Computerized Grammar Checkers 2000: Capabilities, Limitations, and Pedagogical Possibilities

ALEX VERNON
University of North Carolina–Chapel Hill

Since commercial word-processing software integrated grammar checkers in the early 1990s, the composition community has scaled back its printed work on this technology to pursue more urgent issues of new technologies (new media, online composition, etc.) and their theoretical implications. Yet the functionality of grammar checkers has changed dramatically, and we should continue to explore the pedagogical possibilities of a tool now available whenever students compose on a computer. Grammar checkers remain troublesome and inaccurate—we should consider teaching students how to manage them and realize their potential despite their limitations. Grammar checkers allow us to discuss grammar at what William Wresch (1989) called the most “teachable moment” (p. 46): the moment of direct application to student writing. We might also use them to discuss the problematic nature of standard, normative written language conventions and the authority behind such conventions. In the hope of re-engaging some scholarly and practical interest, this article a) reviews the literature of grammar checker technology in composition; b) analyzes the current grammar checking capability of the most popular word-processing programs in the United States, Microsoft WORD 2000 and Corel WORDPERFECT 9.0 (2000); and c) concludes by suggesting uses of grammar checkers for composition teachers.

Reviewing recent evidence, Donald Ross (1991) found that “current style-checking programs make little positive difference in a writing classroom” (p. 88). Ross wanted a more powerful program adaptable to his particular classroom needs; specifically, he wanted the ability “to increase or decrease the lists of problematic items” for which checkers check (p. 87). Ross described his ideal program, one we infer he would find appropriate for the classroom:

A proper style-checking program should begin a text analysis by highlighting all the phrases or words in its repertoire. . . . The menu could also include (successively) a possible revision of each
error and an explanation. With this approach, the passive handbook would be replaced with an active look-up procedure sensitive to context, written in nontechnical language, and backed up with examples and tutorials. (pp. 98–99)

Such a potentially fruitful program is neither “an automated editing or proofreading environment, but one that depends on human judgment” (p. 102); such programs, which permit tailoring the list of checked potential problems, which regularly offer revisions while depending finally on human judgment, and which have context-sensitive Help features providing detailed, not-too-technical discussions with examples, are today a few mouse clicks away (see Figure 1).

Ross’ (1991) article appeared at a turning point in the history of grammar checker technology. Under consideration at the time were the stand-alone programs of the 1980s developed either by commercial companies or academic institutions. By the early 1990s, commercial word-processing software had begun integrating grammar checkers, beginning with Lifetree Associate’s VOLKSWRITER 4 (January 1989). VOLKSWRITER’s Perfect Grammar was derived from Houghton Mifflin’s CORRECTEXT—the same core program adopted by Microsoft for its initial integration of grammar checkers into Word for

---

1The terms grammar, style, and usage checker are used throughout the literature; style analyzers are slightly different programs. This article uses grammar checker, the most popular term. Style checker makes more sense, because with current checkers you really check a “style” defined by a set of checkable options many of which are not even grammatical errors. Usage checker may be the best umbrella term, as it avoids confusion with various programs’ titles as well as with the fuzzy distinction between grammar and style.
WINDOWS 2.0 in late 1991. WORDPERFECT followed a year later, adding GRAMMATIK 5 to WORDPERFECT for WINDOWS 5.2 (WORD for Mac 5.0 in 1992; WORDPERFECT for Mac interim version 2.1.3 and version 3.0 in 1993). Since the grammar checker boom of the early 1990s, starting in 1994 when WORDPERFECT for Windows 6.1 fully integrated GRAMMATIK 6, the technology has become more robust—the earlier version of WORDPERFECT had only partially integrated GRAMMATIK 5, resulting in an awkward interaction between the two programs. For WORD 97, Microsoft’s most radical change was its abandoning of CORRECTEXT for its own proprietary checker, created by its Natural Language Processing group. The grammar programs in WORD 2000 and WORDPERFECT 9.0 (OFFICE 2000) have changed negligibly from their 1997 versions.

Grammar checkers of the Y2K generation have made undeniable gains in functionality over their predecessors of a decade ago; their reliability and accuracy, however, have improved only minimally. I suggest that we revisit the pedagogical possibilities of grammar checkers not because they are any more accurate, but because they are more functional and—with their incorporation into word-processing programs—they are as ubiquitous as any software program can hope to be. Composition professionals need to continue to explore how we might take advantage of the current capabilities of this admittedly faulty yet readily available resource. Perhaps we can avoid purchasing tutorial software or workbooks with lessons disconnected from the student’s writing. Both our students and grammar checkers have trouble with grammar; the unreliability of grammar checkers, given their ubiquity, becomes a reason to address them in the classroom. Students who understandably allow confusing grammar checker feedback to reinforce feelings of intellectual inadequacy for grammatical knowledge could very well need the reassurance of a classroom critique of these programs. If students encounter these troubling programs whenever they compose on a computer, and if we accept our responsibility to teach grammar in context and to help students exploit the potential and understand the limitations of whatever resources they have at hand, then we should review our approach toward integrating grammar checkers into composition instruction.

This article a) reviews the literature of grammar checker technology in composition; b) describes and analyzes the current grammar programs of the most popular word-processing software in the U.S. today, Microsoft WORD 2000 and Corel WORDPERFECT 9.0 (OFFICE 2000); and c) concludes by suggesting uses of grammar checkers for composition

---

2 WORD 2000 has added five new checking options (likely inspired by WORDPERFECT): Sentence Length, Successive Nouns, Successive Prepositional Phrases, Use of First Person, and Words in Split Infinitives. Most notably, one can now disable the check for lengthy sentences (now defined as more than 60 words, an increase of 20 from WORD 97).

