDEF-SG/PL). The relatively **safest** clausal position for assuming an error, i.e. the position with the least chances for false alarms, is the initial pre-finite-verb position (= PRE-FV), e.g. *Det stora*

problem(@ERR) *har lösts*. The **next safest** position for assuming an error is the post-non-finite-verb position (= POST-NONFV), e.g. *Hon har löst det stora problem*(@ERR). More conditions are pertinent in POST-NONFV than in PRE-FV. You need to check that the non-finite verb is not ditransitive, cf. *Ni har vållat det stora problem*(@OK), and also that no relative clause follows, cf. *löst det stora problem*(@OK) *som ni funderat på*. The **least safe** position for assuming an error is the post-finite-verb position (= POST-FV). In addition to the possibilities in POST-NONFV, in POST-FV you have to consider that *det* may function as subject, e.g. *Här framkallar det stora problem*(@OK), or as formal subject, e.g. (8) and *Här finns det stora problem*(@ERR). Because such POST-FV uses of *det* are so common, and in any case much more frequent than erroneous uses, it would seem to be motivated to prevent the @ERR rules here concerned from applying in POST-FV. However, not all POST-FV contexts are equal. For instance, it is relatively safe to assume an error if the finite verb is an auxiliary and a non-finite verb follows, e.g. *Nu har det stora problem*(@ERR) *lösts*. This is more or less as far as we have come with the description of the *det* pattern here discussed. – A clause construction where Grammatifix at present misses an error is exemplified by *Hennes assistanter löste det stora problem*(@OK, missed error). Crucial factors here are that the clause-initial constituent is a (non-adverbial) noun phrase (*Hennes assistanter*), and that the finite verb is monotransitive (*löste*). On the basis of that information it would be motivated to flag *problem* as @ERR in the above example, but, as noted, this has not yet been worked into the Grammatifix rule set.

Users soon get tired of a language checker that makes a lot of false alarms. In a practical grammar checker it is therefore motivated to make false alarm avoidance a priority even at the expense of errors being missed – but there is a limit, of course. It was noted above that, in the corpus used for rule refinement purposes, we tried to eliminate false alarms "as far as reasonable". This eludes exact definition, but the flexible idea is that we do not insist on conditions for false alarm avoidance if they would unduly compromise the rule's error detection power in unintended contexts. To illustrate, below are given two examples of a problematic construction, ellipsis of the verb gapping type. Grammatifix makes here false alarms: it believes that *de andra medlemmar* is a phrase with a definiteness form error, and that *den andra hormonet* is a phrase with a gender agreement error.

Nitton av dem ska ha varit medlemmar av Umma-partiet och de andra medlemmar(@ERR, false alarm) *av det förbjudna arabsocialistiska Baathpartiet.*
Ena kammaren innehåller destillerat vatten, och den(@ERR, false alarm) *andra hormonet.*

It would be possible to eliminate many of the false alarms due to ellipsis, e.g. by using conditions that refer to a coordinator or a comma in the left context. However, such conditions would not be precise enough – they would prevent the rules from making valid detections in many contexts that have nothing to do with ellipsis. False alarms due to ellipsis are not reasonably to be avoided, the ultimate reason being that ellipsis is too elusive for us to identify exclusively.

A grammar checker is a compromise between error detection and false alarm avoidance. One way of describing such a compromise is to provide test results on precision and recall (see the final chapter). In the end, the user is the arbiter of whether the compromise is acceptable or not.

4. Performance tests

Introduction. For this presentation, we tested our set of grammar error detection rules (i.e. @ERR selection rules) for overall precision and recall with texts new to the system. Precision and recall can here be defined as follows (cf. Bernth 1997:159, Paggio & Music 1999:278).

