# Evaluating second language grammar checkers ◊

**Cornelia Tschichold**

## Abstract

Different aspects of an evaluation of grammar checkers for non-native speakers are presented. They include extra-linguistic topics such as user-friendliness as well as various linguistic aspects. The importance of using "authentic" (i.e. second language) text material is emphasized. The point of view of a potential user is proposed as a strategy for designing and selecting appropriate test material to be used in an evaluation kit.

## 1. Introduction

This article is a short summary of the "Lizentiatsarbeit" I wrote in 1991. Its topic was the evaluation of bilingual grammar checkers and comprised a ready-made kit that could be used by potential users to evaluate and compare grammar checkers. A black-box approach to evaluation was used, i.e. an evaluation where the internal functioning of the tool is not examined; only performance observable from a user's point of view is tested.

Grammar and style checkers have been developed as a step forward from spelling checkers now available for most word-processing software. They claim that they can detect the most frequent grammatical and stylistic errors made by native speakers. In particular, bilingual grammar checkers are supposed to find typical interference errors made by non-native speakers. Over the last few years, several such bilingual checkers, mostly for English texts, have appeared on the market. They are relatively inexpensive programs and are therefore only rarely evaluated in more than informal articles in computer magazines.

Evaluation of software tools can be defined as the comparison of the actual behavior of the evaluated tool with the requirements of the potential users (Guida & Mauri, 1986). However, evaluation of NLP (natural

---

◊ This research was supported by a grant from the Swiss CERS/KWF (2054.2).

language processing) systems is hampered by the facts that, on the one hand, there are no formal criteria for measuring the deviations between the requirements and the performance, and, on the other hand, only samples can be used for the testing procedure due to the nature of natural language. There is, however, no general agreement about the establishment of such sample sets (Palmer & Finin, 1990). Another problem is simply the lack of a formal standard relating to the requirements for such software programs.

Evaluation of grammar checkers is further complicated by the basic conflict inherent in grammar checking formalisms. Parsing relies on correct structures and therefore violations of these grammatical structures would normally lead to a failure of the overall parse (Thurmair, 1990). But this does not necessarily help the user find the error in the sentence. Nevertheless, I believe that it is possible to arrive at a meaningful statement on the usefulness of grammar checkers if the point of view of a potential user is adopted and if as many as possible of the relevant aspects are tested and compared to the user's needs, even if relatively informal criteria are used. Such an evaluation will not be independently valid, but can serve as a comparison of different tools.

An evaluation should judge not only the purely linguistic capacities of a grammar checker to detect and correct errors in a text but also the usefulness of such a product on a wider scale. Consequently, non-linguistic considerations such as compatibility and user-friendliness become more relevant. Linguistic aspects should include not only the actual correcting capacity of the tool but also on-line references that can help the user with specific grammatical points or vocabulary problems. Such on-line references can be very useful during the writing process and during correction as today's grammar checkers are still far from being perfect. Finally, an evaluation procedure for such a relatively inexpensive program should be simple and not too time-consuming to perform.

## 2. Extra-linguistic aspects

One of the primary aspects of a software program for potential users is compatibility with their own equipment and software. In the case of grammar checkers, this mainly concerns the operating system and word



processor already in use. A grammar checker that is not compatible with the user's word processing software is obviously at a strong disadvantage compared to a program that works with (or within) his or her usual word processor and is able to cope with and preserve the formatting information already there.

The other non-linguistic aspect is user-friendliness in a broad sense. This is an important point as a grammar checker is a product intended for a fairly general group of users. The program should be easy to install and use, and be robust against unscheduled user inputs. Speed, visual appearance, and the quality of the editing facilities should also be assessed. This can be done by using simple scales to evaluate personal impression. Finally, the messages that the grammar checker gives when an error is found should preferably be in the user's first language and also be meaningful to those who have neither a linguistic nor a computer science background. Messages should be polite and provide enough information about the error to allow the user to understand the nature of the error and then to take a decision on its correction.

## 3. Linguistic aspects

The linguistic aspects of a grammar checker can be divided into the actual error detection and correction facilities and the on-line reference options offered, such as dictionaries and grammars. Such on-line language tools are intended to help the user during the writing process, but should also be readily available during correction. They can be evaluated for content, for completeness (as compared to paper dictionaries and grammars), for easy accessibility and input error tolerance, for updating options and for general user-friendliness, including visual appearance and speed. A user-friendly look-up function for a thesaurus, for example, would include automatic morphological changes, e.g. a verb in the past tense could be replaced with the past tense of another verb if the user chooses to replace that verb with a synonym found in the thesaurus. Similarly, the correct article "a" or "an" would be automatically chosen before the following noun or its adjective.