3 Implicit in this proposal is the assumption that the reader has made ideological peace with the idea of teaching some measure of normative, standard English language use at the risk of “reproducing oppression or colonial mentalities” (p. 482). This phrase, from Cynthia Selfe and Richard Selfe (1994), described their general concern over the potential danger of education with computers, but also applies to their suspicion of any “overly narrow—and erroneous—vision of ‘correct’ language use” as threatening a kind of linguistic and cultural imperialism (p. 489). For a number of us, grammar checkers might be more pedagogically intriguing less for teaching final draft editing skills than for deconstructing the authority of grammatical rules, thus challenging students with a conversation about the determinants of written language convention.
teachers—techniques applicable in noncomputer classroom environments as well as the most networked and virtual settings. This article does not attempt to prove whether grammar checkers will improve student competence, nor does it provide case study data; I only hope to reengage some scholarly consideration and practical interest, perhaps to encourage other specialists to pursue case studies and other field research.

First, however, a hypothetical: Midway through the semester, a student comes to your office frustrated with your persistent comments about the “wordiness” of his prose. You open his most recent paper on your computer in Corel WORDPERFECT and show him how to instruct the grammar checker to look for nothing except possibly wordy phrases (by checking the Formalisms, Jargon, Overstated, Redundant, Consecutive Elements [nouns and prepositional phrases], and Wordy options). You work through the paper, looking at the program’s suggested revisions, exploring its general discussion of those stylistic issues, perhaps rejecting some of its recommendations in favor of his solution and judging some of its flagged potential problems as nonproblems, and maybe identifying a few wordy phrases not indicated by the program. You can’t be sure what he has learned in those 20 minutes; you send him away and ask him to do this exercise with the grammar program of any word-processing software on future final drafts. And maybe by the end of the semester, his prose is noticeably more concise, and he has begun marking and suggesting revisions for wordiness in his peers’ drafts.

The technology has changed; pedagogical reconsideration is in order.

REVIEW OF THE LITERATURE

According to Gail E. Hawisher, Paul LeBlanc, Charles Moran, and Cynthia L. Selfe (1996), textual analysis computer programs have been around since the 1960s. These programs were tremendously popular in the 1970s “because they were the easiest writing tool to create” and because “they reflected existing pedagogical theory and practice” (p. 37). But the microcomputer and word processor would not sweep the country until the 1980s, and it wasn’t until the 1980s that we had forums willing or devoted to publishing about computers and composition. The first three articles appeared in 1983 (Daiute; Harris & Cheek; Kiefer & Smith). The literature of the 1980s focused on the efficacy and pedagogical merits of various university and commercially developed stand-alone programs. By the late 1980s and early 1990s, commercial software dominated the literature almost exclusively, a trend driven by the expense and difficulty for universities to design and maintain their own software during a period of rapid development and standardization of hardware and operating systems (which left the specialized university systems behind); the increasing skepticism of the efficacy of grammar checkers; and the sudden proliferation of inexpensive commercial packages. (The much-praised WRITER’S WORKBENCH cost a mere $2000, but ran on the expensive and specialized UNIX system. By 1990, commercial stand-alone checkers for common microcomputer operating systems cost under $100.)

My study of the literature through the early 1990s concurs with previous reviews. Hawisher et al. (1996), Eric Hobson (1995), and Rick Fischer and Elinor Grusin (1993) all reported a mixed reception of grammar checkers by composition professionals: Although the programs were widely included in the classroom, they also received weighty criticism. All of these reviews, however, considered the technology of the early 1990s at
the latest and response data from the 1980s. That Hawisher et al.’s most recent citation on grammar checker literature is from 1990 (Dobrin) only reflects the available scholarship. Since the early 1990s, the composition community has substantially scaled back printed work done on the subject, as specialists in computers and composition have focused their energies on the pressing issues presented by new media, online composition, and hyper-technologies, as well as their theoretical and sociopolitical implications (and arguably as grammar checkers hit their accuracy ceiling). Current academic literature on grammar checkers tends to involve business education and communication, and special interest education (MacArthur, 1996; Ray, 1997; Riviere, 1996; Scheffer, 1995; Spinks, Wells, & Meche, 1997 [though analyzing Grammatik IV, a 1990 product]).

Through the early 1990s, the arguments against using grammar checkers in composition classrooms made four contentions. First, the minimal benefits to be accrued were not worth the cost of purchasing or writing one’s own program. Second, the programs’ functionality was limited and their feedback inaccurate, unreliable, and unreasonable. Third, as mentioned above, grammar checkers enforced very strict and normative language practices and therefore became ideologically problematic. And, fourth, they reflected then-outmoded product-based pedagogy. Hawisher et al. (1996) stated this last position bluntly: Those who use grammar checkers “resist meaningful change by using computers to reinforce older and often conventional ways of thinking about learning” (p. 205).