Precision: the ratio 'good alarms / all alarms'
Recall: the ratio 'detected errors / all errors'

Precision is a measure of how good the checker is at avoiding false (unintended, irrelevant) alarms, and recall is a measure of how good the checker is at identifying the errors in a text – the higher the recall and the precision, the better. As the term 'error' is used here it covers, in addition to undisputable grammar violations (e.g. the verb chain *kan* + *blir* in *Då kan bland annat så kallade utbildningskonton blir aktuella.*), also constructions targeted by Grammatifix which, acceptable to some, are not regarded as impeccable by everybody (e.g. the verb chain *kommer* + *sätta* in *De kommer sätta stenhårt tryck på oss.*). We are here concerned with grammatical errors, so nothing will be said about e.g. spelling errors (the concern of a spell-checker) and writing convention errors (the concern of a separate set of Grammatifix rules, cf. chapter 1).

There is no standard for how the performance of a (Swedish) grammar error detector should be evaluated. One issue here is the kind of test data used. Research groups often seem to use their error corpuses, i.e. collections of sentences containing errors of the types the system is concerned with (cf. Domeij & Knutsson 1999, Paggio & Music 1998, Cornu & al. 1996, Bolioli & al. 1992). In the tests here reported we used running newspaper text. Results based on such data are not comparable to results based on a collection of errors. Especially, running newspaper text, with a high proportion of grammatically correct sentences, puts precision to a hard test. – A fundamental kind of problem is also that there are no agreed-upon criteria for what should count as a grammar error or as a good alarm (in border-line cases). A further factor that would complicate comparative evaluation is the variation in the error types targeted by different systems, e.g. where one system tries to detect only easiest-to-identify errors while another system tries to detect also more-difficult-to-identify errors.

In anticipation of more carefully planned and documented test schemes – schemes that would address open issues such as the ones noted above – we present below our precision and recall tests.

Precision. The test data is a 1,000,504 word extract from the Swedish newspaper *Göteborgs-Posten* 1998. Of the grammatical categories that Grammatifix has checks on (presented in Arppe & al. 1999, listed also in Arppe in this volume) two were excluded from the test. These two are 'no verb' (e.g. *Ungefär som en kansler.)* and 'no finite verb' (e.g. *Göra independentfilm till exempel.*). Grammatifix points out these properties of sentences, but in almost all cases no error is involved. These properties are rather frequent and easy to detect reliably.[5]

The test data was analysed by the @ERR selection rules (in Unix); each alarm was studied as to whether it was good or false; the numbers of good and false alarms were used for calculating the precision rate. The result is given below, both as a percentage and as absolute figures.

Precision of Grammatifix in a 1,000,504 word extract from *Göteborgs-Posten* 1998:

Good alarms	False alarms	**Precision**
374	160	**70%** (374 / 374 + 160)

Is 70% a good or a bad overall result? It is hard to say as we have not found similar test reports to compare with. What the result would be with other types of text remains an open question, too.

Perhaps the most relevant point to make concerning the 374 good alarms is, simply, that simple grammar errors do occur even in published texts produced by native writers. We may illustrate with some noun phrase internal agreement errors (1–9 below) detected by Grammatifix in the test data.[6] These are typical agreement violations in the sense that each of them involves only one agreement feature, gender in (1–3), number in (4–6), and definiteness in (7–9).

(1) ... *som beskriver världen utifrån en*(@ERR) *annan*(@ERR) *paradigm än till exempel Newton och Descartes.*
(2) ... *blev i stället en stort*(@ERR) *besvikelse för Pernilla Wiberg.*
(3) *Det är ju utlänningar i nästan varenda*(@ERR) *bås nuförtiden.*
(4) *Men på de*(@ERR) *mest framskjutna platsen i montrarna står det mer rustika porslinet ...*
(5) *Mir har under det*(@ERR) *senaste åren drabbats av flera svåra olyckor.*
(6) *Polisen gjorde färre fordonskontroll*(@ERR) *förra året ...*
(7) *I går häktades den 32-årige stockholmare*(@ERR) *vid Stockholms tingsrätt, misstänkt för grovt häleri.*
(8) ... *inte hade lyckats uppnå samma ekonomisk*(@ERR) *utveckling.*
(9) ... *utan att någondera familj*(@ERR) *förstod varför katten blev allt fetare.*