Once the text has been written, the core of the grammar checker comes into action. For a bilingual grammar checker, there are more

linguistic aspects to take into account than for a monolingual checker. Users make more errors, and more varied ones, when they write a text in their second language. Some of these errors are influenced by the structure of their native language. Ideally a bilingual tool should therefore be able to deal with the many types of errors that occur in written text. These include:

| | |
|---|---|
| - punctuation | * The increase was 4,8 %. |
| - spelling | * adress |
| - morphology | * Reactions will be more strong. |
| - syntax | * I would have never dreamt of this. |
| - lexicon | * Some afraid people ran away. |
| - style | ? You can say good-bye to your business. |

Within each of these error categories, the user's native language should be taken into account both in detection and in the proposed correction(s). Above all, a bilingual grammar checker should be capable of detecting at least those errors that are most frequent for a particular pair of languages. The fact that a grammar checker can correct an error the user is not ever likely to make may be impressive, but it is not very helpful to users. This and the fact that users will have varying levels of knowledge of English and differing preferences concerning points of grammar are reason enough for the program to have an option that lets the user turn off those checks not needed. In addition, turning off certain checks can be used as a less than ideal solution to the problem of overflagging.

In a black-box evaluation, linguistic testing is carried out using texts containing various errors that are run through the grammar checker. As we are dealing with natural language, it is obviously impossible to test every error in every possible context; we therefore have to work with samples of texts. The problem that arises here is the selection of test sentences. They should include different classes of errors but also similar errors in varying contexts. Furthermore, the chosen texts should correspond to the type of evaluation being done. The users' level of English, their native language, and the style they prefer to use should be taken into account. Therefore, one of the best solutions seems to be to



choose "real" texts containing typical errors. If required, these can be complemented by constructed sentences to test specific features.

Using authentic texts in an evaluation of bilingual grammar checkers has several advantages. Very often there is more than one error in a single sentence, and these can influence each other to a point where the grammar checker is at a total loss. Furthermore, the problem of taking into account the exact frequency of certain error types can be bypassed to a certain extent with this approach. Such a procedure also helps to pinpoint the particular weaknesses of these tools, e.g. their inability to deal with lexical choice, which is a particularly important point for non-native writers.

An evaluation of the detection process should include the number and type of wrong detections. The importance of this "overflagging" should not be underestimated. Non-natives are naturally less sure of their language skills and therefore more prone to be thrown into doubt by an unclear or wrong message. These superfluous messages are both confusing and make the grammar checking process much more time-consuming for the user. An adequate number of penalty points should therefore be given for every wrong message. In the 50-page kit compiled for my "Lizentiatsarbeit", the following six-part classification for error messages is proposed.

| | **Flag** | | **No flag** |
|---|---|---|---|
| | Detection (DF) | Warning (WF) | (nF) |
| **Error** | Case 1 | Case 2 | Case 3 |
| **No error** | Case 4 | Case 5 | Case 6 |

The first three cases are applicable if there is an error in the text.

**Case 1** represents the ideal situation: An error is detected and the message adequately describes the problem and proposes an acceptable solution.

**Case 2** occurs when the error is found but the message includes restrictions, e.g., "If this word is a noun, then...". Such a warning message should enable the user to solve the problem.

**Case 3** applies when the grammar checker misses an error in the text.

Grammar checkers also issue error messages and warnings if there is no error in the text (overflagging). In this context, the following three cases can be distinguished:

**Case 4** represents the worst case: The grammar checker detects an error where there is none.

**Case 5** is slightly less disastrous than the above because the message only gives a warning that should enable the user to rule out the error described in the message.

**Case 6** is applicable when there is no error in the text and no flag is produced.

This classification is quite simple and computational linguists and tool designers may want to use finer distinctions. However, it seems easy enough to handle for inexperienced users. As some grammar and style checkers never issue detection flags but simply give warnings and statements of caution, cases 1 and 4 could be omitted or used only for spelling errors.

Here are some examples. The first three each contain one error which is underlined.

*I come to visite you.*
Message: spelling error Detection flag (DF) **Case 1**



*A sheriff's wife is married with the law.*
Message: This preposition may
be wrong or superfluous. Warning flag (WF) **Case 2**

*I have selected those which apply for this story.*
No message No flag (nF) **Case 3**

The following examples are correct and should not provoke any flags.