Today, the integration of grammar checkers into standard word-processing software means that the issue of whether we should purchase stand-alone checkers or develop our own is moot. I will discuss the functionality and the accuracy of current grammar checkers below in my analysis of Word 2000 and WordPerfect 9.0. The ideological concern cannot be adequately addressed within the scope of this article, as it involves the much larger issue of whether and how we teach standard written language conventions at all. Grammar checkers, like grammar handbooks, should serve the instructor’s own theoretical approach to teaching grammar in general, or they should not be used. Today’s more flexible grammar checkers are less monolithic than those of the past; a teacher who does not consider the split infinitive taboo simply does not have the program check for split infinitives. Moreover, the teacher can use the grammar checker to introduce students to the

---

4 One of the most recent articles in composition studies, although published in 1998, appears to have been written much earlier. Beals’ “Between Teachers and Computers: Does Text-Checking Software Really Improve Student Writing?” tested the accuracy of the MLA’s Editor style-checking software, which came out in 1990 and was last updated in 1992, and compared it to the three predominate stand-alone checkers of the early 1990s: RightWriter, Correct Grammar, and Grammatik. Not only did Beals write about outdated early 1990s stand-alone products, he also cited Grammatik as belonging to Reference Software International when WordPerfect purchased Grammatik from RSI in 1993, and he analyzed an article from “a recent issue” of The New Yorker—the 11 October 1993 issue (p.70). All of which suggest that he originally wrote his article at the very latest in early 1994. Beals did cite two 1995 sources and one 1996 source, though these citations appear in his two endnotes, not the main body of his text. His use of the two 1995 sources is also suspicious: “Editor, for instance, is built with the assistance of works like the MLA Handbook for Writers of Research by Joseph Gibaldi (Fourth Edition, 1995. . . ) [and] Language, Gender, and Professional Writing by Francine Watterman and Paula A. Treichler (1995)” (72 note 1; emphasis added). (Beals’ article appeared in two publications in 1998; this article specifically refers to the English Journal version.)
very problematic nature of grammatical authority and normative language. By what right does Bill Gates prescribe language practices? The commercial interest of a near-global monopoly like Microsoft makes for an easy lesson. The major issue remaining to be addressed, then, is the pedagogical relevance of such seemingly product-oriented programs.

Arguments against grammar checkers as supporting old-fashioned product pedagogy were made during the heyday of process pedagogy. Part of the problem in the 1980s arose from the technological ordeal of implementing idiosyncratic university-developed programs and first-generation commercial checkers, an ordeal that inevitably occurred at the end of the drafting process and thus created a linear and cumbersome product-oriented event that interfered with a more naturally recursive writing process. Nancy Kaplan (1991) criticized grammar checkers on exactly these grounds: “When it operates after composing is complete, a program for checking style and usage implicitly enacts a model of the writing process that distances editing and polishing from composing, from the essence of meaning-making” (p. 29). If Kaplan found last decade’s technology inappropriate for process-pedagogy, we might presume that she would find today’s integrated grammar checkers—which an author can run at any time, even while composing with the grammar-as-you-go (GAYG) function that underlines potentially problematic sentences as soon as they are keyed (see Figure 2)—less linear, more recursive, more flexible, and more appropriate for the classroom (conversely, teachers who feel that students should not worry about grammar until the final stages of drafting can insist their students do just that; or we can empower student writers to make these decisions themselves). Furthermore, as composition studies has moved into a postprocess pedagogy of teaching academic discourse and the social-epistemic nature of language, any argument from an entirely process-oriented position—any pedagogical argument that does not consciously engage language use and convention—is itself old-fashioned. Most of us recognize that we cannot afford to neglect the product altogether.

Figure 2. GAYG Option of Word
Process-based arguments against grammar checkers—and many of the complaints about their feedback—often presuppose the absence of instructor guidance and tend to treat the programs as surrogate teachers or editors. William Scheibal and Gary Kohut (1991) remarked that checkers are no “substitute for an experienced human editor” (p. 48), and James Schick (1990) concluded that they do not adequately “substitute for knowing the applicable rule of English” (p. 35). Again, this position either mistakenly posits the role of teacher as mere editor, or mistakenly presupposes the absence of the instructor in the learning process. Current integrated grammar checkers are not designed to teach grammar, but to assist the writer in the identification of potential problems. Students without guidance might misuse the technology, become befuddled by it, and feel even more frustrated, discouraged, and “stupid” when facing future writing tasks.

One of the most recent studies comparing students using grammar checkers with a control group—from data collected in 1991 using a stand-alone checker—actually incorporated the absent-instructor assumption into its methodology. Fischer and Grusin (1993) compared in-class written assignments from two groups of students. The control group submitted their papers without having the opportunity to use a grammar checker, whereas the experimental group handed their papers in and then ran them through the checker. At the end of the term, Fischer and Grusin found no difference in grammatical competence between the groups. But why should they? “Since Grammatik gives the rule with each suspected error,” they wrote, “learning was presumed to occur” (pp. 22–23)—a presumption that does not stand to reason. Perhaps the students adopted the suggested revision without study; perhaps they supplied their own reason; perhaps they didn’t bother with the error at all. Analogously, Fischer and Grusin might have sent students home with grammar textbooks, without providing even extra credit as incentive, and presumed learning to have occurred. And their method of having the experimental group run the checker after turning in their papers defeats the purpose, for the students, of using and trying to learn from the checker. It would be, for them, merely academic.

A year before Fischer and Grusin (1993) collected their data, Mike Markel (1990) examined a number of studies making claims on the effectiveness of using grammar checkers in class, and he cautioned us against subscribing unquestioningly to their conclusions because the studies are often “methodologically flawed” or “anecdotal.” Richard Mende’s (1994) “Grammatik 5 for Windows: Random Thoughts From a White, Male, Middle-Class German Canadian,” for example, which has been cited as evidence against grammar checkers, is little more than the anecdotal grumblings of a business professor who spent an afternoon looking at a single grammar checker’s feedback on an essay; his most fervent complaint involves the “political correctness” of the checker, which noted what it perceived as potentially biased language.