The general-level sources for the 160 false alarms in the test data are: lexical gaps or errors (46 out of 160); disambiguation errors (18); not accurate enough @ERR selection rules (96). Some of these false alarms will be easy to eliminate, the easiest ones being among those due to a lexical source, e.g. the word *partnerskap*, treated as SG in the current SWETWOL (and in SAOL), could be tagged as SG/PL for the elimination of the false number agreement alarm in *Humankapitalet skyddas lämpligen genom nya*(@ERR) *partnerskap vid företagande (...)*. On the other hand, some of the false alarms are such that we do not consider it reasonable even to try to avoid them, e.g. the ellipsis-induced (cf. chapter 3) false number agreeement alarm in *Antal toaletter: tre, varav två tjej(@ERR) och en kill.*

Recall. The test data is a 87,713 word extract from *Göteborgs-Posten* 1998. Two linguists (the present author and Eva Orava, also at Lingsoft) read the extract, marked all the grammar errors they found, and discussed problem cases. What they ended up with was 135 grammar errors distributed over different categories as follows: agreement errors (31), most of them NP-internal (22), the rest (9) involving complements, postmodifiers, and anaphoric pronouns; verb form compatibility errors (28), especially violations of verb chain internal constraints; preposition errors (26); missing or superfluous endings (21), e.g. genitive, passive, or adverb endings; compounds written as separate words (8); sentence structure errors (8); word order errors (3); others (10).

Of the 135 grammar errors found by the linguists in the test data, 55 belong to the categories targeted by Grammatifix[7]. Any computational grammar checker is, of course, only a partial grammar checker; no current systems have anything like comprehensive checking of, say, sentence structure, anaphoric pronoun agreement, missing endings, and even preposition use (in other than some types of fixed phrases perhaps). Now, is it more to the point to calculate recall in relation to 'all errors in all the error categories', or in relation to 'all errors in the error categories targeted by the system'? The results of both calculations are given below. Recall in relation to the targeted categories is an overall measure of how well the rule set does what it tries to do.

Recall of Grammatifix in a 87,713 word extract from *Göteborgs-Posten* 1998:

	All errors in text	Detected errors	**Recall**
Targetted error categories:	55	47	**85%** (47/55)
All error categories:	135	47	35 % (47/135)

The general-level sources for missed errors in the targeted categories (in the test data, 55-47 = 8) are the same as those for false alarms, i.e. lexicon, disambiguation, and @ERR rules (cf. above). 'Not accurate enough' @ERR rules means here that errors are missed due to overly prohibitive conditions for the avoidance of false alarms (in the test data, 5 of the 8 misses). One of the problems associated with the recall test here reported is the small size of the test data.

Notes

[1] For spell-checking Lingsoft has a separate program (Orthografix).

[2] The part-of-speech tags referred to in this paper are: A = adjective, ADV = adverb, DET = determiner, N = noun, PRON = pronoun, SC = subordinating conjunction, V = verb. Nominal minor feature tags include: DEF = definite, GEN = genitive, INDEF = indefinite, NOM = nominative, NEU = neutrum, PL = plural, SG = singular, UTR = utrum. Other tags and set names, if not transparent, will be explained when referred to.

[3] The construction 'modal auxiliary + supine without *ha*' (e.g. *skulle ... skrivits*) is not regarded by everybody as recommendable in polished style. The Grammatifix message is that, in polished style, a modal auxiliary is combined rather with *ha* + supine than with supine alone. Cf. Wellander (1973:139).

[4] There are differences between subordinate clauses as to the usability of the word order 'finite verb + clausal adverb' (Teleman & al. 1999:537-9). In formal written Swedish, however, the order 'clausal adverb + finite verb' can be regarded as recommendable in all types of subordinate clauses. Cf. Reuter (1996:8), Åberg (1995:34), *Dagens Nyheters Skrivregler* (1997:30).