*on the 15th of March*
Message: capitalization
error (*th* ) detected Detection flag (DF) **Case 4**

*I have decided to go.*
Message: Warning: "to decide"
is a false friend. Warning flag (WF) **Case 5**

*I have decided to go.*
No message No flag (nF) **Case 6**

On the following page we present an extract of a text written by a French native speaker. It shows how the error message classification can be used for continuous texts. The same classification can be applied to individual sentences where more specific errors are tested. The six possible cases are given next to each error and are represented by a number. The evaluator simply has to circle the number that corresponds to what the checker does for that error.

| | Error | | | No error | | |
|---|---|---|---|---|---|---|
| | DF | WF | nF | DF | WF | nF |
| But in 1992, the fact that some people are afraid of the loss of our neutrality risks to (risks ø) | 1 | 2 | 3 | 4 | 5 | 6 |
| preventing us to realize (from realizing) our professional ambition. | 1 | 2 | 3 | 4 | 5 | 6 |
| An other (Another) | 1 | 2 | 3 | 4 | 5 | 6 |
| problem could be brought by (caused) | 1 | 2 | 3 | 4 | 5 | 6 |
| the geographical place of Switzerland in the middle of Europe. If the Swiss Government still refuses the passage (transit) | 1 | 2 | 3 | 4 | 5 | 6 |
| of lorries of more than 40 tones (tons) | 1 | 2 | 3 | 4 | 5 | 6 |
| across the land (country), | 1 | 2 | 3 | 4 | 5 | 6 |
| the European Government could simply stop the negotiations (negotiations) | 1 | 2 | 3 | 4 | 5 | 6 |
| leads with us on other fields: (ø with us in) | 1 | 2 | 3 | 4 | 5 | 6 |
| Europe can introduce a lot of little obstacles towards us (for) | 1 | 2 | 3 | 4 | 5 | 6 |
| what could (which) | 1 | 2 | 3 | 4 | 5 | 6 |
| destroy our economic (economy) | 1 | 2 | 3 | 4 | 5 | 6 |
| by preventing us to export (from exporting) | 1 | 2 | 3 | 4 | 5 | 6 |
| our production by the introduction of technical norms or by taxes on exportation, for example. (exports) | 1 | 2 | 3 | 4 | 5 | 6 |
| **TEXT TOTAL:** | | | | | | |

The number of occurrences of each case can then be counted and multiplied by coefficients that reflect the quality of the flag.

Total of **Case 1**: multiply by +2
Total of **Case 2**: multiply by +1
Total of **Case 3**: multiply by -1
Total of **Case 4**: multiply by -2
Total of **Case 5**: multiply by -1
Total of **Case 6**: multiply by 0



This gives a total which should be positive for a grammar checker that is of some help to users. If the result is a negative total, then users should not use the program regularly to check their English. This type of numbered evaluation makes it easy to compare different products.

## 4. Concluding remarks

A bilingual grammar and style checker is a tool which will have a similar - if smaller - group of users to that of a word processor. It should therefore be easy to install, to learn and to use. All the tool's features should be comprehensible to users who know how to use a word processor but who are not computer scientists. User-friendliness also includes such questions as use of the mouse, well-organized help options, speed, and a certain level of robustness against unexpected inputs by the user. Instead of trying to establish independent standards of user-friendliness, the evaluator can assess it on the basis of personal impression.

The check lists in an evaluation kit should cover all areas of compatibility, user-friendliness, quality of written documents and tutorials, messages, speed, and on-line help. To check the actual detecting and correcting facilities, "authentic" texts, together with a simple error message classification system, seem to be a viable solution to the problem of evaluation. This will allow the evaluator to arrive at a meaningful statement concerning the usefulness of the tool.

## 5. References

FITIKIDES, T.J. (1963): *Common Mistakes in English*, Harlow, Longman.

GUIDA, G. & G. MAURI (1986): "Evaluation of natural language processing systems: Issues and approaches", *Proceedings of the IEEE*, 74 (2), 1026-1035.

LEECH, G. (1986): "Automatic grammatical analysis and its educational applications", in: LEECH, G. & C. CANDLIN (Eds.), *Computers in English Language Teaching and Research*, London, Longman.

PALMER, M. & T. FININ (1990): "Workshop on the evaluation of natural language processing systems", *Computational Linguistics*, 16 (3), 175-181.

RICHARDS, J. (Ed.) (1974): *Error Analysis: Perspectives on Second Language Acquisition*, London, Longman.

SANDERS, A. & R. SANDERS (1989): "Syntactic parsing: A survey", *Computers and the Humanities*, 23 (1), 13-30.

SINCLAIR, J. (Ed.) (1990): *Collins Cobuild English Grammar*, London, Collins.

THURMAIR, G. (1990): "Parsing for grammar and style checking", *Coling 90*, Helsinki, 365-370.

TSCHICHOLD, C. (1991): *The evaluation of computer-assisted writing tools for non-native speakers of English*, Lizentiatsarbeit, Universität Basel.