One final pedagogical argument against using grammar checkers for composition entails a threat to teacher authority. Dobrin (1986) argued that the grammar checkers gain authority simply by being “embedded in a computer program” (p. 27), and Ray (1997) echoed this concern eleven years later: “If students feel that spell[ing] and grammar checkers make their work ‘correct,’ it sets up potential confrontations between student and professor when points are deducted for mechanical errors” that the checkers didn’t catch (p. 96). Such logic contradicts claims that grammar checkers should not be used because they are inaccurate and unreliable. When students consider 45% of grammar checker feedback as unhelpful or confusing (Thomas & Thomas, 1991, p. 24), the programs hardly
pose a threat to the teacher’s authority. They undermine themselves, not the teacher—they can’t undermine both. Because grammar checkers do tend to undermine their own potential value, and because students might object to human corrections not noted by the program, we ought to teach the checkers’ limitations and how students might work productively with them anyway. Finally, such confrontations between grammar checker and human grader are much less difficult to manage than those we face when students receive lower scores for mechanical or content problems not caught by their draft workshop peers, events that more seriously challenge the theoretical validity of the socially oriented peer classroom.

Ironically, process pedagogy informed one of the reasons for using grammar checkers. By relying on students using checkers outside of class, teachers could remove grammar instruction from the classroom and thus engage in more substantive dialogue about writing (Hawisher et al., 1996; Hull, Ball, Fox, Levin, & McCutchen, 1987; Thomas & Thomas, 1991). The assumption once again was that grammar checkers are teachers and learning necessarily happens, an assumption we know to be faulty. And much of the positive grammar checker literature from the 1980s no longer applies—these articles argued for the merits and uses of particular but now defunct programs in defense of financial expenditures for them.

However, we can find some substantive literature asserting the potential classroom benefits of grammar checker technology. Some of this literature is quantitative and sits in direct contrast with those studies demonstrating the technology’s ineffectiveness (e.g., Renshaw, 1991; Scheffer, 1995; though we should review their methods as well). As for pedagogical possibilities, we can look to either those articles on outdated programs or more general literature about teaching with grammar checkers. William Oates (1987) recognized that grammar checkers can inspire rewriting, and William Wresch (1989) similarly noted that they attract student interest and have the significant advantage over textbooks and drills in that they offer grammar instruction at its most “teachable moment,” the moment of direct application to student writing (p. 46). For Glynda Hull, Carolyn Ball, James L. Fox, Lori Levin, and Debora McCutchen’s (1987) “pedagogy of editing” (p. 104), such a moment becomes “an occasion for writers to recognize an error and to modify the rule that produced it” and become analytical about their own language use (p. 117). In his brief 1990 article, Michael Milone cautioned his reader that “guidance of the teacher is still the best way” for students to learn grammar, and that teachers need to emphasize that these programs try to help produce “correct writing, not good writing” (p. 23). With these cautions in mind, he presented several particularly sound teaching ideas:

- Model the grammar checker editing process in front of the class via projection, and discuss why the computer flagged items, the teacher’s response to the program’s feedback, and the technical limitations of the program.
- Have groups of two or three students analyze student texts and grammar checker feedback.
- Have students enable only certain checking options to focus on particular kinds of error.

This last item helps alleviate the concern that students will be too overwhelmed by the quantity of flagged items (and incorrect or trivial flags) to make any progress on any front, and reflects Hull et al.’s editing pedagogy as well as Erika Lindemann’s (1987) advice that teachers responding to student papers should focus on only one or two problems per paper.
Other works that reiterate the above grammar checker pedagogy are Wresch, Donald Pattow, and James Gifford (1988); Thomas and Thomas (1991); Kenneth Mayer (1991); and James Strickland (1997).5 These works all subscribe to the general advice provided by Ellen Nold (1975) for computers and composition in the field’s first published article, that “teachers can be fully in control of the experience”—can, and ought, because “a computer cannot substitute for a teacher” (pp. 272–273).

CAPABILITIES OF MICROSOFT WORD 2000 AND COREL WORDPERFECT 9.0

Before discussing the classroom possibilities of grammar checkers, we need to examine the capabilities of the latest technology. With the acknowledgment that my analysis for the purposes of this article of WORD and WORDPERFECT is more suggestive than it is comprehensive, it remains evident that the technology has changed dramatically.

WORD’s and WORDPERFECT’s grammar checkers operate similarly. Each program features several distinct writing “styles,” such as Student Composition and Technical, and check a text against the selected style. The styles are defined through different combinations of checkable options culled from a fixed list. Figure 1 shows the dialog box and the visible checking options for WORDPERFECT’s Student Composition style, which includes all but four of the grammar checker’s 67 total checking options (“Rule Classes”), from Abbreviation to Wordiness. During an actual check, both programs identify the potential error, attempt an explanation, and often offer a suggested solution.6 Both programs also offer a grammar-as-you-go (GAYG) option (see Figure 2), which underlines a potentially problematic sentence once it is written (though GAYG and the normal process do not always catch the same potential errors).