[5] In the precision test data, the rules made 7145 'no verb' alarms, and 321 'no finite verb' alarms. The latter ones were studied more in detail: of the 321 alarms, 312 were good, i.e. cases where the sentence included a non-finite verb but

no finite verb. The precision for this alarm type was 97% (312/321). A few real errors detected by the rules (e.g. *Pulsen bli för kraftig*) were ignored in the precision test as a consequence of excluding the alarm types.

[6] Of the 374 good alarms, 134 were concerned with noun phrase internal agreement. The other big group was verb form compatibility with 176 good alarms; 99 of these were supines without *ha*, e.g. *kunde den fått*(@ERR) (cf. note 3).

[7] Among these 55, the two big groups were noun phrase internal agreement violations (22) and verb chain internal compatibility violations (21).

References

Arppe, Antti (this volume). Developing a grammar checker for Swedish.

Arppe, Antti, Juhani Birn, and Fredrik Westerlund 1999. Lingsoft's Swedish Grammar Checker. http://www.lingsoft.fi/doc/swegc/.

Bernth, Arendse 1997. Easy English: A Tool for Improving Document Quality. *Proceedings of the Fifth Conference on Applied Language Processing*, Washington, 159–165.

Bolioli, Andrea, Luca Dini, and Giovanni Malnati 1992. JDII: Parsing Italian with a Robust Constraint Grammar. *Proceedings of the 15th International Conference on Computational Linguistics*, Nantes, 1003–1007.

Bustamante, Flora Ramiréz and Fernando Sánchez León 1996. GramCheck: A Grammar and Style Checker. *Proceedings of the 16th International Conference on Computational Linguistics*, Copenhagen, 175–181.

Birn, Juhani 1998. Swedish Constraint Grammar: A Short Presentation. http://www.lingsoft.fi/doc/swecg/.

Cooper, Robin 1998. Finite state grammar for finding grammatical errors in Swedish text. http://www.ling.gu.se/~sylvana/FSG/.

Cornu, Etienne, Natalie Kubler, Franck Bodmer, Francois Grosjean, Lysiane Grosjean, Nicolas Lévy, Cornelia Tschichold, and Corinne Tschumi 1996. Prototype of a second language writing tool for French speakers writing in English. *Natural Language Engineering* 2, 211–238. Cambridge University Press.

Domeij, Rickard and Ola Knutsson 1999. Granska - ett effektivt hybridsystem för svensk grammatikkontroll. http://www.nada.kth.se/theory/projects/granska/rapporter/nodalidaabstrakt.html.

Karlsson, Fred, Atro Voutilainen, Juha Heikkilä, and Arto Anttila (eds.) 1995. *Constraint Grammar. A Language-Independent System for Parsing Unrestricted Text.* Berlin and New York: Mouton de Gruyter.

Knutsson, Ola 1999. Granskas regelspråk. At: http://www.nada.kth.se/theory/projects/granska/.

Paggio, Patrizia and Bradley Music 1998. Evaluation in the Scarrie project. *Proceedings of the First International Conference on Language Resources & Evaluation*, Granada, 277–282.

Reuter, Mikael 1996. *Reuters rutor 2.* Esbo: Schildts.

SAOL, *Svenska Akademiens ordlista över svenska språket.* Tolfte upplagan. 1998. Norstedts.

Sågvall Hein, Anna 1998. A Chart-Based Framework for Grammar Checking. Initial Studies. *Proceedings of the 11th Nordic Conference on Computational Linguistics*, Copenhagen, 68-80.

Teleman, Ulf, Staffan Hellberg, and Erik Andersson 1999. *Svenska Akademiens grammatik 4.* Norstedts.

Wellander, Erik 1973. *Riktig svenska.* Stockholm: Esselte studium.

Åberg, Gösta 1995. *Hur ska det heta? Tidens lilla språkriktighetslexikon.* Stockholm: Tidens förlag.