A strictly quantitative comparison suggests WORDPERFECT’s grammar checker as the more capable. Its 67 checking options support 11 built-in styles, each of which can obtain one of three different levels of formality (informal, standard, formal). WORD, on the other hand, gives only 21 checking options supporting only five styles that are essentially levels of formality (casual, standard, formal, technical, and custom). It does not, therefore, claim multiple formality levels for each style. Additionally, WORDPERFECT allows users to adjust the sensitivity to certain potential errors such as sentence length that can be numerically defined (see Figure 1), whereas for WORD, though these options can be disabled, the numbers cannot be changed: No more than 60 words per sentence, three nouns or

5Little (1987) found that the HOMER text analysis program encouraged fruitful discussion, and John Day (1988) argued that the grammar checker process helps students think about their writing. Michael Neuman (1991) contended that checker technology requires teachers to spend too much time explaining why problems are beyond its abilities and how to judge the relevance of the suggestions. More radically, Randy Smye (1988) and Joel Nydahl (1990) suggested that grammar checkers may lead to deep-structure revision, though Robert Kozman (1991) found no evidence to support this.

6With both programs, users must pay attention to how the online Help discussions define checking options. Parallelism checks, for example, tend to focus on verb forms over the parallel use of articles.
prepositional phrases in a row, and no more than one word splitting an infinitive (that it always allows one seems remarkably progressive).

A quantitative comparison, however, doesn’t adequately address the differences, especially as to how the programs would affect the composition classroom. WordPerfect’s 67 checking options are very specific; Word’s 26 generally include more in each option. Word’s Sentence Structure option checks

sentences, run-on sentences, overuse of conjunctions (such as ‘and’ or ‘or’), nonparallel sentence structure (such as shifts between active and passive voice in a sentence), incorrect sentence structure of questions, and misplaced modifiers. (online Help)

WordPerfect has individual options for most of these, making it more suited to classroom targeting of particular grammar troublespots. But WordPerfect isn’t perfect. Many of its checking options are redundant; it has, for example, one option that checks for unbalanced parentheses, brackets, and quotation marks, and a separate option that checks just for complete quotation mark pairs. Its Incorrect Verb Form, Conditional Clause, and Subjunctive checking options all check for the subjunctive mood. Finally, some WordPerfect checking options are really functions—User Auto Replacements and the User Replacements—and some are relatively worthless if easy to program. The ambitious seeming Paragraph Problem option identifies paragraphs of only one sentence.

Furthermore, WordPerfect’s 11 styles with three different levels of formality per style is rather misleading. We should hardly consider Gayg a style, yet WordPerfect does, so that selecting this option through the toolbar does not, as we might intuit, utilize the user’s selected default style—Student Composition, say—but instead uses the Gayg’s style and defined checking options. The Spelling Plus and Quick Check styles are arguably indistinguishable. Even more problematic are the formality levels. The program supposedly allows the choice of among three levels of formality for each style, yet selecting one does nothing. The Student Composition style, for example, checks the exact same options regardless of the selected formality level. If the formality levels somehow change the tolerance for errors, the program doesn’t indicate this, nor can this user imagine how such a function would work. The formality selection is further belied by the pop-up description for each style, which defines the style partially in terms of its formality, as well as the name of two of the styles: Formal Memo or Letter and Informal Memo or Letter.

Finally, WordPerfect does provide the more robust textual analysis (even attempting a parsing tree diagram for any sentence), though I find that Word has the more user-friendly online Help. Its Assistant function is very context specific and can remain on screen for the entire grammar check, changing its information as the checker encounters the next error (see Figure 3).

COMPARATIVE PERFORMANCE ANALYSIS

Despite the notable gains in grammar checker functionality, improvements in error-catching ability and decreases in false error identification have not been sufficiently
quantified through research. I found no data comparing the performance of particular grammar checkers from the mid 1980s to their current incarnations. Such data would be useful, though likely not worth the technical hurdles involved in reconstructing the old systems to apply them against the same texts. The technical limitations of grammar-checking technologies have long been known; we have not crossed, and might never cross, the conceptual barrier to the next level of programming competence. Accuracy data are less important when we treat grammar checkers not as editors but as tools, so that to evaluate the software’s effects on student learning, we instead need reliable data contrasting student performance in a traditional classroom with one that includes judicious and innovative grammar-checker instruction using today’s technology.

Still, anyone interested in using grammar checkers in composition instruction ought to have a sense of the current technology’s accuracy and reliability. I strongly recommend

---

7The only source I know to have tested subsequent versions with similar methodology is Rubin Rabinovitz, a commercial magazine commentator who in 1994 found that GRAMMATIK 6 increased its error detection performance from GRAMMATIK 5 (1992) by only 4% (42–46%). He did not say whether he used the same text. His 42% for GRAMMATIK 5 contrasts with PC Magazine’s 53% (Bender, 1994, p. 110). Ray’s 48% for GRAMMATIK 6 more closely matches Rabinovitz’ 46%.
instructors independently examine the technology available at their institutions and in the context of their particular pedagogical goals. The following analysis summarizes my examination and sense of the technology.

First, I tested the style checkers of \textsc{Word} and \textsc{WordPerfect} against \textit{The Allyn & Bacon Handbook}'s list of 28 sample items (single or multiple sentence constructions) containing 36 errors that represent over 90\% of grammar problems (Rosen & Behrens, 1997, “Spotlight on Common Errors,” inside back cover). Next, I applied the checkers to student essays from a second-semester first-year composition course. The two papers are drafts of short thematic explications of different 20th century American novels. For test purposes, all checking options in both checkers were selected for operation on the texts.

Of the 36 representative errors in \textit{The Allyn & Bacon Handbook}, \textsc{WordPerfect} correctly identified 17, offered incorrect solutions for two of these, and mistakenly identified two errors. \textsc{Word}, on the other hand, correctly identified 12 errors, offered no suggestions for three of these, and made no mistaken identifications. Both programs additionally identified potential problems that these sentences weren’t intended to illustrate, though they did so differently and inconsistently. Table 1 provides the \textit{Handbook}'s sample sentences, the errors illustrated, and the results of each program’s check.

Such a limited test obviously cannot claim a comprehensive review of the functionality and reliability of the grammar checkers, as a quick exercise demonstrates. To the \textit{Handbook}'s 36 representative errors, I added a 37 by changing the last sentence—from \textit{The leader, whom I admire most, is Lincoln}, to \textit{The leader, who I admire most, is Lincoln}. Neither grammar checker flagged the case error, though both claim the ability to do so. Yet this test does lead to some interesting general observations. Most significantly, the checkers are restricted to the sentence level. They cannot identify inconsistent verb tenses or vague pronoun references across terminal punctuation. Checking for such items across sentences may be impossible, because it requires contextual knowledge beyond the checker’s purely structural knowledge. Unfortunately, a fair portion of novice writers’ errors are exactly these cross-sentence culprits. A related context-dependent set of errors, which constitute a good portion of sentence-level difficulty and which both checkers utterly fail to identify, involves misplaced and dangling modifiers. Finally, interrogative constructions confuse the checkers. Both failed to identify pronoun case errors in interrogatives but successfully identified them in simple declarative sentences.

With the analysis of the two student drafts, I will not compare all instances of potential error flagged by the checkers, their descriptions and suggested replacements, with those of the other nor a human editor; this analysis instead is meant to supplement the \textit{Allyn & Bacon} test’s rather limited purview through select, noteworthy examples culled from the feedback provided on the two sample first-year essays, limited to the following more substantial observations.\(^8\)

\(^8\)I also used both programs to check a technical document, the “Solid Rocket Booster” section of NASA’s \textit{NSTS 1988 News Reference Manual} on the space shuttle. This document, published online, contains no real grammatical errors. It was checked using the Technical style setting for both checkers. Both grammar checkers thus turn off the Passive Voice checking option, and they mainly stumble where one might predict: special word usage, particularly nouns and verbs.
<table>
<thead>
<tr>
<th>#</th>
<th>Sample Sentence(s)</th>
<th>Illustrated Error</th>
<th>WORD Results</th>
<th>WORDPERFECT Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The scarf is Chris.</td>
<td>apostrophe use</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>It is her's.</td>
<td>apostrophe use</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Give the dog it's collar.</td>
<td>apostrophe use</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>It's a difficult thing.</td>
<td>apostrophe use</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>This is him.</td>
<td>pronoun case</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>It was me.</td>
<td>pronoun case</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>Is that her?</td>
<td>pronoun case</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>The ball landed between she and I.</td>
<td>pronoun case</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>The ball landed between she and I.</td>
<td>pronoun case</td>
<td>+use of first person</td>
<td>✔</td>
</tr>
<tr>
<td>10.</td>
<td>Her and me practice daily.</td>
<td>pronoun case</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td>Her and me practice daily.</td>
<td>pronoun case</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.</td>
<td>She liked the work. Still, she keeps to herself.</td>
<td>inconsistent verb tense</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13.</td>
<td>If it would be any colder, the pipes would freeze.</td>
<td>faulty mood (subj.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.</td>
<td>Lie the books here.</td>
<td>improper verb</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.</td>
<td>Then lay down.</td>
<td>improper verb</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16.</td>
<td>The reason she wins are her friends.</td>
<td>subject-verb agreement</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>17.</td>
<td>My friends and Sue likes her pizza hot.</td>
<td>subject-verb agreement</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Identifies as a contraction, which may be too informal.

If the user ignores the advice, it tries again, by recommending a subject pronoun instead of an object pronoun after the linking verb.

Identifies/corrects verb form; does not then catch the need for plural pronoun their instead of her.

✔ identified/corrected; — not identified; ✓— identified/not corrected; + additional error; ! nonerror identified

(continued)
### TABLE 1
(Continued)

<table>
<thead>
<tr>
<th>#</th>
<th>Sample Sentence(s)</th>
<th>Illustrated Error</th>
<th>WORD Results</th>
<th>WORDPERFECT Results</th>
</tr>
</thead>
</table>
| 18. | Neither her friends nor Sue like their pizza cold. | subject-verb agreement | ✓ | ✓ “Neither her’ “these modifiers aren’t usually used together”.
| 19. | If our cousins arrive today. | sentence fragment | ✓— | ✓— “The subordinating conjunction if can’t begin an independent clause”—no fix. |
| 20. | He’s here now later he’ll go to Iowa. | fused sentence | — | Correctly identified, but gives three bad solutions: “. . . now later. He’ll . . .” “. . . now later, but he’ll . . .” “. . . now later, and he’ll . . .” +! catches contraction he’s, but not he’ll. |
| 21. | He’s here now, later he’ll go to Iowa. | comma splice | — | same as #20 |
| 22. | If Amy and Mae arrive by noon, she will call. | unclear pronoun reference | — | — |
| 23. | His late arrival caused delays. This couldn’t be avoided. | vague pronoun reference | + suggests changing second sentence to active voice + catches contraction | + suggests changing second sentence to active voice + catches contraction |
| 24. | Transfer students must take care. The transfer may cost you credits. | inconsistent pronoun reference | — | + suggests avoiding second person address |
| 25. | The car was made for a demanding driver with a wide body. | misplaced modifier | + suggests making was made active voice | + suggests making was made active voice |
| 26. | Those who walked quickly reached home. | misplaced modifier | — | — |

✓ identified/corrected; — not identified; ✓— identified/not corrected; + additional error; ! nonerror identified

(continued)
<table>
<thead>
<tr>
<th>#</th>
<th>Sample Sentence(s)</th>
<th>Illustrated Error</th>
<th>WORD Results</th>
<th>WORDPERFECT Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>27.</td>
<td>Being from the country, the skyscrapers looked threatening.</td>
<td>dangling modifier</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28.</td>
<td>He could speak both with passion and could be coldly calculated.</td>
<td>parallelism fault-verb form</td>
<td>✓—</td>
<td>+suggests making could be active voice</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Verb Use error—all verbs should be either passive or active. +first gives passive voice error.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29.</td>
<td>Wash all these: the floor, the window, and clean up my sink.</td>
<td>Parallelism fault</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30.</td>
<td>Jen needed cash so Amy loaned her some.</td>
<td>comma use-joining sentences with conjunction</td>
<td></td>
<td>+Suggests that loaned should be lent. Jen is not in spelling dictionary.</td>
</tr>
<tr>
<td>31.</td>
<td>Incidentally I’ll arrive, after sunset.</td>
<td>comma use-introductory expression</td>
<td>✓</td>
<td>+catches contraction</td>
</tr>
<tr>
<td>32.</td>
<td>Incidentally I’ll arrive, after sunset.</td>
<td>comma use</td>
<td></td>
<td></td>
</tr>
<tr>
<td>33.</td>
<td>Tim, and I collect bugs fish and rocks.</td>
<td>comma use</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>comma use-series</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35.</td>
<td>My oldest brother Greg is 28.</td>
<td>comma use-essential</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>information</td>
<td></td>
<td></td>
</tr>
<tr>
<td>36.</td>
<td>The leader, whom I admire most, is Lincoln.</td>
<td>comma use-essential</td>
<td></td>
<td>+suggests that 28 should be spelled out</td>
</tr>
<tr>
<td></td>
<td></td>
<td>information</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

✓ identified/corrected; — not identified; ✓— identified/not corrected; + additional error; ! nonerror identified
WORDPERFECT’s grammar checker is more aggressive. It detects more errors, suggests changes for more detected errors, wrongly flags more nonerrors, and finds more problematic sentences but improperly identifies the problem. WORDPERFECT discovered nearly half-again as many legitimate errors, though many of these were relatively minor and 16% weren’t actually errors—its greater error detection is threatened by the user’s impatience. We can partially compensate for this problem by taking advantage of WORDPERFECT’s more flexible checking options and disabling those less useful to our pedagogical goals.

Checkers frequently flag a troubled sentence but incorrectly identify the issue. Something is wrong, they just can’t determine what. The real problem is not always easy to discover—a subtle syntax change to a technically correct sentence may appease the checker. This problem of perplexing feedback has led some composition specialists to recommend that grammar checkers not be used for basic writers. As speculated earlier, however, teaching the program’s limitations might help ease students’ feelings of intellectual inadequacy.

Both programs have trouble with more elaborate sentences (especially subject-verb agreement) and sometimes with multiple potential errors in one sentence and with checking the grammar of their own suggested revisions.

COMPOSITION PEDAGOGY

We can make creative pedagogical application of the expanded functionality of grammar checker programs, and we can use both their quirky prose classifications and their unreliable feedback as points of instruction. The new technology allows us in workshops to focus on very specific applications to student texts with clear, limited, and manageable goals. We can do this in as few as one or two workshops per semester, and thus avoid a surface-obsessed, product-based course. I see two distinct pedagogical methods for incorporating grammar checkers: a) as a springboard for conversation about language conventions and usage authority, and b) as a practical tool for improving student revising and editing skills.

Language Conventions and Usage Authority

In my class last year, I began by observing the inextricable link of the programs’ checking options with certain styles of writing. This link led to a discussion of how rhetorical context determines grammar and style expectations and standards, and of how these language practices and standards aren’t absolute. Students in the class were quick to point out the artificiality of defining specific discourses through particular grammar and style features. Although discourses and genres do shape our style, we also appreciate the variety of styles within particular discourses, the negotiations authors make between their idiosyncrasies and discourse conventions, and the rather hazy line distinguishing discourses at all.

We might also discuss in class what styles the word-processing software of our choice employs, the checking options it considers appropriate for that style, and the implications thereof. WORDPERFECT’s Student Composition style, for example, raises a number of questions. First, I asked my class why the grammar and style conventions for students are
any different from those for nonstudents (which I suppose we are to equate with adult professionals). Students then challenged the assumption of a universal student/academic discourse—as if the conventions of business communication and poetry explication are the same, and again are different for students as for businesspeople and professors (not to mention how this strict style applies to personal essays written in composition classes). And what about the specific checking options? Student Composition checks for instances of Second-Person Address as likely too informal; we might expect it to also check for instances of First Person, or First Person Singular perhaps in conjunction with certain verbs: I feel, I think, I believe. No such option exists. Nor does the Student Composition style look for those categorical words we love to caution our students against: never, always, all, every, none, obviously, clearly. By raising these examples for students, we can also explore whether they are valid, noteworthy “errors.” Microsoft Word lends itself to a discussion of style versus grammar, as it neatly categorizes its checking options as one or the other. Why does Word insist that whether numbers are presented in numeral or word form is a matter of grammar instead of style, and conversely why does it consider “sentences in which there is more than one possible referent for a pronoun” (online Help) a matter of style instead of grammar? Is an absolute, context-independent distinction between style and grammar possible?

One possible answer to some of these questions may involve the target market for these products: businesses and the adult professional. We could then discuss with students how the grammar checkers are designed especially for that group of writers. Selfe and Selfe’s (1994) paper on the white-collar audience of interface design might prove useful in such a conversation. We can, in other words, use grammar checkers not solely to replace our red pen, but to raise open-ended issues of language use.

We especially need to discuss with students what stylistic features these programs simply cannot check, both because of technological limitations and because of what Patricia Bizzell (1992) has called the “infinite regress of context” (p. 96). In addition to inconsistent verb tenses and vague pronoun references as mentioned above, grammar checkers can’t handle the literary present tense, word meaning and choice, word variety, appropriate tone and diction, sentence variety beyond repetition of the first word, parallelism in adjacent independent clauses, graceful integration of quotations, and so forth.

Revising and Editing Skills

Revising and editing are often difficult to distinguish. As noted above, Kaplan (1991) saw grammar checkers as enforcing an artificial and wrongheaded segregation of writing as meaning-making versus editing as nonmeaning-making. But the classroom application, not the programs themselves, bear the blame. Granted, any editing or language change at the very least affects nuance, but I suspect many first-year composition teachers enact a more practical pedagogy that views student editing as a clarification and strengthening of prose rather than as deeper revisionary work. In addition to the techniques mentioned at the end of the literature review section, concrete editing activities with which we can engage students include the following:

- Ask students to brainstorm other possible checking options, other styles, and the checking options appropriate for that style.
• Have students, either individually or in small groups, articulate responses to program feedback to their own writing. Students could present their responses either in writing (in a separate document, or in annotations) or orally. Having students work together especially helps students decipher incorrect and misleading program feedback.

• Task students to run *Grammar-as-you-go* on a completed text and—since the GAYG function simply underlines suspect constructions—to address the flagged items without accessing the checker’s feedback (by clicking the right mouse button). An improved sentence makes the underlining go away. This is a version of the technique of Hull et al. (1987) for marking the text without identifying the problem, thereby teaching students to learn to edit themselves. It also avoids the confusion of misleading feedback altogether.

• Focus on a particular issue by having students check their papers with only one checking option (or several options that look for the same essential issue), as described with “wordiness” above. Increase the number of issues checked as the semester progresses.

• Ask students to write “bad” sentences, either to successfully trigger the checker or fool it.

• Hold contests, pitting human checkers against one another and against the computer. Have the computer and the students independently search for subject-verb agreement errors.

• Compare the rules of the grammar checker with the discussions of the same issues in your writing handbook. If you choose an issue on which the sources disagree, you have instantly challenged any monolithic sense of language use students might possess, and have taken a solid step toward enabling students to analyze the rules, the rhetorical situation of their writing, and deciding for themselves (not to mention just getting the handbook off the shelf).

• Analyze how *WordPerfect* parses sentences when they are flagged. With both programs, discuss the statistical analyses and its relevance for revision, perhaps mentioning workplace notions that specify certain reading levels for certain audiences (Fischer & Grusin, 1993 and Klem & Moran, 1991, made the pragmatic argument for teaching grammar checkers: because students will encounter their use in the workplace).

Although all of these exercises can be creatively adapted to networking, online, and distance learning situations, none of them requires a fully established computer classroom. A single computer with projection capability, computer screen shots either distributed as handouts or projected via an overhead, and homework assignments can all work to get around the technology restrictions of the most low-tech and no-tech classroom settings. Many of these suggestions take advantage of the shortcoming of grammar checkers, yet we should remember that grammar checkers are not entirely broken. They can help polish our prose. And in no way do I intend to propose what Hawisher et al. (1996) politely called a “less forward-looking” pedagogy (p. 196). Process and social-epistemic pedagogies may be more theoretically sophisticated, but we should not abandon the product and the conventions of the discourse we expect students to learn. The healthiest pedagogy will incorporate all three approaches in its classroom practices. Theoretical (and political) objections to grammar checkers can become points of instruction, and students can further learn to appreciate the flexibility and complexity of the writing process by experimenting
with grammar checkers, using them in different ways at different points in the process. As we guide students through editing and revision, we can use grammar checkers to teach grammar not abstractly but in the context of their papers, and perhaps engage their interest in a way that hardbound handbooks cannot. At a minimum, students need to be taught about the limitations of the technology so that they do not—when working in their rooms or in computer labs—let those limitations discourage them about their abilities.

The literature of grammar-checker technology in composition has largely not kept pace with the technology. Moreover, the available literature needs to be scrutinized in light of current composition thinking and practices, and smart, rigorous classroom experimentation is in order.

The largest challenge for anyone teaching grammar in the composition classroom remains convincing students that correct writing is not necessarily good writing. How do we encourage students to stretch their syntactic muscles, to risk incorrect usage in pursuit of complicated structures expressing complex ideas? How do we teach them to play with sentences? How do they learn what we know—that effective language use does not always obey conventional notions of structural clarity even in a conventional academic essay? How do we teach them to judge for themselves? These questions pester all writing instructors, however they teach (or don’t teach) grammar.


REFERENCES

Hawisher, Gail E.; LeBlanc, Paul; Moran, Charles; & Selfe, Cynthia L. (1996). Computers and the


Nydahl, Joel. (1990). Teaching word processors to be CAI programs. College English, 52(8), 904–915.


